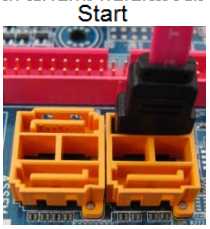
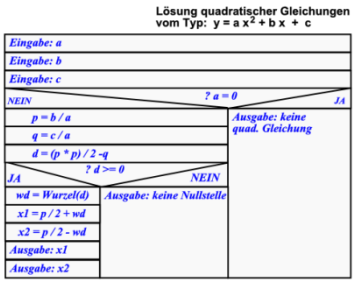
Informatik und Medienbildung

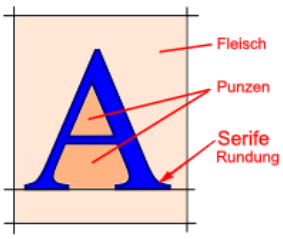
für die Orientierungs- und Sekundarstufe I
(Gymnasium, Gesamtschule)

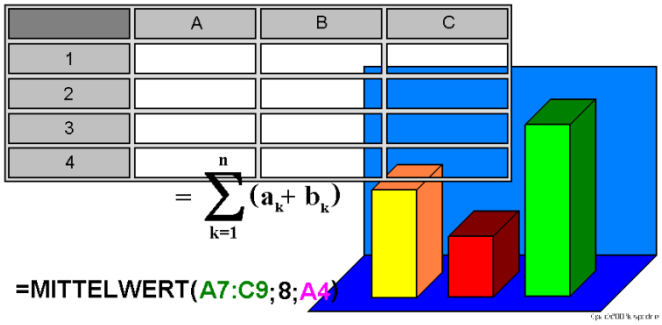
- Computer und Medien -

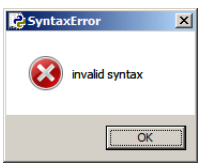
Autor: L. Drews

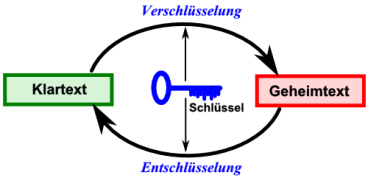










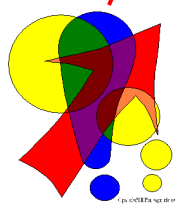


unredigierte Entwicklungs-Version 0.3d (2020)

Orientierung am RLP "Informatik und Medienbildung" MV (und an der an Anhörungsfassung)

(c,p) 2015-2020 lern-soft-projekt: drews

Legende:
mit diesem Symbol werden zusätzliche Hinweise, Tips und weiterführende Ideen gekennzeichnet



Nutzungsbestimmungen / Bemerkungen zur Verwendung durch Dritte:

(1) Dieses Skript (Werk) ist zur freien Nutzung in der angebotenen Form durch den Anbieter (lern-soft-projekt) bereitgestellt. Es kann unter Angabe der Quelle und / oder des Verfassers gedruckt, vervielfältigt oder in elektronischer Form veröffentlicht werden.

(2) Das Weglassen von Abschnitten oder Teilen (z.B. Aufgaben und Lösungen) in Teildrucken ist möglich und sinnvoll (Konzentration auf die eigenen Unterrichtsziele, -inhalte und -methoden). Bei angemessen großen Auszügen gehört das vollständige Inhaltsverzeichnis und die Angabe einer Bezugsquelle für das Originalwerk zum Pflichtteil.

(3) Ein Verkauf in jedweder Form ist ausgeschlossen. Der Aufwand für Kopierleistungen, Datenträger oder den (einfachen) Download usw. ist davon unberührt.

(4) Änderungswünsche werden gerne entgegen genommen. Ergänzungen, Arbeitsblätter, Aufgaben und Lösungen mit eigener Autorenschaft sind möglich und werden bei konzeptioneller Passung eingearbeitet. Die Teile sind entsprechend der Autorenschaft zu kennzeichnen. Jedes Teil behält die Urheberrechte seiner Autorenschaft bei.

(5) Zusammenstellungen, die von diesem Skript - über Zitate hinausgehende - Bestandteile enthalten, müssen verpflichtend wieder gleichwertigen Nutzungsbestimmungen unterliegen.

(6) Diese Nutzungsbestimmungen gehören zu diesem Werk.

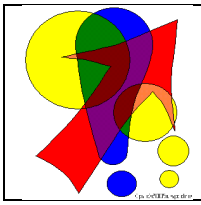
(7) Der Autor behält sich das Recht vor, diese Bestimmungen zu ändern.

(8) Andere Urheberrechte bleiben von diesen Bestimmungen unberührt.

Rechte Anderer:

Viele der verwendeten Bilder unterliegen verschiedensten freien Lizenzen. Nach meinen Recherchen sollten alle genutzten Bilder zu einer der nachfolgenden freien Lizenzen gehören. Unabhängig von den Vorgaben der einzelnen Lizenzen sind zu jedem extern entstandenen Objekt die Quelle, und wenn bekannt, der Autor / Rechteinhaber angegeben.

public domain (pd)	Zum Gemeingut erklärte Graphiken oder Fotos (u.a.). Viele der verwendeten Bilder entstammen Webseiten / Quellen US-amerikanischer Einrichtungen, die im Regierungsauftrag mit öffentlichen Mitteln finanziert wurden und darüber rechtlich (USA) zum Gemeingut wurden. Andere kreative Leistungen wurden ohne Einschränkungen von den Urhebern freigegeben.
gnu free document licence (GFDL; gnu fdl)	
creative commens (cc) creative commons	BY: od. neu … Namensnennung … nichtkommerziell … in der gleichen Form … unter gleichen Bedingungen
Die meisten verwendeten Lizenzen schließen eine kommerzielle (Weiter-)Nutzung aus!	



Bemerkungen zur Rechtschreibung:

Dieses Skript folgt nicht zwangsläufig der neuen **ODER** alten deutschen Rechtschreibung. Vielmehr wird vom Recht auf künstlerische Freiheit, der Freiheit der Sprache und von der Autokorrektur des Textverarbeitungsprogramms microsoft ® WORD ® Gebrauch gemacht.

Für Hinweise auf echte Fehler ist der Autor immer dankbar.

Inhaltsverzeichnis:

Seite

Vorwort **10**

Einleitung **12**

Jahrgangsstufe 5 ..... **13**

0. Warum Informatik und Medienbildung? **13**

1. Programmieren? Kinderleicht! **15**

Merksätze: Algorithmus 16

Merksätze: Sequenz 17

1.1. Algorithmen in unserem Leben **18**

Merksätze: Algorithmus 19

Merksätze: Programm 22

1.2. Programmieren von Einplatinen-Minirechner / Experimentier-Board's **24**

1.2.0. Welches Board haben wir? 24

1.2.0.1. Calliope mini 26

1.2.0.2. bbc micro:bit 27

1.2.0.3. OxoCard 29

1.2.0.4. Arduino / Genuino 29

1.2.0.5. BOB3 30

1.2.0.6. ESP-32 und ESP-8266 31

1.2.0.7. IoT-Bausteine 31

1.2.1. Programmieren mit NEPO (Open-Roberta-Lab) 33

1.2.2. Wir sagen dem Computer, was er machen soll! – Turtle-Graphik 34

1.2.2.x. Kara und Karol steuern 34

1.2.2.x.0. Aufbau der Welt von Kara und Karol 34

1.2.2.x.1. Erzeugen von Mustern 34

1.2.2.x.2. Kara, Karol und ihre Welt informatisch gesehen 35

1.2.2.x.3. Vereinfachung von Algorithmen 35

Merksätze: Wiederholung 36

2. Bilder und Grafiken gestalten **37**

2.1. Malen am Computer - Rastergrafiken **38**

Merksätze: Raster-Grafik 38

2.1.1. Speicherplatz-Bedarf bei Raster-Grafik-Dateien 40

BMP-Dateien 41

BMP-Dateien mit Lauflängenkodierung 42

BMP-Dateien mit Farb-Tabellen 42

moderne Dateitypen 43

PNG 43

JPG 43

2.1.2. Rastergrafik in der Praxis 43

Scanner 43

Digital-Kamera's 43

2.1.2.1. Flächen-Berechnung bei Pixel-Grafiken 43

2.1.3. Objekte und Strukturen in Rastergrafiken 44

2.2. Zeichnen am Computer – Vektorgrafiken **46**

Merksätze: Vektor-Grafik 46

Objekte und Strukturen in Vektor-Grafiken 47

2.3. Was ist besser – Pixel- oder Vektor-basiertes Zeichnen? **49**

2.4. Bild- und Foto-Bearbeitung **51**

3. Präsentationen mit Bildern gestalten **53**

3.1. Bilder suchen 54
3.2. Präsentation im Stil von PowerPoint und Co 55
5.x.1. informatische Grundstrukturen in Präsentationen 58
3.2.x. Präsentationen beurteilen 64
3.2.x. Projekt: Erstellen einer Themen-gebundenen Präsentation (Auftragswerk) 65
Objekte und Strukturen in Präsentationen 67
3.x. Präsentation von Zusammenhängen mit MindMap und Co 71
Vergleich von MindMap und ConceptMap 72
Objekte und Strukturen in MindMap- und ConceptMap-Dokumenten 72

A. Grob-Aufbau und Nutzung eines Rechners 74
A.1. Maus und Tastatur 74
A.2. Touch-Pad und Touch-Bildschirme 76
A.3. Sprach-Steuerung 77
A.4. Gesten-Steuerung 78

B. Computer-Netze 79
B.1. Login - Anmeldung in Computer-Netzen 79
B.1.1. Logout – muss das sein? 80
B.2. Verhalten in Computer-Netzen 80

C. Grundprinzipien der Informatik 81
C.x. Hard- und Software 81
Merksätze: Hardware 81
Merksätze: Software 81
C.x.2. große und kleine Computer 82
C.x. graphische Nutzungs-Oberflächen – der Desktop 84
C.x.1. Was sind Betriebssysteme? 84
Merksätze: Betriebssystem 85
C.x.1.1. mircosoft Windows 85
C.x.1.2. apple macOS und iOS 86
C.x.1.3. Linux – die kostenfreie Alternative 86
C.x.1.4. Android – Linux für Smartphone's 86
C.x. der eigene / die eigenen Arbeits-Rechner 89
C.x.y.1. Welche Software brauchen wir? 89
C.x.y. portable Software 89
C.x.y.0. Prinzip der portablen Programme / Apps 89
C.x.y.1. das "Portable Apps"-Menü-System 90
Exkurs: Wie richte ich einen USB-Stick für portableApps ein? 91

D. Daten – Was ist das? 93
D.1. Was sind Daten? 93
Definition(en): Informations-Einheit bit 93
Definition(en): Daten 93
Definition(en): Information 94
D.2. Wie wichtig sind meine Daten? 94
D.3. Wie werden meine Daten gespeicher! 94
Definition(en): Datei 98
D.3. Wo kann ich am Besten speichern? Wo finde ich meine Daten? 98
D.3.0. Laufwerke, Ordner und Dateien 99
Exkurs: Was sind Objekte und Klassen? 103
D.3.1. lokale Datenträger 104
Exkurs: Speicher-Kapazität 105
D.3.1.2. moderne Datenträger 107
Festplatten 107
CD / DVD / Blue-ray 108
USB-Stick's 108
Speicher-Karten 108

die Welt der Kunden-Karten ... 108
D.3.2. Netzwerk-Laufwerke ... 109
D.3.3. die Cloud (Speichern in der Wolke) ... 109
D.3.4. Sicheres Speichern von Daten ... 110
D.3.y. Schutz vor Daten-Verlusten ... 112
D.4. Arbeit mit der Zwischenablage ... 112
D.5. Wem gehören die Daten? ... 112
Datenspuren ... 113
"Daten sammeln" als Geschäfts-Modell ... 115
Vermeiden von Datenspuren ... 116
anonymes Surfen ... 116
Browser für anonymeres Surfen einrichten / nutzen ... 117
Datenschutz – welche Daten müssen geschützt werden? ... 122
Definition(en): Datenschutz ... 122
Gesetze: Datenschutz (augewählte Auszüge) ... 123
Definition(en): persönliche Daten ... 123
Festlegung allgemeingültiger Regeln ... 124
Sicherheitstipps für das Internet ... 126

Jahrgangsstufe 6 ... 128

4. Entscheidungen treffen und Spiele gestalten ... 128
4.1. das EVA-Prinzip ... 129
4.2. Erstellen kleiner Spiele ... 130
6.x.1.1. Programmieren mit Scratch ... 131
6.x.1.1.0. Wege zu Scratch ... 131
Scatch in der modernen Version 3 ... 133
6.x.1.1.1. fortgeschrittenes Programmieren mit Scratch ... 142
6.x.1.2. Programmieren mit Snap! ... 143
6.x.1.3. Programmieren mit NEPO (open Roberta) ... 160
6.x.1.3.0. Wege zu "Roberta" ... 160
6.x.1.3.1.fortgeschrittenes Programmieren mit NEPO ... 169
6.x.1.4. Programmieren des Calliope mini mit MakeCode ... 170
6.x.1.5. Programmieren von Arduino's und kompatiblen / abgeleiteten IoT-Bausteine ... 172
6.x.1.5.0. Wege zur Programmier-Umgebungen ... 172

5. In der vernetzten Welt kommunizieren ... 176
5.x. ??? ... 178
soziale Netzwerke ... 179
unzugängliches Internet ... 180
TOR-Browser ... 180
Exkurs: Wie arbeitet das TOR-Netzwerk? ... 181

6. Texte strukturieren und gestalten ... 183
Exkurs: Welches Textverarbeitungs-Programm soll ich benutzen? ... 185
6.1. Text-Erfassung ... 188
6.1.1. Zwei Lösch-Tasten – Wozu braucht man die denn? ... 188
6.2. Text-Bearbeitung ... 189
6.2.1. Auswählen und Markieren von Texten ... 189
6.2.2. Zeichen-Formatierung ... 189
6.2.3. Absatz-Formatierung ... 190
6.2.4. Seiten-Formatierung ... 191
6.2.5. Texte professionell formatieren ... 192
Arbeits-Regeln: ... 195
Objekte und Strukturen in der Textverarbeitung ... 198
4.3. (gesellschaftliche / historische Aspekte) ... 204

Jahrgangsstufe 7 208

7. Spiele entwickeln und multimedial dokumentieren 208
7.x. Multimedia-Spiele erstellen 209
6.x.1.6. Programmieren des eigenen Smartphone's 209
6.x.1.6.0. Wege zum MIT App Inventor 210
Vorarbeit: Anlegen eines google-Konto's 210
Anmeldung mit google-Konto 213
Anmeldung ohne google-Mail-Konto 215
Sprache ändern 215
App-Komponenten und Bedienelemente 216
Beispiel-Projekt: Bild-Ton-App 217
Projekt anlegen 217
Zusammenstellen der Anzeige- und Bedien-Elemente 218
Programmierung der Funktionalität 219
Veröffentlichen und Testen der App 220
Erweitern und Verbessern der App 221
Beispiel-Projekt: Glas-Kugel-App 222
Projekt anlegen 222
Design einrichten / Arrangement der Bedien-Elemente 223
Programmierung der Funktionalität 223
Veröffentlichen und Testen 224
Beispiel-Projekt: Mal-App 225
Projekt anlegen 225
Programmierung der Funktionalität 225
Erweitern und Verbessern der App 227
Veröffentlichen und Testen 229
Beispiel-Projekt: Hund-jagt-Katze-Spiel 230
Projekt anlegen 230
Programmierung der Funktionalität 230
Erweitern und Verbessern der App 232
Veröffentlichen und Testen 233
7.x. multimediale Präsentation 234

E. Probleme untersuchen und durch Programmierung lösen 235
E.x. vom Problem zum Programm 235
E.x.y. App-Entwicklung mit dem App Inventor 236

8. Sicher kommunizieren 238
8.0. Grundlagen 239
8.0.1. Kommunikation 239
Definition(en): Kommunikation 239
8.0.1.1. ein einfaches Kommunikations-Modell 241
8.0.1.2. Arten / Einteilung der Kommunikation: 244
8.0.2. Kooperation 247
8.1. Codierung und Chiffrierung 248
8.1.1. Codierung im Alltag 248
8.1.2. Codierung von Mengen und Zahlen 249
Definition(en): Kodierung 250
Definition(en): Dekodierung 250
8.1.3. Codierung von Buchstaben und anderen alphanummerischen Zeichen 252
8.1.4. Codierung zur Erzeugung "geheimer" Nachrichten 256
BRAILLE-Schrift / Tastschrift für sehschwache und blinde Personen 260
der Freimaurer-Code 261
8.2. kleine Geschichte der Verschlüsselung (Chiffrierung) mit praktischen Beispielen 263
8.2.0. Grundlagen, Begriffe, … 263

Definition(en): Chiffrierung ... 263
Definition(en): Dechiffrierung ... 263
Zeichen ersetzen → Substitution ... 264
Zeichen(-Reihen)folge verändern → Transposition ... 268
Tabellen-Verschlüsselung ... 272
Zeichenfolgen erweitern → Expansion ... 273
Chiffrieren und Dechiffrieren mit verschiedenen Schlüsseln ... 275
8.2.1. die Anfänge der geheimen Botschaften ... 277
Zick-Zack-Methode ... 278
Schablonen nach FLEIßNER ... 280
Atbash 281
Weiterentwicklungen der ATBaSch-Methode ... 283
CÄSAR-Chiffre ... 284
Bedienung der Multi-Kodier-Scheibe ... 287
Weiterentwicklungen der CÄSAR-Chiffrierung ... 288
Verwendung eines Schlüsselwortes ... 288
Verwendung von Schlüsselwort und Verschiebung ... 288
Verwendung eines zufällig geordneten Zeichensatz ... 288
Verschlüsselung mit mehreren Schlüssel-Alphabeten ... 290
Enigma – eine Legende und ein Mysterium ... 294
Das Geheimnis sicherer Passwörter ... 295
Steganographie – die Kunst des Versteckens ... 296
Verschlüsselung und Verstecken mit Snap! ... 300
8.3. Nachrichten verstecken ... 302

9. Daten erfassen und darstellen ... 303
Objekte und Strukturen in der Tabellen-Kalkulation ... 306

F. Textverarbeitung für den Anwendungsalltag ... 312

Jahrgangsstufe 8 ... 313

10. Sensorgesteuerte Anwendungen entwickeln ... 313
10.x. Umwelt – Computer – Umwelt ... 314
10.x. Internet of Thinks, Smarthome und kein Ende ... 314

11. Daten automatisiert auswerten ... 315
14.x. Sortieren im Alltag ... 316
14.x. Sortier-Algorithmen ... 316
out-of-place-Verfahren ... 316
Beispiel: Select-Sort ... 316
inplace-Sortierung ... 317
Beispiel: Bubble-Sort ... 317
Beispiel: Quick-Sort ... 318
14.x. Implementieren von Sortier-Algorithmen ... 318

12. Suchmaschinen verstehen ... 321
x.y.z. Suchen und Finden im Internet ... 322
Suchen und Finden (ohne Suchmaschinen) ... 322
Exkurs: Aufbau einer HTML-Datei ... 323
Suchen und Finden mit Suchmaschinen ... 324
Festlegen von Rang-Ordnungen ... 331
Durchsuchen von Ordner-Strukturen ... 334
Experten-Suche in Suchmaschinen ... 336
Objekte und Strukturen in Hypertexten ... 337
5.x.y. Erstellen von HTML-Texten ... 337
5.x.y.z. mozilla Thimble ... 337
5.x.y.z. JSbin ... 338

12.x. Internet-Suchmaschinen – gesellschaftliche Aspekte ... 341

G. Netzwerke und kooperatives Arbeiten ... 342

H. kritischer Umgang mit Medien ... 343
H.x. Medien in unserer täglichen Welt ... 343
H.x. Medien in der Schule ... 343
H.x. kritische Analyse von Medien ... 344
Fakten prüfen im Netz ... 344
Schritt / Tipp 1: Stopp ... 344
Schritt / Tipp 2: Quellen überprüfen ... 344
Schritt / Tipp 3: weitere Informationen / Quellen recherchieren ... 344
Schritt / Tipp 4: Zurück zum Original ... 345

I. Textverarbeitung trifft Tabellenkalkulation ... 346

Jahrgangsstufe 9 ... 348

13. Sensordaten erfassen und auswerten ... 348

14. Publikationen und Präsentationen getalten ... 350

15. Prinzipien der Datenübertragung verstehen ... 351
15.x. Daten zuverlässig und korrekt übertragen ... 351
15.x. Wie funktionieren Netzwerke? ... 352
15.x. Datenübertragung mit Sicherheit ... 354

J. Medien analysieren und produzieren ... 356
J.x. Medien-Komplex: Internet ... 356
J.x. Medien-Komplex: Fernsehen ... 356
J.x. Medien-Komplex: Werbung ... 356
J.x. Medien-Komplex: Information und Nachrichten ... 356
J.x. Medien-Komplex: Kommunikation und Globalisierung ... 356
J.x. Medien-Komplex: Unterhaltung ... 357
J.x. Medien-Thema: Gewalt-Darstellung ... 357
J.x. Medien-Thema: Simulation von Medien ... 357
J.x. Nutzungs-Verhalten ... 357
J.x.1. Online-Spiel-Sucht ... 358
K.x. Textverarbeitung mit großen Dokumenten ... 359
K.x.y. große Dokumente mit Libre Office WRITER ... 359
K.x.y. große Dokumente mit microsoft WORD ... 359
K.x.y. große Dokumente mit Libre Office TEXTMAKER (FREE) ... 359

Jahrgangsstufe 10 ... 361

16. Sprachen und Sprachkonzepte ... 361
16.x. natürliche und künstliche Sprachen ... 362
16.x. Programmier-Sprachen ... 362

17. Problemlösen durch Programmieren ... 363
17.x. Projekt: Informationssystem ... 363
17.x. Übersetzungs-Konzepte für Programmier-Sprachen in Maschinen-Sprache ... 364
17.x.y. Entwicklung einfacher Programme mittels Text-Anweisungen ... 367
17.x.y.z. Programme mit Python entwickeln ... 367
17.x.y.z. Programme mit BASIC entwickeln ... 367
17.x.y.z. Programme mit JavaScript entwickeln ... 367
17.x.y.z. Programme mit Ruby entwickeln ... 368

L. Daten-Strukturen 369
(Warte-)Schlange FIFO 369
Feld Array Vector 370
Keller Stapel Stack FILO 370
Baum Tree ??? 370
Binär-Baum 371
ausgewogene Bäume 371
Netz Graph 371

18. Grundlagen der Digitalisierung 372
18.x. Digitalisierung in unserem Leben 373
18.x. Aufbau eines VON-NEUMANN-Rechner's 373
18.x.y. Alternativ-Konzept: die Harvard-Architektur 377
Exkurs: der Rechner Mark I von IBM 377
18.x. Simulation eines VON-NEUMANN-Rechner mit MOPS 380
18.x.y. Weg zu MOPS / Installation 380
18.x.y. Kurzvorstellung MOPS 381
Arbeitsblatt für MOPS-Spiel und die Dokumentation von Programmier-Aufgaben und Simulationen 383

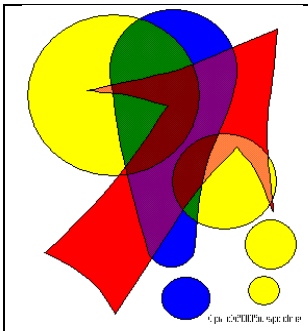
ohne Jahrgangsstufe 385

27. Alles, was noch keinen Platz gefunden hat 386
x.y. Digitales Storytelling 387
Erstellen eines Video's mit Adobe Spark Video 389
Erstellen einer Info-Grafik mit Adobe Spark Post 390
Erstellen einer Info-Grafik mit Adobe Spark Page 391
x.y. diverse Programmier-Tools 392
x.y.z. Code.org 392
x.y.z.1. Flappy Bird 394
x.y.z.2. Sprite Lab 394
x.y.z.3. Artist 394
spielerische Einführung des Variablen-Konzepts und Daten-Typen 395
x.y.z.4. App-Lab -- App-Programmierung mit JavaScript 397
x.y.z. in vielen Sprachen unterwegs - Programmieren mit repl.it 398
x.y.z.1. Programmierung in QBasis 401
x.y.z.2. Programmierung in Python 401
x.y.z.3. Programmierung in JavaScript 402
x.y.z.4. Programmierung in PHP 402
x.y.z.5. Programmierung in Ruby 402
x.y.z.4. Programmierung in Java 403
x.y.z.1. Programmierung in 403
x.y.z. thematische Link-Listen 404
x.y.z.a. Material / Material-Listen 404
x.y.z.b. Tools 404

Anhang: Code- und Chiffre-Tabellen 405
Freimaurer-Code 405
-Code 405
Vorlage 405

Literatur und Quellen: 407

Vorwort



Hinweise zu Themen / Überschriften / Skript-Struktur / …:
Derzeit ist die Struktur des neuen Lehrplans noch in der Erprobung und Diskussion. In Folge dessen sind die Kapitel teilweise intern / teilweise am Lehrplan angelehnt betitelt worden. Eine entgültige Entscheidung meinerseits ist noch nicht gefallen. Inhaltlich sind viele Themen eher experimentell zu sehen. Die Zuordnung zu den Klassenstufen muss ev. auch noch konkretisiert werden.
Sehen Sie dieses Skript zuersteinmal als Skizzenblock mit Skizzen unterscheidlicher Fertigstellungs-Grade.
Wenn sich der Lehrplan konkretisiert, dann werden die Themen wahrscheinlich neu geordnet.

In den Phasen der Entwicklung und Überarbeitung der einzelnen Themen sind die Beziehungen zu den Elementen des Rahmenlehrplans (z.Z. in Form des Curricularen Konzeptpapieres /*/) noch im Skript enthalten und farblich abgesetzt. Die Farbkennung orientiert sich an den verwendeten Farben in Konzeptpapier.

sicher

reaktivierbar

exemplarisch

für die Jahrgangs-Ziele / -Kompetenzen verwenden wir hier einen hellgrauen Hintergrund

allgemein

Die Texte wurden wörtlich übernommen und werden hier als Zitate verstanden!

Mitlerweile ist diese Strukturierung überholt. In der aktuell verfügbaren Anhörungsfassung zum Rahmen-Lehrplan "Informatik und Medienbildung" sind (neue) "Verbindliche Ziele und Inhalte" festgelegt worden. Diese werden ab der Version 0.1 dieses Skriptes so dargestellt:

verbindliche Ziele und Inhalte

Die Hinweise und Anregungen entsprechend etwas dezenter unterlegt:

Hinweise und Anregungen

Alle Texte sind wörtlich aus der Anhörungsfassung übernommen und werden hier als Zitate verstanden. Die Sequenz "Schülerinnen und Schüler" wird durch "SuS" ersetzt.
In späteren Versionen werden alle diese Hinweise / Bezüge entfernt. Derzeit dienen sie der Orientierung und als Diskussions-Basis über dieses Skript.

Schrittfolge

- **Benutzen**
- **Analysieren**
- **Gestalten**
- **Verankern**

Q: MBWK, 2008, S.2

klassisches "Use-Modify-Create"-Konzept

- **Benutzen (, use)** — einfaches, unreflektiertes Anwenden von Tools, Medien, …
- **Verändern (, modify)** — Anpassen vorhandener Tools / Medien / … für eigene Zwecke
- **Erstellen (, create)** — Nachproduzieren von Tools / Medien / … auch für andere Nutzer kreativer Umgang mit Tools und Medien

erweitert aus dem Dreiklang: "Use – Modify – Create"

(BEVEN-) Schritt- oder Stufenfolge

- **Benutzen (, use)** — einfaches, unreflektiertes – z.T. intuitives – Anwenden von Tools, Medien, …
- **Erkunden (, explore)** — Beobachten, Analysieren, Verstehen der Zusammenhänge um den Sachgegenstand, die Tools / Medien / … hinter die Kulissen schauen
- **Verändern (, modify)** — Anpassen vorhandener Tools / Medien / … für eigene Zwecke
- **Entwickeln (, develope)** — Nachproduzieren von Tools / Medien / … auch für andere Nutzer kreativer Umgang mit Tools und Medien
- **Neuentdecken (, discovery)** — Erstellen neuartiger Tools / Medien / ++

Einige der angegebenen Links zu externen Inhalten enthalten eine Bewertung mit Sternen (☆). Hierbei handelt es sich um die **persönliche Meinung des Autors!** Eine Nicht-Bewertung ist keine Wertung. Nur bei Seiten, die mir besonders positiv aufgefallen sind, habe ich eine Bewertung vergeben. Das beste Ergebnis habe ich mit 5 Sternen (☆☆☆☆☆) angesetzt.

Einleitung

Jahrgangsstufe 5

0. Warum Informatik und Medienbildung?

Problem-Fragen für Selbstorganisiertes Lernen
Welche Bedeutung haben Computer, Tablet's, Laptop's und Smartphone's in unserem heutigen Leben? Kann man ohne sie auskommen? Wie sind die Geräte aufgebaut? Gibt es gemeinsame / allgemeine Bauelemente? Was ist das EVA-Prinzip? Warum muss ich mich bei vielen Geräten immer anmelden? Ist es vernünftig, Anmeldungen zu umgehen? Warum muss ich mich ausloggen?

Der Computer und seine kleinen Verwandten – wie Laptop's, Tablet's und Smartphone's – haben unsere Welt stark verändert. Wie sehr sie unserer Leben beeinflussen, merken wir besonders, wenn wir auf sie verzichten müssen.

Aufgaben:

1. ***Wählt in der Klasse einen Tag (24 Stunden) aus, an dem Ihr mal ohne Computer, Laptop, Tablet und / oder Smartphone auskommen wollt! Die Benutzung eigener Geräte ist unbedingt zu vermeiden – also Geräte ausschalten und wegstellen. Notiert, wie oft Ihr in den 24 Stunden an anderer Stelle (z.B. Fahrkarten-Automat usw. usf.) trotzdem einen Computer benutzen musstet!***
2. ***Stellt Eure Beobachtungs-Notizen vor! Wer kam mit den wenigsten Computer-Nutzungen aus?***

Um Deine Geräte vernünftig benutzen zu können, hast Du schon viele Sachen so ganz nebenbei gelernt. Um die ganze Leistung eines Gerätes ausnutzen zu können, muss man et-

was mehr über den Aufbau und die Arbeitsweise wissen. Besonders wichtig ist es auch die Gefahren kennen zu lernen, mit denen man nun konfrontiert ist.
Das kennen wir aus dem Strßenverkehr. Es ist toll die Straßen mit seinem Fahrrad oder einem ähnlichen Gefährt zu benutzen, aber man muss auch die Regeln und Gefahren kennen.

Aufgaben:

1. *Nenne drei Vorschriften aus dem Straßenverkehr, an die man sich halten muss!*
2. *Von welchen Gefahren bei der Computernutzung hast Du schon gehört? Tragt sie an der Tafel zusammen!*
3.

Teilgebiete der Informatik

• **technische Informatik**	Hardware-Komponenten	OS, Sek.I
	Mikroprogrammierung	
	Rechner-Organisation	
	Rechner-Architektur	Sek.II
	Schnittstellen-Technik	
	Rechner-Netze	Sek.II
• **praktische Informatik**	Algorithmen	OS, Sek.I+II
	Daten-Strukturen	OS, Sek.I+II
	Programmier-Methoden	
	Programmier-Sprachen	Sek.I+II
	Betriebssysteme	
	Software-Technik	
	Mensch-Maschine-Kommunikation	
• **theoretische Informatik**	Automaten-Theorie	Sek.II
	Theorie der formalen Sprachen	Sek.II
	Berechenbarkeits-Theorie	
	Komplexitäts-Theorie	
	Algorithmen-Analyse	
	Theorie der Programmierung	
	automatische Programm-Erstellung	
• **angewandte Informatik**	Informations-Systeme	
	Computer-Grafik	Sek.I
	künstliche Intelligenz	
	digitale Signal-Verarbeitung	Sek.I
	Modellierung und Simulation	
	Text-Verarbeitung	Sek.I
	Tabellen-Kalkulation	Sek.I
	Präsentation	Sek.I
	Webseiten-Erstellung	
	weitere spezifische Anwendungen / Apps	

1. Programmieren? Kinderleicht!

Algorithmische Grundbausteine verwenden

- *sequentielle Abläufe interpretieren und zielgerichtet verwenden*
- *Wiederholungsstrukturen mit einer festen Anzahl von Wiederholungen interpretieren und zielgerichtet verwenden*

Eine Sequenz ist eine Nacheinanderausführung von Anweisungen.
Die SuS ersetzen eine Abfolge gleicher Sequenzen durch eine Wiederholungsstruktur mit einer festen Anzahl von Wiederholungen.

Algorithmen verstehen

- *die Idee eines gegebenen oder selbsterstellten Algorithmus beschreiben*
- *die Korrektheit eines Algorithmus praktisch testen*

Unter einem Algorithmus verstehen die SuS eine Handlungsvorschrift mit einer eindeutigen und endlichen Abfolge von elementaren Anweisungen.

Daten verarbeiten

- *den Einfluss von Parameterwerten auf den Programmablauf erkennen und zielgerichtet verändern*

Vorschlag zur inhaltlichen Vertiefung:
Über den Vergleich der Anzahl der Anweisungsblöcke ist eine Einschätzung der Effizienz möglich.

Problem-Fragen für Selbstorganisiertes Lernen

Kann ein Computer alles verstehen, was wir ihm sagen?

Was ist ein Algorithmus?
Wie kann man Algorithmen beschreiben / notieren?

Wer legt fest, was Computer machen?

Können Algorithmen auch so gestaltet werden, dass sie verändliche Aufgaben lösen können?
Wie kann man abschätzen, ob ein Algorithmus gut / besser ist?

Computer als Maschinen, die ohne klare Anweisungen nichts machen

Computer können Befehle auch nicht interpretieren
Wird einem Roboter gesagt er soll vorwärts fahren, dann tut er das konsequent. Auch eine Wand ändert daran nichts. Der Roboter versucht immer weiter vorwärts zu fahren. Das ändert sich erst, wenn die Antriebe kaputt, gehen die Energie zu ende geht oder jemand die Befehls-Abarbeitung abbricht.

Merksätze: Algorithmus
Ein Algorithmus ist eine eindeutige Arbeitsvorschrift, die zur Lösung einer Aufgabe führt.

Programmierung als die Tätigkeit für den Computer Befehle zusammenzustellen

Ergebnis ist ein sogenannter Algorithmus. Ein Algorithmus ist eine Befehls-Folge, die irgendwann endet und bei der in jedem Schritt klar ist, welcher Schritt als nächstes gemacht werden soll.

Programmiersprachen sind für die Bereitstellung von Befehls-Folgen für die Computer verantwortlich. Dazu benutzt man Sprachen, die von Menschen gut verstanden werden und ihm die Arbeit vereinfachen. Die Programmiersprachen sind so festgelegt, dass aus den Menschen-verständlichen Anweisungen durch eine eindeutige Übersetzung eine Anweisungs-Folge für den Computer wird.
Da Programme in einer Programmiersprache direkt und ohne Interpretation in die Maschinen-Befehle übertragen werden, müssen die Programme unbedingt Fehler-frei sein. Computer können keine Fehler in Befehls-Folgen erkennen. Sie tun das, was der Programmierer ihnen in seiner Programmiersprache mitgeteilt hat und das Übersetzer-Programm daraus gemacht hat.

Die meisten Programmiersprachen benutzen Kommando-Texte zur Formulierung von Programmen. Dabei dürfen keine Schreib-Fehler auftreten. Diese können die Übersetzer-Programme aber auch noch erkennen. Programmierer sprechen hier von Syntax-Fehlern. Anders sieht das bei Fehler aus, die aufgrund falsch kombinierter Befehle entstehen. Diese kann weder das Übersetzungs-Programm noch der Computer erkennen. Computer sind eben sture Befehls-Empfänger, ohne ein Nachdenken über ihre Arbeit oder ein Gewissen. Inhaltliche Fehler werden Semantik-Fehler genannt.

Syntax-Fehler kann man als Fehler der Grammatik und Rechtschreibung einer Sprache verstehen. Semantik-Fehler sind Fehler im Verständnis oder der Bedeutung der Texte.

Aufgaben:

1.
2. *Übernehme die nachfolgenden Sätze in Deinen Hefter! Streiche Synthax-Fehler mit einem roten Stift und Semantik-Fehler mit einem blauen an!*

   a) Ein Haus hat ein Dach.
   b) Das Haus hat eine Türen und vier Fenster.
   c) Ein Maus frisst vieler Katzzzen.
   d) In Katzen leben viele Mäuse.

3. *Welche der gefundenen Fehler in Aufgabe 2 lassen sich sicher finden und berichtigen? Begründe Deine Meinung!*

Graphische Programmiersprachen verringern die Probleme mit Computer-Befehlen. Die Befehle sind als Blöcke dargestellt und brauchen nicht als Text eigegeben werden. Dadurch sind deutlich weniger Syntax-Fehler möglich. Die Programmierer müssen nicht mehr die ganzen Befehle einer Programmier-Sprache lernen, sie können die Blöcke aus einem menü auswählen. Durch bestimmte Formen wird erreicht, dass nur noch passende Befehls-Teile kombiniert werden. So sinkt die Fehler-Häufigkeit weiter.
Um Semantik-Fehler zu vermeiden, hilft es nur, sich die Aufgabe für den Computer gut zu überlegen und das Programm gut zu planen. Das ist eigentlich der schwierigste teil der Programmierung.
Block-orientierte Programmiersprachen, wie MakeBlock, Snap! oder Scratch sind besonders für Anfänger geeignet. Das Lernen der Programmiersprache als solche tritt in den Hintergrund. Fast alles dreht sich nun nur noch um das Lösen des Problems, bei dem uns der Computer (od. Roboter) helfen soll.

Blöcke der graphischen Programmiersprachen stehen für die Elementar-Befehle oder Befehls-Teile.

Da der Computer also nicht eigenständig Probleme aus unserem Leben interpetieren kann, müssen wir das im Vorfeld machen. Wir müssen Programme schreiben oder zusammenstellen, die dem Computer die Arbeitsschritte vorgeben.
Wer heute nicht versteht, wie die Befehls-Vorgabe an Computer funktioniert, wird schnell den ausgeliefert sein, die Programme für ihre Zwecke nutzen und vielleicht noch so tun, als wäre der Computer an irgendetwas Schuld. Es sind immer die Programmierer und die beauftragenden Personen dafür verantwortlich, was ein Computer macht. Wenn ein Computer ungewöhnliche oder problematische Ausgaben macht, dann sind es entweder schlecht getestete Programme oder absichtlich gemachte "Fehler" in Programmen, die Ursache für die Computer-Leistung sind. Und dafür sind immer Menschen verantwortlich. Die Ausrede "Das hat der Computer aber so berechnet / ausgegeben." kann so nicht allgemein gelten gelassen werden.

Merksätze: Sequenz
Eine Sequenz ist eine Abfolge direkt hintereinander ausgeführter (Einzel-)Anweisungen.
Eine Reihe aufeinander folgender Arbeitsschritte / Sachverhalte nennt man Sequenz.

1.1. Algorithmen in unserem Leben

Bauanleitungen und Koch- bzw. Back-Rezepte als alltägliche Arbeitsvorschriften

(freies) Setzen einer Spielfigur beim "Mensch-ärgere-Dich-nicht!"
das erneute Würfeln bei einer Sechs und das Rausschmeißen ignorieren wir zuerst einmal

einfaches Setzen einer Spielfigur

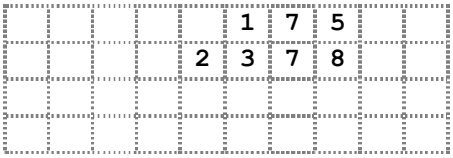
1. Würfele einmal
2. Nehme die Spielfigur hoch
3. Beginne mit der Zahl "1"
4. Wiederhole die nächsten Schritte so oft, wie Du Punkte auf dem Würfel siehst
   - 4a. Sage die Zahl
   - 4b. Setze die Figur (andeutungsweise) ein Feld nach vorne
   - 4c. Erhöhe die Zahl um 1
5. Setze die Spielfigur ab
6. Gibt den Würfel weiter

Berechnungen in der Mathematik sind ganz typische Algorithmen. Besonders deutlich wird das bei den schriftlichen Additionen, Subtraktionen, Multiplikationen und Divisionen. Hält man die Vorschriften genau ein, dann bekommt man immer eine Lösung.

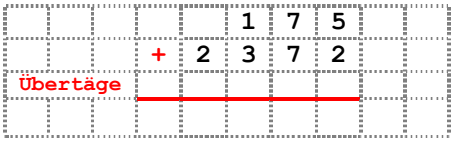
Addition von zwei (natürlichen) Zahlen

z.B.: `175 + 2378`

1. Schreibe beide Zahlen so untereinander, dass sie Ziffern-genau mit gleicher Wertigkeit (Einer, Zehner, …) angeordnet sind

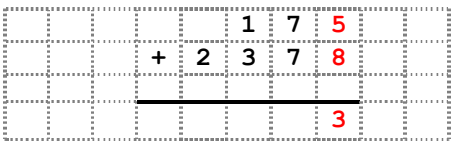


2. Unterstreiche die unterste Zahl und ergänze ein Plus-Zeichen vor der untersten Zahl (lasse etwas Platz für Überträge)

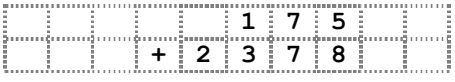


3. Beginne bei der kleinsten Stelle (Einer)

4. Addiere die übereinanderstehenden Ziffern; notiere die letzte Stelle (Einer) in die Position unter dem Strich

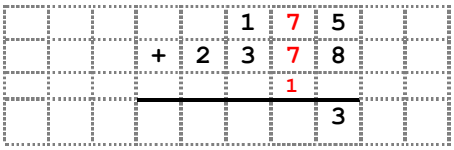


5. WENN ein Übertrag entstanden ist (also Zehner

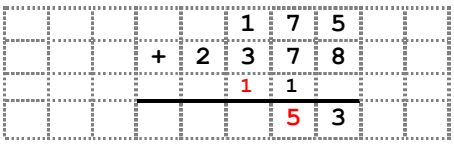


usw.), DANN notiere diese stellenwertgenau in die Übertragszeile

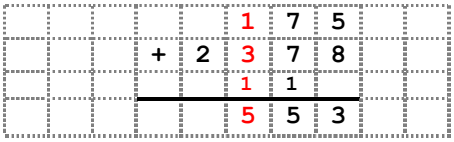
6. Wechsle nun zur nächsthöheren Wertstelle und führe die Schritte ab 4. aus, SOLANGE BIS keine Ziffern (einschließlich der Überträge) mehr zu addieren sind



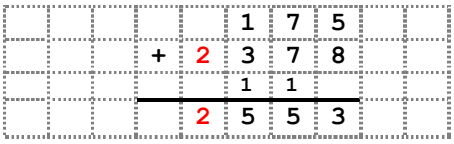
Notieren des Summe (einschließlich Übertrag) für die Zehner



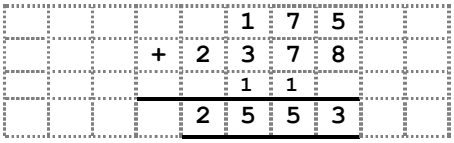
Berechnen der Summe für die Hunderter



Berechnen der Summe für die Tausender



Fertig!, weil keine weiteren / höheren Stellenwerte mehr vorhanden sind



7. Unterstreiche die neu gebildete Ziffernfolge als Ergebnis doppelt

Aufgaben:

1.

2.

3.

typische Arbeitsfolgen und Absprachen (z.B.: Wer geht morgens als erster ins Bad? Wer folgt dann? usw. usf.) sind im Prinzip Algorithmen, um ein morgentliches Chaos und unnötige Wartezeiten zu vermeiden. Das ist dann auch das Ziel dieses "Algorithmus". Wir wollen damit ein Ziel erreichen und es soll auch zuverlässig klappen.

Merksätze: Algorithmus
Ein Algorithmus ist eine Arbeitsvorschrift, mit deren Hilfe ein Problem / eine Aufgabe gelöst werden kann.

Algorithmen helfen uns, auch Dinge zu tun, die wir vorher noch nie gemacht haben.

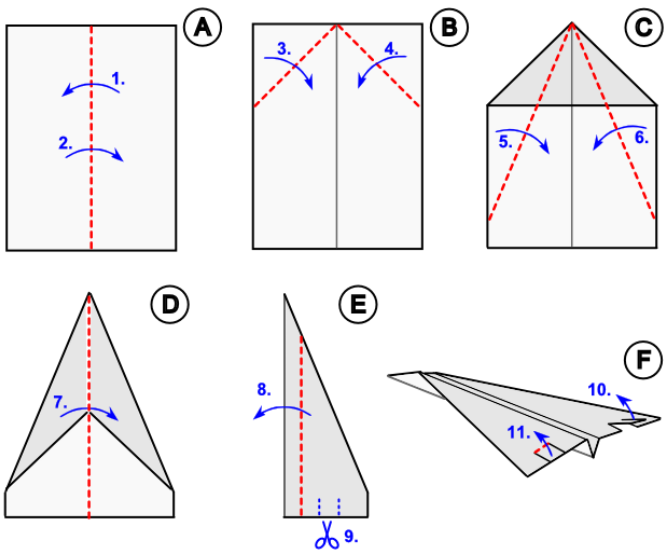
(c,p) 2015 – 2020 lsp: dre

Dazu sind Arbeitsvorschriften in irgendeiner Form darsgestellt. Bei Koch- und Back-Rezepten sind dies meist kurze stichpunktartige Anweisungen, wie z.B. das Eier-Kochen.

Eier kochen

1. Wähle die Anzahl der zu kochenden Eier aus
2. Wähle einen passenden Kochtopf aus
3. Fülle (warmes) Wasser in den Topf (so dass die Eier bedeckt wären)
4. Erwärme das Wasser bis zum Kochpunkt
5. Lege die Eier mit einem Löffel in das kochende Wasser
6. Lege die Kochzeit in Abhängigkeit von der gewünschten Härte fest (4:30 [min:s] für weiche Eier; 5:00 mittlere Härte (Frühstücks-Ei); 6:00 hartes Ei)
7. Stelle den Wecker auf die Kochzeit ein und starte ihn
8. Lasse das Wasser die gesamte Kochzeit gerade so kochen
9. Nehme den Topf vom Herd
10. Gieße das Kochwasser ab
11. Lasse ev. mehrfach kaltes Wasser über die Eier laufen (Abschrecken)
12. STOPP

Selbst wer noch nie Eier gekocht hat, bekommt das mit dieser Vorschrift hin.
Viele Algorithmen sind auch in Textform verfügbar.
Aus dem berühmten schwedischen Möbel-Häusern kennen wir die Aufbau-Anleitungen für die verschiedenen Möbel. Die bebilderten Anleitungen zeigen schrittweise, wie man aus der Packung ein Möbelstück macht.
Das gleiche Prinzip wird durch die nebenstehende Falt-Anweisung zum Bau eines Papierfliegers genutzt. Schrit für Schritt können wir so aus einem Batt Papier einen funktions-fähigen Flieger bauen.

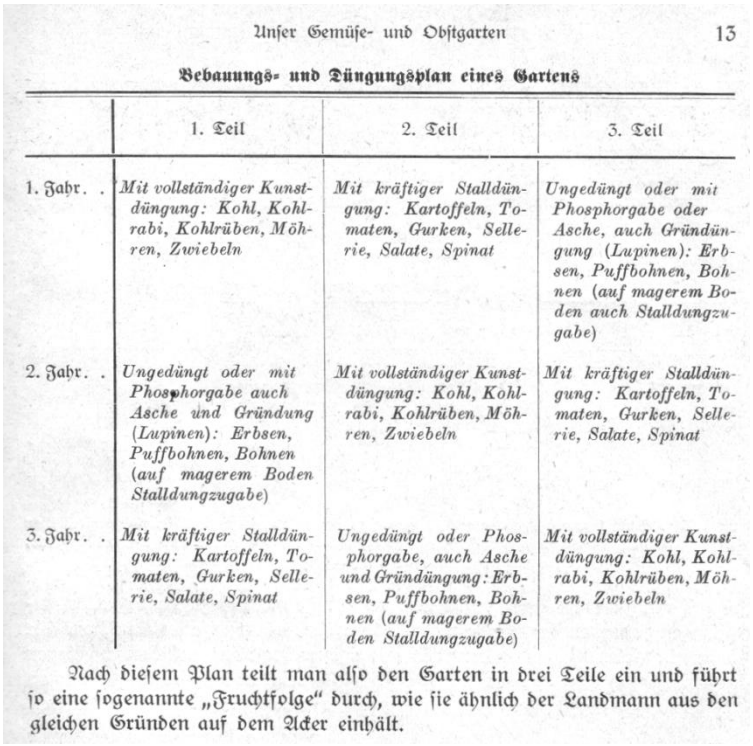


Aufgaben:

1. *Baue einen Papierflieger nach der Falt-Anweisung! Prüfe, ob der Flieger auch gut funktioniert! Wessen Flieger schafft die größte Strecke? Wessen Flieger erreicht ein Ziel am Genauesten?*

2. *Wer kennt andere Papierflieger! Recherchiert in Büchern oder im Internet! Baut die Vorschläge nach und lasst sie im Wettbewerb gegeneinander fliegen!*
3. *Besorgt Euch die Faltanleitung (Origami) für einen Frosch! Faltet jeweils einen Frosch und lasst sie um die Wette springen! Wessen Frosch springt am Weitesten? Ergründe, woran das vielleicht liegen könnte!*
4.

Wieder andere Algorithmen kommen in Form von Tabellen vor. Hier ein Beispiel aus einem alten Büchlein.



Unser Gemüse- und Obstgarten 13

Bebauungs- und Düngungsplan eines Gartens

	1. Teil	2. Teil	3. Teil
1. Jahr. .	*Mit vollständiger Kunstdüngung: Kohl, Kohlrabi, Kohlrüben, Möhren, Zwiebeln*	*Mit kräftiger Stalldüngung: Kartoffeln, Tomaten, Gurken, Sellerie, Salate, Spinat*	*Ungedüngt oder mit Phosphorgabe oder Asche, auch Gründüngung (Lupinen): Erbsen, Puffbohnen, Bohnen (auf magerem Boden auch Stalldungzugabe)*
2. Jahr. .	*Ungedüngt oder mit Phosphorgabe auch Asche und Gründung (Lupinen): Erbsen, Puffbohnen, Bohnen (auf magerem Boden Stalldungzugabe)*	*Mit vollständiger Kunstdüngung: Kohl, Kohlrabi, Kohlrüben, Möhren, Zwiebeln*	*Mit kräftiger Stalldüngung: Kartoffeln, Tomaten, Gurken, Sellerie, Salate, Spinat*
3. Jahr. .	*Mit kräftiger Stalldüngung: Kartoffeln, Tomaten, Gurken, Sellerie, Salate, Spinat*	*Ungedüngt oder Phosphorgabe, auch Asche und Gründüngung: Erbsen, Puffbohnen, Bohnen (auf magerem Boden Stalldungzugabe)*	*Mit vollständiger Kunstdüngung: Kohl, Kohlrabi, Kohlrüben, Möhren, Zwiebeln*

Nach diesem Plan teilt man also den Garten in drei Teile ein und führt so eine sogenannte „Fruchtfolge" durch, wie sie ähnlich der Landmann aus den gleichen Gründen auf dem Acker einhält.

Q: EDDELBÜTTEL, BURMEISTER: Der junge Naturforscher – ein Hilfsbuch der Biologie auf arbeitsunterrichtlicher und heimatlicher Grundlage; Berlin: Verl v. Paul Parey.-Teil IV S. 13 (wahrscheinlich aus dem Jahr 1931)

Aufgaben:

1. *Suche zu Hause nach einer möglichst orginellen / besonderen Arbeitsvorschrift (altes Rezept, Aufbauanleitung, fremdsprachige Bedienungsanleitung, …)! Bringe diese zum nächsten Unterricht mit! Erkundige Dich bei den Eigentümern der Arbeitsvorschrift darüber, ob es zu dieser eine besondere Geschichte gibt!*
2.
3.

für die gehobene Anspruchsebene:

4. *Recherchiere zu den Autoren des Büchleins "Der junge Naturforscher . ein Hilfsbuch … "! Wo haben sie gelebt und gearbeitet? Welche Rolle spielten sie damals?*

Aber warum funktionieren solche Arbeitsanleitungen? Im Prinzip sind wir – genau wie unsere Computer – programmierbare Maschinen. Wir sind eben Bio-Maschinen. Wenn wir eine Arbeitsanleitung bekommen, dann können wir diese schrittweise abarbeiten und ein bestimmtes Ziel erreichen. Ist der Algorithmus gut dargestellt oder beschrieben, dann erreichen wir auch sicher das gewünschte Ergebnis.
Für die Computer müssen die Arbeitsvorschriften in einer für sie nutzbaren Form vorkommen – und das sind die (Computer-)**Programme**. Programme gehören zur Software eines Datenverarbeitungssystem.

Merksätze: Programm
Ein (Computer-)Programm ist eine Sammlung von Arbeitsvorschriften für eine Computer, mit deren Hilfe der Computer eine bestimmte Aufgabe erfüllen kann.

In der Informatik benutzt man verschiedene Arten zur Beschreibung von Algorithmen. Eine recht weit verbreitete Form ist der Pseudo-Code. Man könnte auch Pseudo-programme dazu sagen. Beim Pseudo-Code gibt man immer zuerst eine Tätigkeit an, die dann ev. noch um Zusatz-Informationen ergänzt wird.
Typische Pseudo-Code-Anweisungen sind:

START
STOPP oder **STOP**
warte 10 Sekunden
drehe 90° nach links
gehe_vorwärts 1 Meter

Die groß geschriebenen Anweisung sind solche, die den Ablauf des Algorithmus steuern. Sie werden intern im Programm gebraucht. Zu solchen Anweisungen zählen auch:

WIEDERHOLE 10x
TUE … **BIS** …
WENN … **DANN** …

Man muss sie nicht zwangsläufig groß schreiben, aber die Großschreibung erhöht die Übersichtlichkeit von Programmen und später auch die Umsetzung der Pseudo-Programme in echte Computer-Programme.
Die anderen Tätigkeits-Anweisungen sind frei wählbar. Im Prinzip werden die Tätigkeiten nur dadurch beschränkt, was ein Computer / Roboter von sich aus kann. Hat er einen Lautsprecher, dann können wir auch einen Pseudo-Code-Befehl:

sprich "Hallo ihr zusammen! Was geht ab?"

ausführen. Hat er einen Bildschirm, dann könnte ein Anweisung:

zeige "Hallo, der Computer ist jetzt arbeitsbereit."

lauten. Genau so geht:

schreibe "Hallo, der Computer ist jetzt arbeitsbereit."
drucke "Hallo, der Computer ist jetzt arbeitsbereit."

weil praktisch immer das Selbe gemeint ist. Wir Menschen verstehen den Pseudo-Code noch gut. Für einen echten Computer wäre nur eine einzige Schreibweise zugelassen. Dann auch wahrscheinlich nur in der englisch-sprachigen Übersetzung, z.B.:

print "Hallo, der Computer ist jetzt arbeitsbereit."

Der große Vorteil des Pseudo-Code's ist es gerade, das dieser dann später auf völlig verschiedene Computer übersetzt werden kann. Das machen immer jeweils spezialisierte Programmierer. Sie kennen ihr Computer-System schließlich am Besten.

Aufgaben:

1.
2.
3.

1.2. Programmieren von Einplatinen-Minirechner / Experimentier-Board's

Einplatinen-Minirechner sind quasi sehr kleine Computer. Man nennt sie Mikrocontroller. Sie verfügen fast immer eine Stromversorgung durch ein Mini-USB-Netzteil. Dadurch kann es bei unseren Experimenten keine Gefährdungen durch hohe Spannungen usw. kommen. Trotzdem müssen die Board's vorsichtig behandelt werden. Die Mikroelektronik auf den Board's verträgt nicht jede falsche Schaltung oder falsche Polungen.

Also immer zuerst die Schaltung ohne Betriebs-Spannung aufbauen. Als nächstes wird noch einmal alles geprüft. Das macht am Besten eine zweite Person. Dann kann die Spannung angeschaltet oder das Netzteil in die Steckdose gesteckt werden. In jedem Fall muss man sich an die Arbeits- und Sicherheits-Vorschriften halten.

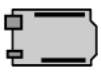
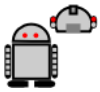
Aufgaben:

1. ***Notiere die allgemeine Schrittfolge für das Arbeiten mit Experimentier-Board's! Was muss unbedingt beachtet werden?***
2. ***?***
3. ***?***

1.2.0. Welches Board haben wir?

Für das richtige Auswählen der Programmier-Umgebung müssen wir unser Board richtig erkennen. Nicht jedes Board kann mit jeder App programmiert werden.
Hier einige

ausgewählte Experimentier-Board's:

- **Calliope mini**
- **micro:bit**
- **OxoCard**
- **Arduino** 
- **BOB3** 
- **IoT-Bausteine**
  (Duino- und Nano-Board's, ESP, …)

-
-

Man sieht schon, dass die einzelnen Board's sehr gut zu unterscheiden sind. Jedes hat seine Stärken und Schwächen. Wir wollen hier das Maximum aus unserem vorliegenden Board herausholen.
In den folgenden Abschnitten werden jetzt die einzelnen Board's etwas ausführlicher vorgestellt. Natürlich muss man sich nur mit dem Board genauer beschäftigen, dass man auch vorliegen hat.
Viele Board's lassen sich mit einer recht universellen App programmieren. Das "Open-Roberta-Lab" (→ lab.open-roberta.org) bietet mit der Programmiersprache NEPO eine Sprache für mehrere Boards an. Sie ist Block-orientiert und deshalb leicht verständlich und intuitiv benutzbar. Keine Angst - durch reines Programmieren kann man ein Board nicht zerstören.
Die Programmierung mit NEPO stellen wir nach den Board's vor. Man muss nämlich bei NEPO auswählen, welches Board man hat. Wenn man das falsche auswählt, dann funktioniert nichts.
Mit NEPO können wir dann später auch verschiedene Roboter-System programmieren. Dazu gehören z.B.:

ausgewählte (programmierbare) Roboter-Systeme

- **LEGO ® WeDo** Einsteiger-Version mit einem kleinen Steuer-Baustein
- **LEGO ® Mindstorms EV3** Version mit einem Steuer-Baustein, der komplexere Aufgaben erfüllen kann
- **Fischer Technik**
- **NXT** älteres LEGO-System
- **Bot'n Roll**
- **NAO** Der Mercedes unter den Robotern im Bildungs-Bereich. Wegen seines hohen Preises eher an Universitäten zu finden.
- **BOB3**
- **mBot**
- **LEGO ® Spike** neueste Version mit neuem Steuer-Baustein
- **Texas Instruments** über Taschenrechner programmierbarer Steuer-Baustein (für einfache Roboter)
-

Auch diese Roboter-Systeme können wir für unsere ersten Programmier-Versuche nutzen.
Für einige Board's gibt es Simulationen im Internet. Sollte mal kein Board zur verfügung stehen oder man will die Programmierung des speziellen Teils nur mal ausprobieren, dann sind Simulationen eine gute Hilfe. Man spart sich dann villeicht einen Fehlkauf.

1.2.0.1. Calliope mini

wichtige Bau-Elemente (Sensoren / Aktoren)

- **5x5 LED-Matrix** — 25 rote LED's, die sich einzeln schalten lassen
- **RGB-LED** — 1 Stück; es ist die Darstellung von rund 1,7 Mio. Farben, auch in verschiedener Helligkeit möglich
- **Lautsprecher /** — zum Erzeugen von Tönen
- **Außen-Kontakte** — 4 Eingänge; dazu (+) und (-)
- **Tasten A und B**
- **Mikrofon** — z.B. zum Erfassen der Umgebungs-Lautstärke
- **Temperatur-Sensor** — zum Messen der Umgebungs-Temperatur
- **Helligkeits-Sensor** — ermittelt Lichtstärke
- **Lage-Sensor** — zur Lage- und Bewegungs-Erkennung
- **Motor-Anschlüsse** — 2 Stück für Robotik-Anwendungen
-

mögliche Programmier-Umgebungen

- **mini editor**
  Calliope-eigener Editor
  ohne Simulator (Programme lassen sich nur auf einem Calliope mini testen)
  Block-orientierte Programmierung
  gut für jüngere Schüler geeignet; einfach aufgebaut
  kein Umblenden zu einem Programm-Text möglich
  → https://miniedit.calliope.cc/

- **MakeCode**
  früher: **PXT**
  Editor von microsoft
  mit Simulator
  Block-orientierte Programmierung
  Umschalten zwischen Block- und Text-Ansicht möglich
  Text-Programmiersprache ist JavaScript
  → https://makecode.calliope.cc/

- **NEPO**
  (Open Roberta Lab)
  Editor vom FRAUENHOFER-Institut
  Block-orientierte Programmierung
  mit Simulator
  auch für andere Board's und Roboter-Systeme geeignet (fließender Übergang zur Roboter-Steuerung möglich)
  → https://lab.open-roberta.org/

-

allgemeiner Ablauf der Programmierung

1. **Erstellen des Programms im Editor** — Zusammenstellen der Blöcke zu einem Programm, dass eine bestimmte Aufgabe erfüllt
2. **ev. Testung im Simulator** — Ausprobieren des Programm's (nicht bei allen

Editoren möglich)

3. Herunterladen	Speichern des Programm auf dem eigenen Rechner (zumeist Downlod-Ordner) es sollte jedes Mal ein neuer Name vergeben werden, damit alte Programme nicht überschrieben werden
4. Kopieren auf den Calliope	Kopieren der heruntergeladenen / gespeicherten HEX-Datei auf den Calliope (z.B. per Drag-and-drop)

Links:
zum Calliope mini:

zur Programmierung mit MakeCode:
http://www.code-your-life.org/Lehrmaterial/mediabase/pdf/2780.pdf (Code your Life)
https://appcamps.de/unterrichtsmaterial/calliope-mini/ (Unterlagen von AppCamps; Registrierung notw.)
http://appcamps.de/handout-konzepte-informatik-fuer-pxt/ (Handout Konzepte Informatik (von AppCamps))
https://www.lpm.uni-sb.de/typo3/index.php?id=5892 (Info's und Lernkarten (Landesinst. f. Pädagogik und Medien Saarland))
https://portal.wissensfabrik.de/portal/fep/de/dt.jsp?setCursor=1_554644 (Download-Bereich f. div. Materialien (Projekt: IT2School auf Wissensfabrik.de))

zur Programmierung mit dem Open Roberta Lab (Nepo):
http://cdn.mint-zirkel.de/mint-zirkel-media/2017/09/Arbeitsblatt_Calliope_Freundschaftstest_MINTZirkel.pdf (Klett MINT-Zirkel)
http://appcamps.de/handout-konzepte-informatik-fuer-nepo/ (Handout Konzepte Informatik (von AppCamps))
https://calliope.cc/content/3-schulen/schulmaterial2/9783066000115-gesamt_pdf.pdf (Cornelsen: kostenlose PDF; Schüler-Arbeitsbuch)
https://calliope.cc/content/3-schulen/schulmaterial2/cornelsen_lehrerhandreichung.pdf (Cornelsen: kostenlose PDF; Lehrer-Handreichung)

weitere Themen:
https://www.hackster.io/calliope-mini/projects (div. Projekte auf hackster.io)

1.2.0.2. bbc micro:bit

wichtige Bau-Elemente (Sensoren / Aktoren)

-
-

-

Links:

https://github.com/carlosperate/awesome-microbit#-python (Link-Listen; diverse Links zu allen Themen (Schwerpunkt Python))

1.2.0.3. OxoCard

Wer hat's gemacht? – die Schweitzer. OxoCard kommt als relativ große Leiterplatine mit einem einfachen Papp-Gehäuse. Auf der Platine sind:

wichtige Bau-Elemente (Sensoren / Aktoren)

- **8 x 8 LED-Matrix**
-
-

1.2.0.4. Arduino / Genuino

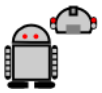
vor allem für Steuerungs-Aufgaben gedacht
Arduino UNO R3 ist derzeit quasi ein Standard
in vielen Experimentier-Kästen mit mehr oder weniger viel Zubehör verfügbar
derzeit wohl die preiswertiste Experimentier-Plattform

praktisch keine Sensoren oder Aktoren (nur einzelne LED's)
dafür diverse Ports / Pin's, an die zusätzliche Elektronik, Sensoren oder Aktoren angeschlossen werden kann
vor der Programmierung steht dann immer der Aufbau einer elektronischen Schaltung
meist auf oder über ein Steckbrett
alternativ gibt es Aufsteck-Board's, die diverse Aktoren und / oder Sensoren enthalten, hier entfällt das etwas längere Aufbauen und es bleibt mehr Zeit für das Programmieren
Die Programmierung der Arduino's ist sehr anspruchsvoll. Es wird eine Programmiersprache verwendet, die sich C++ nennt. Sie wendet sich an mehr freakige Programmierer. Dafür ist sie aber auch sehr leistungsfähig. Es ist eben nicht immer alles Gute beisammen. U.a. deswegen haben ja all die verschiedenen Programmier-Möglichkeiten eine Chance auf dem Markt.

wichtige Bau-Elemente (Sensoren / Aktoren)

-
-
-

1.2.0.5. BOB3



Hersteller: nicai systems
Arduino-basiertes System
Vertreib über diverse Elektronik-Händler (z.B. elv.de, pollin.de, conrad.de, reichelt.de, …)
Experimentier-Board, dass vor allem jüngere Schüler (vorrangig Grundschule (ev. auch Orientierungsstufe)) ansprechen soll
dazu passt die C++-basierte, Text-orientierte Programmierung wieder weniger
eher für etwas ältere Schüler, für die ist dann aber das Board wieder etwas zu verspielt und mit zu wenigen Sensoren und Aktoren ausgestattet
es gibt das System auch als Bausatz, so dass sich jeder seinen eigenen "BOB3" zusammen-löten kann / muss
System besteht aus der Roboter-förmigen Basis-Platine und einem "USB-Programmier-Helm"
der BOB3-"Roboter" kann individuell beim Nutzer bleiben, der Helm ist für die Programm-Übertragung per USB aber unbedingt notwendig und muss bei der Kosten-Planung beachtet werden
für den Solo-Betrieb ist eine Lithium-Ionen-Batterie CR2032 notwendig

in der zugehörenden Programmier-Umgebung sind geführte Projekte realisiert, die auch Alternativen beim Abarbeiten zulassen

wichtige Bau-Elemente (Sensoren / Aktoren)

- **weiße LED** 2 Stück
- **RGB-LED's** 2 Stück
- **Multifeld-Touch-Sensor** 2 Stück
- **IR-Sende-LED**
- **IR-Foto-Transistor** IR-Empfänger
-

mögliche Programmier-Umgebungen

- **ProgBob** Editor mit Anleitungen
  Text-orientierte Programmierung
  C++-ähnliche Programmier-Sprache
-

Links:
www.bob3.org (Projekt-Seite)
www.progbob.org (Programmier-Umgebung)

1.2.0.6. ESP-32 und ESP-8266

vielfach auf Arduino oder ähnlichen Board's basierende Technik
neue Board's mit neuen Mikroprozessoren ESP-8266 und ESP-32
Der ESP-8266 ist dabei die etwas einfachere und ältere Microcontroller-Version. Sie hat aber immer noch ihre Berechtigung in der modernen Informatik (→ IoT … Internet of Thinks).
da freies Platinen-Layout sehr viele Nachbauten mit unterschiedlichsten Namen

fast durchgehend mit WLAN ausgestattet
neuerste Versionen mit Bluetooth

gut für Versuche um Smart-Home, Vernetzung, Steuern und Regeln, App-Programmierung geeignet

Programmerung ist ähnlich anspruchsvoll, wie die der Arduino's (→ 1.2.0.4. Arduino / Genuino).

wichtige Bau-Elemente (Sensoren / Aktoren)

- Ein- oder Zwei-Kern-Prozessor
- **Temperatur-Sensor** mißt die Temperatur des Microprozessors auf dem Board
-

1.2.0.7. IoT-Bausteine

eigentlich oft nur Ableger in den Ausstattungen Duino und Nano
da freies Platinen-Layout sehr viele Nachbautn mit unterschiedlichsten Namen

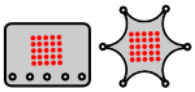
Duino, Pretzel-Board, IoT WiFi Board, IoT Bluethink Board
Nano Board

es gibt Versionen ohne Netzwerk, dabei entsprechen die Bausteine sehr kleinen Arduino's
einige Versionen besitzen eine WLAN-Erweiterung
neuerste Versionen mit Bluetoth

wichtige Bau-Elemente (Sensoren / Aktoren)

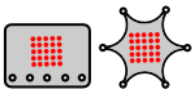
-
-
-

1.2.1. Programmieren mit NEPO (Open-Roberta-Lab)



Aufgaben:

1. *Lasse Dein Experimentier-Board das Lied "Alle meine Entchen" spielen!*
   Notenfolge:
2. *Kombiniere die Töne mit kleinen symbolischen Bildchen von einer schwimmenden Ente! Verwende immer ein Bild für 3 Noten!*
3.



Aufgaben:

1. *Lasse das Experimentier-Board Deinen Namen als Laufschrift anzeigen!*
2. *Erweitere Dein Programm nun um eine Abfrage der Taste "A" und wenn diese gedrückt wurde, soll die Anzeige gelöscht werden!*
3. *Nach dem Löschen soll jetzt die Buchstaben-Folge Deines Namen in umgekehrter Reihenfolge als Laufschrift ausgegeben werden!*
4. *Die Laufschrift soll nun durch das Drücken von Taste "B" beendet werden und dann die Buchstaben Deines Namens einer nach dem anderen für jeweils 1 Sekunde angezeigt werden!*

Aufgaben:

1.
2.
3.

1.2.2. Wir sagen dem Computer, was er machen soll! – Turtle-Graphik

sequenzielle Abläufe mithilfe blockbasierter Programmierumgebungen intepretieren, modifizieren und zielgerichtet gestalten

eine Abfolge gleicher Sequenzen durch eine Wiederholungsstruktur mit einer festen Anzahl von Wiederholungen ersetzen

die Arbeitsweise eines Algorithmus erklären

den Einfluss von Parameterwerten auf den Programmablauf untersuchen und beschreiben

Algorithmen in Hinblick auf Korrektheit beurteilen und hinsichtlich ihrer Effizienz vergleichen

1.2.2.x. Kara und Karol steuern

1.2.2.x.0. Aufbau der Welt von Kara und Karol

1.2.2.x.1. Erzeugen von Mustern

Bewegen und Setzen von Markierungen

Aufgaben:

1.
2. Erzeuge das nebenstehende Muster mit möglichst wenigen Arbeitsschritten (von Kara bzw. Karol)
3.

Was ist ein Algorithmus?

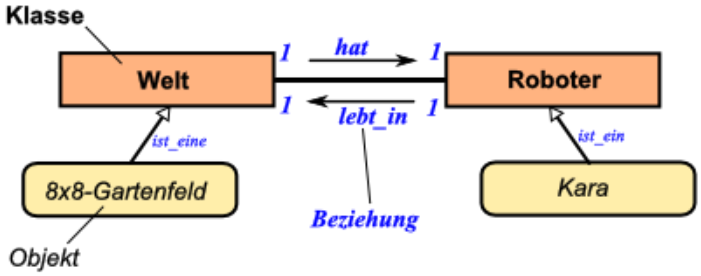
1.2.2.x.2. Kara, Karol und ihre Welt informatisch gesehen

Klassen Welt und Roboter / Käfer

Kara bzw. Karol als Objekte der Klasse Robote / Käfer

das 8x8-Spielfeld als ein Objekt der Klsse Welt

Beziehungen zwischen beiden Klassen



Klasse Welt und Klasse Markierung

Markierung als extra Klasse???
Markierung als Feld in Welt???

1.2.2.x.3. Vereinfachung von Algorithmen

Beschreiben von Wiederholungen

Erzeugen regelmäßiger Muster

Aufgaben:

1.
2.
3.

Erzeugen von Muster durch zwei ineinander geschachtelte Schleifen

Aufgaben:

1.
2.
3.

Merksätze: Wiederholung
Eine Wiederholung (Schleife, Schlaufe) ist die mehrfache Ausführung zusammengehörender (Einzel-)Anweisungen.

2. Bilder und Grafiken gestalten

Rastergrafiken gestalten

- *Werkzeuge zielgerichtet zur Gestaltung von Rastergrafiken einsetzen*
- *Anwendungsbeispiele zuordnen*

Verwendung von Zeichen- und Füllwerkzeugen, Einsatz von Farben, Korrektur von Zeichnungen
Rastergrafiken werden z.B. beim Drucken, bei der Anzeige auf Displays oder beim Legen von Mosaiken verwendet. Fotosensoren und Scanner erzeugen Rastergrafiken.

- *die Größe einer Rastergrafik durch die Angabe der Anzahl der Pixel in Breite und Höhe beschreiben und die Auswirkungen auf die Detailtreue erklären*

Die Bestimmung der Werte kann durch das Auszählen von Pixeln, die Anzeige von Objekteigenschaften oder die Recherche in technischen Dokumenten erfolgen.
Je größer die Anzahl der Pixel eines Bildes, desto genauer wird das Original durch die Rastergrafik repräsentiert.

elementare Werkzeuge pixelorientierter Grafiksoftware zur Erstellung und Bearbeitung von Rastergrafiken auswählen und benutzen

Anwendungsbeispiele für Rastergraphiken zuordnen

Vektorgrafiken gestalten

- *Werkzeuge zielgerichtet zur Gestaltung von Vektorgrafiken einsetzen*

Verwendung von geometrischen Grundformen zur Erstellung zusammengesetzter Figuren, Zuweisung von Eigenschaftswerten

- *Anwendungsbeispiele zuordnen*

Vektorgrafiken werden z.B. zur Beschreibung von Schriften, schematischen zeichnungen, CAD-Anwendungen, 2D-/3D-Modellen und –Animationen verwendet.

- *die Prinzipien der Raster- und Vektorgrafiken vergleichend beschreiben*

Die SuS vergleichen die Qualität von Raster- und Vektorgrafiken vor und nach dem Vergrößern. Sie wählen gezielt eine geeignete Software zur Gestaltung von Grafiken aus.

Vorschlag zur inhaltlichen Vertiefung:
Anhand der Bestandteile einer Vektorgrafik entwickeln die SuS Vorstellungen zu den Begriffen Objekt, Attribut und Attributwert.

Problem-Fragen für Selbstorganisiertes Lernen
Sind alle Bilder auf einem Computer gleich aufgebaut? Ist eine 20Mbit-Fotokamera besser als eine 4Mbit-Kamera? Um wieviel besser ist eine 20Mbit-Kamera im Vergleich zu einem 4Mbit-Gerät? Ist eine 20Mbit-Kamera den Mehrpreis wirklich wert?

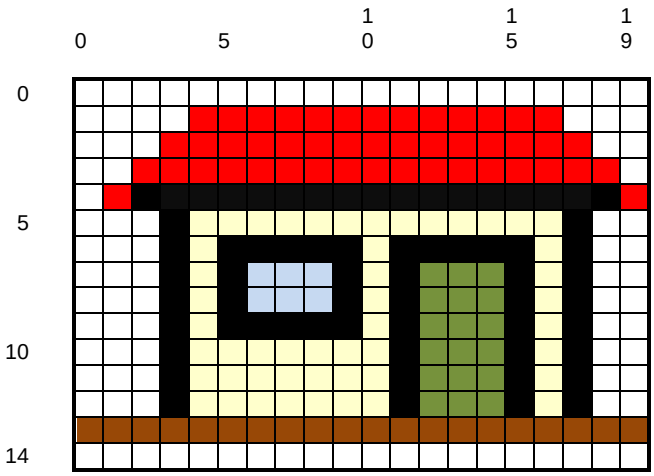
2.1. Malen am Computer - Rastergrafiken

Bei unseren ersten Programmier-Versuchen haben wir schon Raster-Grafiken benutzt. Nichts anderes können die LED-Matrizen auf den Experimentier-Board's darstellen.

Merksätze: Raster-Grafik
Eine Raster-Grafik ist eine bildliche Darstellung, deren Gesamt-Information aus mehr oder wenigen vielen Einzel-Punkten (Pixeln) zusammengesetzt sind.

Nebenstehendes Bildchen wird jeder als Haus identifizieren können. Die charakteristischen Teile, wie Dach, Tür und Fenster sind zwar nur angedeutet, aber wir kennen solche kleine Pictogramme von Computer- und Smartphone-Bildschirmen.

Schaut man sich die Grafik aber genauer an, dann erkennen wir schnell, dass dieses Bild aus einzelnen Quadraten zusammengesetzt ist.

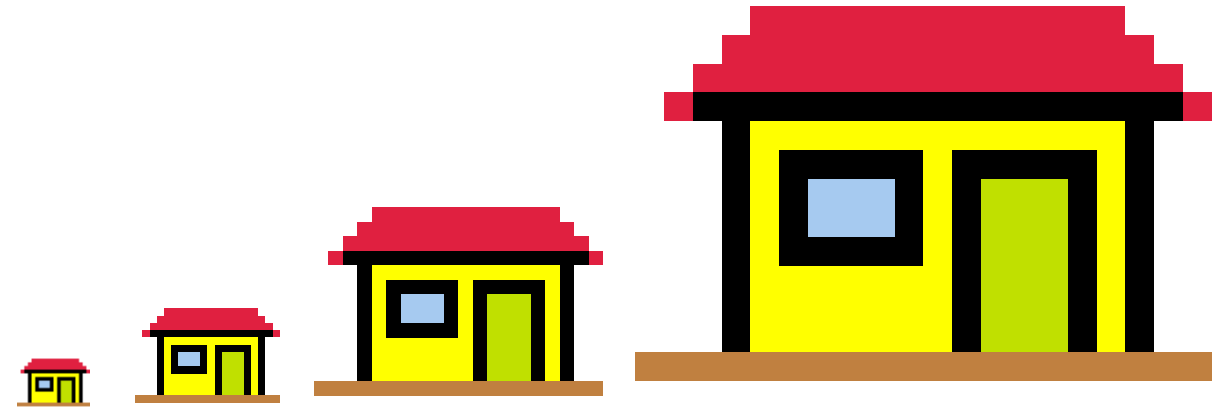


Aufgaben:

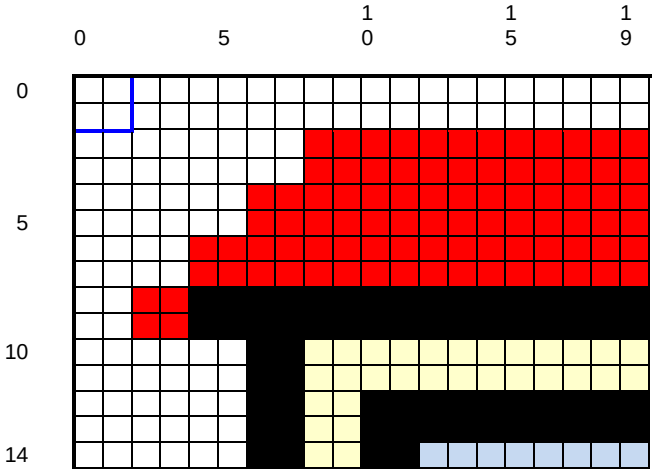
1. *Starte das Mal-Programm "Paint" und ändere sofort die Eigenschaften auf eine Breite von 20 und eine Höhe von 15 Pixeln!*
2. *Speichere die Datei mittels "Speichern unter …" in Deinem eigenen Ordner mit dem Namen "RasterHaus.BMP" (256-Farben-Bitmap) ab!*
3. *Zum Zeichnen wird bei "Ansicht" solange vergrößert, wie es geht!*
4. *Erstelle eine exakte Kopie der Grafik! Speichere zwischendurch und am Ende eigenständig ab!*

Nun wollen wir die Bild-Qualität verbessern und vergrößern das Haus-Rasterbild mehrfach.

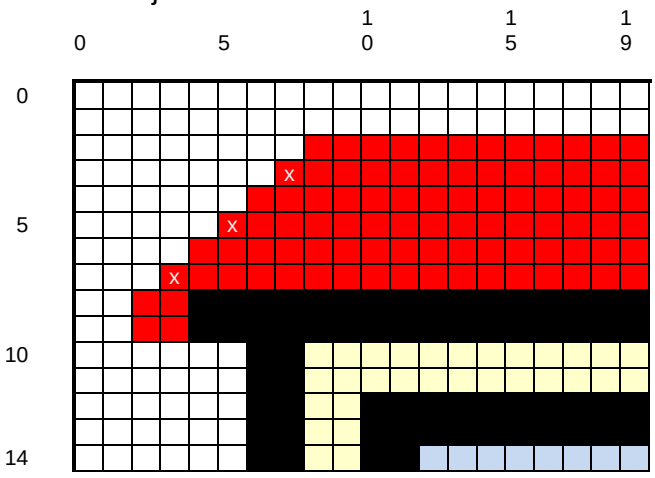
den Zusammenhang zwischen Anzahl der Pixel und der Detailtreue einer Rastergrafik beschreiben und bei der Auswahl von Bildern beachten



Das Ergebnis ist aber enttäuschend. Jetzt treten die Pixel richtig deutlich hervor. Da das Mal-Programm stur die Skalierung (Vergrößerung) vornimmt, werden aus jeweils einem Pixel zum 2 x 2 Pixel.
Für das ursprünglich oberste, linke Pixel (Pixel-Position: 0,0) ist das durch einen blauen Rahmen verdeutlicht worden.
Für den Computer ist das Arbeiten mit Raster-Grafiken extrem einfach. Es müssen jan nur stur bestimmte Umrechnungen gemacht werden. Das konnten schon die ersten PC's sehr gut. Aus dieser Zeit stammt auch das Mal.Programm "Paint", auch wenn es jetzt modern aussieht.



Damit das Bild nun in der etwa gleichen Qualität, wie das Original, erscheint, müsste man die Kanten nachbearbeiten und einzelne Pixel dazusetzen oder weglöschen. Dadurch verbessert man die Rasterung.
Im nebenstehenden Bild sind die zusätzlichen Pixel mit einem weißen x gekennzeichnet. Bei kleinen Bildern mag das noch gehen – für große Bilder ist dieses Verfahren aber nicht geeignet.



Aufgaben:

1. *Ändere die Bildgröße bei den Eigenschaften auf das Doppelte! Vergrößere dann über "Auswählen" und "Größe ändern" das Haus auf 200 % (Horizontal und Vertikal)! Speichere die Datei dann als "RasterHaus 2x.BMP" ab!*
2. *Wiederhole das Verfahren von 1. noch zweimal! Ändere dabei den Dateinamen auf 4x bzw. 8x!*

Was passiert, wenn man nun das Bild verkleinert – also runterskaliert?

Aufgaben:

1. *Öffne jetzt wieder das Original-Bild "RasterHaus.BMP"!*
2. *Gehe bei "Auswählen" auf "Alles Auswählen" und ändere dann die Größe des Bildes auf jeweils 50 % (Horizontal und Vertikal)!*
3. *Stelle die Bildgröße auf 10 x 8 Pixel ein und speichere dann das Bild als "RasterHaus verkleinert.BMP"!*
4. *Vergrößere die Ansicht in Paint solange, wie es geht! Was ist passiert? Wie kann man diesen Effekt erklären?*
5. *Vergrößere nun wieder die Bildgröße auf 40 x 30 Pixel! Wähle wieder das gesamte Bild aus und vergrößere auf jeweils 400 %! Speichere die Datei als "RasterHaus verkleinert 4x.BMP" ab!*

Was da von unserem Haus übrigbleibt, verdient kaum noch den Namen. Die Zusammenfassung von immer 2 x 2 Pixeln zu einem erzeugt einen deutlichen Informations-Verlust.

Dieser ist auch nicht mehr zu reparieren. Einmal weggerechnete Bild-Informationen lassen sich nicht wieder zurückrechnen.

Pixel-Grafiken lassen sich nur begrenzt skalieren.
Meist kommt es dabei zu deutlichen Informations-Verlusten.

2.1.1. Speicherplatz-Bedarf bei Raster-Grafik-Dateien

die Größe einer Rastergrafik durch Angabe der Pixel in Breite und Höhe beschreiben

Aufgaben:

1. *Übernimm die nachfolgende Tabelle und fülle die leeren Zellen aus! Es werden exakte Dateigrößen (in Byte) benötigt. Wo kann man diese herbekommen? (Falls Du keine der abgespeicherten Dateien hast oder Dir einzelne Dateien fehlen, dann musst Du alle bzw. die entsprechenden Aufgaben vom letzten Kapitel noch einmal exakt abarbeiten!)*

Dateiname	Bild-Breite	Bild-Höhe	Gesamt-Pixel-Anzahl	Dateigröße in Byte
RasterHaus.BMP	20	15	300	1'378
RasterHaus 2x.BMP				
RasterHaus 4x.BMP				
RasterHaus 8x.BMP				
RasterHaus verkleinert.BMP				
RasterHaus verkleinert 4x.BMP				

2. *Vergleiche die zusammengetragenen Daten! Was stellst Du fest? Welche Erklärungen kannst Du für die Effekte geben?*

für Experten und zum Knobeln:

3. *Findest Du eine exakte Regel / Gesetzmäßigkeit / Formel zur Berechnung der Dateigröße? Zum Ausprobieren oder Überprüfen der Berechnung kannst Du auch noch weitere Dateien erzeugen! Dokumentiere die Dateien mit ihren Daten in der Tabelle!*

BMP-Dateien

ursprünglich als einfache Kopie des Arbeitspeichers gedacht, hat sich das BMP-Format zu einem sehr weit verbreiteten Grafik-Dateiformat entwickelt.
So eine Kopie des Speichers auf einen Datenträger zu speichern ist eine recht einfaches Verfahren, weshalb es auch so gerne benutzt wird.

ähnlich sind TIF- / TIFF-Dateien organisiert

Bei zwei-farbigen Bildern – also schwarz-weiß-Zeichnungen – braucht man für einen Pixel auch nur 1 bit. In einem Byte lassen sich so 8 Pixel-Informationen speichern.

Aufgaben:

1. *Ein Schwarz-Weiß-Bild ist 1024 x 512 Pixel groß. Wie viel Speicherplatz braucht man zum Abspeichern?*
2. *Wie verändert sich der Speicher-Bedarf, wenn das Bild doppelt so breit und doppelt so hoch ist?*
3.

Mit reinen Schwarz-Weiß-Zeichnungen gibt man sich heute kaum noch irgendwo zufrieden. Für technische Zeichnungen reicht das auch aus.
Aber schon Foto's in der Zeitung sollten zumindestens auch Grau-Töne enthalten. Da unser Auge nur rund 100 Grau-Töne unterscheiden kann, nutzt man in der Computer-Grafik 256 Grau-Töne. Diese Zahl hat man gewählt, weil in einem Byte genau 256 verschiedene Werte

gescheichert werden können. Mit anderen Worten für jedes Grau-Pixel wird genau ein Byte benutzt.

Aufgaben:

1. ***Das obige Schwarz-Weiß-Bild mit einer Ausdehnung von 1024 x 512 Pixel soll als Grau-Ton-Bild gespeichert werden. Überlege Dir, ob sich der Speicher-Bedarf wirlich ändert? Wenn JA, dann berechne den notwendigen Speicherplatz! Wenn NEIN, dann erkläre den Anderen, warum der Speicher-Bedarf gleich bleibt!***
2. ***Wie verändert sich der Speicher-Bedarf, wenn das Grauton-Bild doppelt so breit und doppelt so hoch ist?***
3.

Bei farbigen Bildern werden die Farben in drei Teile zerlegt. Die Teilfarben Rot, Grün und Blau lassen sich zu jeder Farbe mischen.
Dazu werden die einzelnen Anteile nur unterschiedlich stark angezeigt.

Aufgaben:

1. ***Betrachte bei einem älteren Fernseher die Vorderseite durch eine Lupe! Was beobachtest Du? Schaue Dir dazu verschieden-farbige Regionen auf dem Bildschirm an! (Für einen modernen Bildschirm oder Monitor brauchst Du eine sehr gute Lupe!)***
2.
3.

BMP-Dateien mit Lauflängenkodierung

BMP-Dateien mit Farb-Tabellen

Einschränkung auf die Farben, die wirklich im Bild vorhanden sind. Selten sind wirklich alle möglichen Farben in einem Bild enthalten.

Erstellen einer Farb-Tabelle mit allen im Bild enthaltenen Farben und der Vergabe einer Farb-Nummer. Diese Nummer wird Index genannt.
Für jeden Bildpunkt wird jetzt nur die Index-Nummer gespeichert. Das spart 1 bis 2 Byte pro Pixel.

Beim Laden eines Bildes muss dann für jeden Bild-Punkt in der Index-Tabelle nachgesehen, welche Farbe genau dem gespeicherten Index zugeordnet war. Diese Farbe wird dann zur Anzeige gebracht.

moderne Dateitypen

PNG

JPG

2.1.2. Rastergrafik in der Praxis

Scanner

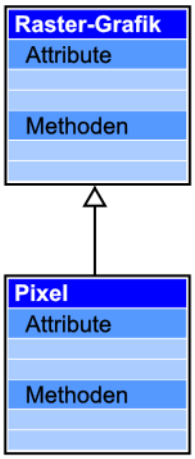
Digital-Kamera's

2.1.2.1. Flächen-Berechnung bei Pixel-Grafiken

2.1.3. Objekte und Strukturen in Rastergrafiken

Objekt:			**Bemerkungen / Hinweise**
Pixel			
→ Attribut(e):		***Wert***	
• Position			
○ x-Position	=	*13*	
○ y-Position	=	*10*	
○ Pixelfarbe	=	*grün*	intern auch Nummern: RGB = 0, 255,0
•	=		
→ Methode(n):			
• Neu erstellen			beim Links-Klick auf die Zeichenfläche
• löschen			mit Rechts-Klick
• (Bearbeiten) ausschneiden			
• (Bearbeiten) kopieren			
• (Bearbeiten) einfügen			
• Farbe ändern			
• skalieren			vergrößern / verkleinern auf dem Bildschirm
•			

Objekt:			**Bemerkungen / Hinweise**
Bild-Dokument			Bitmap-Typ
→ Unterobjekt(e):			
• Pixel			Voll-Liste
•			
→ Attribut(e):		***Wert***	
• Dateiname	=	*Haus*	dieses Attribut kann der Nutzer direkt ändern
• Dateityp	=	*BMP*	
• geändert	=	*wahr*	dieses Attribut wird vom Programm verwaltet
• Breite	=	*30 px*	
• Höhe	=	*20 px*	
•	=		
→ Methode(n):			
• Neu erstellen			
• Speichern unter …			
• Speichern			
• Öffnen			
• Schließen			
•			


Raster-Grafik
Attribute
Methoden
Pixel
Attribute
Methoden

2.2. Zeichnen am Computer – Vektorgrafiken

elementare Werkzeuge vektororientierter Grafiksoftware zur Erstellung und Bearbeitung von Grafikobjekten auswählen und benutzen

Attributwerte eines Objektes der Vektorgrafik bestimmen, strukturiert darstellen und gezielt ändern

Merksätze: Vektor-Grafik
Eine Vektor-Grafik ist eine bildliche Darstellung, deren Gesamt-Information aus graphischen / geometrischen Grund-Objekten (z.B.: Punkte, Linien, Rechtecke, …) zusammengesetzt sind.

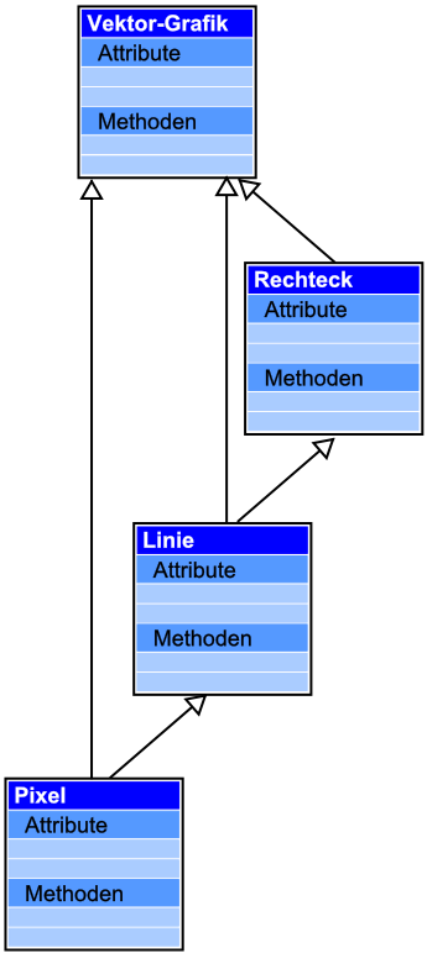
**Vektor-Grafiken lassen sich beliebig skaleren,
ohne das Informations-Verluste auftreten.**

Objekte und Strukturen in Vektor-Grafiken

Objekt:			Bemerkungen / Hinweise
Pixel			
→ Attribut(e):		***Wert***	
• Position			
○ x-Position	=	*126*	
○ y-Position	=	*538*	
○ Pixelfarbe	=	*grün*	intern auch Nummern: RGB = 0, 255,0
•	=		
→ Methode(n):			
• Neu erstellen			beim Links-Klick auf die Zeichenfläche
• löschen			mit Rechts-Klick
• (Bearbeiten) ausschneiden			
• (Bearbeiten) kopieren			
• (Bearbeiten) einfügen			
• Farbe ändern			
• skalieren			vergrößern / verkleinern auf dem Bildschirm
•			

Objekt:			Bemerkungen / Hinweise
Linie			
→ Unterobjekt(e):			
• Pixel			
•			
→ Attribut(e):		***Wert***	
• Pixel1	→	*Pixel*	Veweis auf: Pixel (Startpunkt der Linie)
• Pixel2	→	*Pixel*	Veweis auf: Pixel (Endpunkt der Linie)
• Linienformat			
○ Linienart	=	*durchgezogen*	oder: gestrichelt, gepunktet, …
○ Linienstärke	=	*1 px*	
○ Linienfarbe	=	*blau*	intern auch Nummern: RGB = 0,0,255
•	=		
→ Methode(n):			
• Neu erstellen			durch Ziehen mit der Maus
• Linie zeichnen			Pixel auf Verbindungslinie berechnen
• löschen			
• (Bearbeiten) ausschneiden			
• (Bearbeiten) kopieren			
• (Bearbeiten) einfügen			
• Farbe ändern			
• skalieren			
•			

Objekt:			Bemerkungen / Hinweise
Rechteck			
→ Unterobjekt(e):			
• Pixel			
• Linie			
→ Attribut(e):		***Wert***	
• Pixel1	→	*→ Pixel*	obere, linke Ecke d. Rechtecks
• Pixel2	→	*→ Pixel*	untere, rechte Ecke des Rechtecks
• Linie	→	*→ Linie*	Rahmenlinie
• Innenfläche			
○ Füllart	=	*einfarbig*	oder: zweifarbig, Muster, …
○ Linienstärke	=	*1 px*	
○ Füllfarbe	=	*gelb*	intern auch Nummern: RGB = 0,0,255
•	=		
→ Methode(n):			
• Neu erstellen			durch Aufziehen mit der Maus
• Innenfläche zeichnen			inneres Pixelmuster berechnen
• löschen			
• (Bearbeiten) ausschneiden			
• (Bearbeiten) kopieren			
• (Bearbeiten) einfügen			
• Farbe ändern			
• skalieren			
•			



Aufgaben:
1. Erstelle eine Objekt-Definition für "Dreieck" und "Kreis"!
2.
3.
für Experten und zum Knobeln:
x. !

Anwendungsbeispiele für Vektorgrafiken zuordnen

2.3. Was ist besser – Pixel- oder Vektor-basiertes Zeichnen?

die Eignung von Raster- oder Vektorgrafiken für bestimmte Aufgaben beurteilen

	Raster-Grafik	**Vektor-Grafik**
Gemeinsamkeiten	• dienen der Darstellung und Beschreibung von grafischen Objekten	
Unterschiede Grund-Elemente	• ausschließlich aus Punkten (Pixeln) zusammengesetzt	• aus Punkten, Linien und Flächen zusammengesetzt

Aufgaben:

1. *Übernehme die obige Tabelle und ergänze die fehlenden Sachverhalte!*
2. *Immer zwei Personen bilden ein Team. Der Kursleiter legt fest, ob das Team ein "Rastergrafik-Verfechter" oder ein "Vektorgrafik-Verfechter" sein soll. Immer zwei Teams treten gegeneinander an und müssen ihren Grafik-Typ gegen ein anderes "feindliches" Team debattieren (Redezeit z.B. immer abwechselnd 2 min). Ein Schüler ist Leiter der Debatte. Die Zuhören entscheiden am Schluss (z.B. nach einer bestimmten Zeit) über den Titel "Bestes Team".*

2.

2.4. Bild- und Foto-Bearbeitung

Bildschirmausschnitte und Bildschirmkopien erzeugen

Bild-Verarbeitung mit Snap!

Bilder aus Punkten zusammengesetzt, Pixel genannt, gemeint ist der Bild-Punkt
die individuelle Form unterscheidet sich von einer runden oder quadratischen Fläche
da die Pixel i.A. auf einer (Foto-)Kopie, einem Farb-Ausdruck oder am Bildschirm sehr klein sind, ist die Form nicht so entscheidend
der Begriff Pixel ist aus den verkürzten englischen Begriffen picture (für: Bild) und element (dt.: Element, Teil). Abkürzungen sind z.B. px oder Pel.

RGB-Farbsystem

basiert auf den Fähigkeiten unseres Auges verschiedene Farb-Anteile zu erfassen
Farb-Rezeptoren
neben Helligkeits-Rezeptoren, die quasi Schwarz-Weiß-Rezeptoren entsprechen
jeder Farb-Anteil darf im einfachen Farbsystem Werte zwischen 0 und 255 einnehmen. Das entspricht einem Byte (also 8 bit). Für jeden Bildpunkt braucht man also 3 Byte.
Hat jeder Farb-Anteil den Wert 0 – also RGB = (0,0,0) – dann ist der Bildpunkt schwarz. Die Werte 255 für jeden Farb-Anteil bewirken insgesamt die weiße Farbe.

Aufgaben:

1. *Gehe beim Fernseher zuhause ganz dicht an die Bildfläche heran! Welche Bild-Punkte / -Farben erkennst Du hier?*
2. *Überlege Dir, um welche Farben es sich jeweils handelt! Erläutere Deine Voraussage!*

a)	R=0, G=0, B=255	b)	RGB=(255,0,0)	c)	RGB=(255,255,0
d)		e)	RGB=(255,255,255)	f)	
g)	R=127, G=127, B=127	h)	R=0, G=0, B=0	i)	RGB=(0,0,255)

3. *Probiere mit Snap! die einzelnen RGB-Werte aus! Stimmen Deine Voraussagen zu den Farben? Wenn Du falsch vermutest hast, dann überlege Dir, welchen Denkfehler Du gemacht hast!*

Fortsetzung hinter Bilder und Grafiken gestalten

in Snap! ein Bild als Kostüm laden
Bild als Liste von Pixeln
man kann einfache Bildb-Bearbeitung machen z.B. das Bild in Grau-Bild wandeln
Rot- Grün- oder Blau-Filter erstellen

s.a. Informatik-Kurs (OpenSAP) 1. Woche Unit 4 BVa +BVb
für fortgschrittene Programmierer:
Wechseln der Farben z.B. Rot-Wert mit Grün-Wert tauschen, …, dann alle drei Werte im Ring tauschen

bemalen der Bühne zufällig (mit zufällig bewegtem Stift)
… mit einem Stift der zufällig seine Farbe ändert
… mit einem Stift der zufällig seine Dicke ändert
"stempeln" der Bühne mit einer selbsterstellten Form auf zufälligen Positionen
… mit veränderter Größe (zufällig)
… erweitern mit zufälliger Farbwechsel
für fortgschrittene Programmierer:
… Nutzen einer speziellen Funktion, die "zerspritzende Tropfen-Muster" erzeugt ("Drip Painting"-Funktion muss importiert werden!)
s.a. Informatik-Kurs (OpenSAP) 1. Woche Unit 5 ZG

Ausgeben von Texten zuerst nur zufällig auf der Bühne
… erweitert durch verschiedene / zufällige Größe / Farbe / Richtung
s.a. Informatik-Kurs (OpenSAP) 1. Woche Unit 5 ZG

Links:
http://codiertekunst.joachim-wedekind.de/projekte/ ()

3. Präsentationen mit Bildern gestalten

Bilder recherchieren und speichern

- *Bildrecherchen unter Nutzung von Filtern zielgerichtet durchführen*
- *Bilder nach inhaltlichen, gestalterischen und technischen Kriterien bewerten, auswählen und unter Beachtung des Urheberrechts speichern*
- *Quellen mit Angabe des Urhebers, des genauen Fundorts und der Lizenz erfassen*

Für die Recherche sollen Ressourcen verwendet werden, deren Werke unter Creative Commons Lizenzen stehen.
Bewertungskriterien:

- *inhaltlich: z.B. Echtheit, Aussagekraft*
- *gestalterisch: z.B. Qualität, Kontraste*
- *technisch: z.B. Bildgröße, Detailtreue*

Es sollen Bezüge zu Lizenzformen und zum Recht am eigenen Bild hergestellt werden. Insbesondere sind auch die Persönlichkeitsrechte anderer zu beachten.

Foliensätze für eine Präsentation gestalten

- *Foliensätze mit klar strukturierten und gut lesbaren Folien einheitlich gestalten*

Es bietet sich die Verwendung von vordefinierten Folienlayouts an. Die SuS achten auf den Einsatz serifenloser Schriften, geeigneter Schriftgrößen und Farbkontraste. Sie setzen Schriftschnitte gezielt zur Akzentuierung von Textelementen ein.

- *Texte eingeben und korrigieren*

Folien enthalten maximal Wortgruppen und kurze Sätze.

- *Bilder einfügen und anpassen*

Eine Skalierung der Bilder erfolgt unter Beibehaltung des Seitenverhältnisses.

Vorschläge zur inhaltlichen Vertiefung:

- *Animationen und Folienübergänge einheitlich und sparsam verwenden*
- *Bildausschnitte und Bildschirmkopien erzeugen*

Problem-Fragen für Selbstorganisiertes Lernen
Wie bekommt man an tolle Bilder für Präsentationen und Vorträge? Was sind (Bild-)Lizenzen und (Bild-)Rechte? Was sind Urheberrechte? Ist alles durch Urheberrechte geschützt? Darf man jedes Bild verwenden? Darf ich alles fotographieren? Darf jeder mich fotographieren? Wer hat das Recht an Bilder von mir selbst?

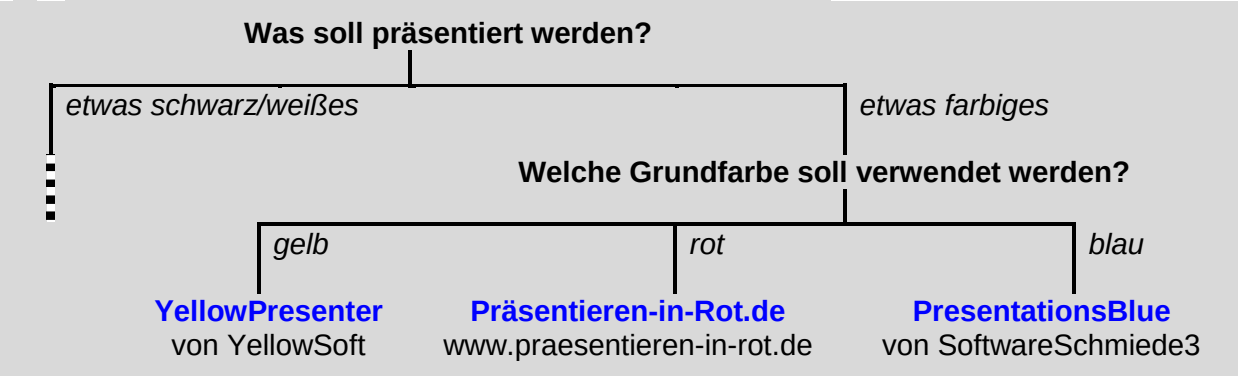
3.1. Bilder suchen

3.2. Präsentation im Stil von PowerPoint und Co

Problem-Fragen für Selbstorganisiertes Lernen
Was heißt eigentlich Präsentieren? Was kann man alles, wie, wann und womit präsentieren? Wie wichtig ist ordnungsgemäßes Präsentieren in der heutigen Welt? Geht es auch ohne Präsentieren? Woran erkennt man gute / erfolgversprechende Präsentationen? Welche typischen Präsentations-Fehler sollte man vermeiden? Welchen Stellenwert hat eine gute Präsentation für einen öffentlichen Auftritt / eine Vorstellung? Was ist ein Präsenter? Was ist ein Präsentierer? Was ist ein Präsentator?

Aufgaben:

1. *Suchen Sie sich mindestens zwei unterschiedliche Definitionen für den Begriff Präsentation heraus! Geben Sie mindestens 10 verschiedene Arten / Typen von Präsentationen an!*
2. *Recherchieren Sie, welche Präsentations-Programme bzw. –Apps verfügbar sind (mindestens 6 Beispiele)! Dokumentieren Sie alle Recherche-Ergbnisse handschriftlich und Stichpunkt-artig! Nutzen Sie dazu die folgenden Orientierungs-Punkte, zu denen Sie Auskunfts-fähig sein müssen:*
   - *Kurzbemerkungen (z.B. Hersteller, Internet-Link, Arbeitsweise, …)*
   - *Vorteile / Vorzüge / Besonderheiten*
   - *Nachteile / Schwachstellen*
   - *Eignung für …*
3. *Überlegen Sie sich, oder recherchieren Sie, Einteilungs-Möglichkeiten für Präsentations-Programme bzw. Apps! Notieren Sie immer Beispiele! (Mindestens sollten immer Ihre Beispiele von Aufg. 2 dabei sein!)*
4. *Erstellen Sie einen Vorschlag für einen sinnvollen und handhabbaren Entscheidungs-Baum, mit dessen Hilfe Sie oder einen andere Person sich ein geeignetes Präsentations-Programm auswählen können! (Orientieren Sie sich am sehr vereinfachten Beispiel unten! Ihre Programme bzw. Apps aus Aufg. 2 sollten mindestens erhalten sein! Der Baum muss praktisch zu nachvollziehbaren Lösungen führen! Für einen deutlich erweiterten Entscheidungs-Baum gibt es eine Zusatz-Bewertung!)*



kleine Hilfs-Liste zu Präsentations-Programmen und Apps:
→ digital storytelling → presentation →
microsoft PowerPoint, libreoffice / openoffice.org / apache Impress, softmaker Presentations, google Präsentation, adobe Spark, microsoft sway, kingsoft Presentation
prezi, google slides, ApowerMirror, ZohoShow, Haiku Deck, emaze, OpenSlides, PowToon, Swipe, Bunkr, Canva, pouchway, impress.js, Impressive, FoilPresenter, SlideRocket, SlideDraw, HTML Slidy, KinetiCast, …
KPresenter, Ease, Wink, …
Visual Understandig Environment (VUE)
apple (iCloud) Keynote, Deckset, …
inkscape mit Sozi oder Jessyink, LaTeX mit beamer, microsoft PowerPoint Web App, …

Der Begriff PowerPoint bzw. PowerPoint-Präsentation ist mitlerweile schon in den normalen Sprachgebrauch übergegangen. Es fällt uns schwer, einen anderen Begriff für diese Art von Präsentation zu finden. Das Programm PowerPoint ® aus der microsoft ® office-Suite ist die Software, die diese Art von Präsentation einer wirklich breiten Masse verfügbar gemacht hat. Es ist leicht zu bedienen und bringt schnell erste Erfolge. Mit etwas Fachkenntnissen oder Experimentiergeist lassen sich viele Spielereien hinzufügen und einarbeiten. Das befriedigt auch die Effekthascher, die glauben, wenn eine Präsentation viele spezielle Spielereien und Effekte enthält, dann wäre sie auch besonders toll.
Das Programm PowerPoint ® hat mitlerweile viele gleichgute Konkurrenten. Da ist z.B. das Impress aus der OpenOffice.org-Suite bzw. Apache OpenOffice-Suite oder der LibreOffice-Suite zu nennen.
Allen gemeinsam ist eine lineare Abfolge von Folien, die aber durch Verlinkungen bereichert werden können.
Vor allem im online-Bereich haben sich neben PowerPoint-Äquivalenten auch neue Umsetzungen des Folien-Präsentations-Konzeptes etappliert. Diese Programme nutzen die Hypertext-Fähigkeit des Mediums Internet aus und bieten vermaschte / verlinkte Präsentationen, bei denen die Folien zum Anzeigen z.B. immer blasen-artig herausgeholt werden. An den Grund-Prinzipien hat sich aber wenig geändert. Einige Apps verzichten aber auf komplexe Schicht-Strukturen und punkten mit einfachen, schnell realisierbaren Elementen und Effekten.
Vertreter dieser neuen Präsentations-Art sind prezi.com, und slider.com.

Registrieren bei prezi.com (→) ersteinmal in der Kosten-freien "Basic"-Variante. Wer dann später mehr prezi möchte, der kann jederzeit zu den Kosten-pflichtigen Versionen aufsteigen. Diese beiten einen deutlichen Mehrwert.

5.x.1. informatische Grundstrukturen in Präsentationen

Bei den meisten Präsentations-Programmen findet man ein dreischichtiges Aufbau-Prinzip. Von den erfahrenen Benutzern wird dieses Prinzip als völlig normal und extrem hilfreich empfunden und benutzt. Intuitive Anfänger mit kleinen Blicken zu Hilfe-Portalen nehmen das Konzept ganz nebenbei auf. Nur die "Ich-weiss-alles-besser"- oder "Ich-mach-das-(schon-immer-)so"-Anwender beißen sich sich oft die Zähne aus.
Dabei ist die Struktur ganz logisch aufgebaut und orientiert sich am Objekt-Konzept.

Drei-Ebenen-Prinzip von Präsentationen

- **Folienmaster (Vorlage, Design)**
  Hintergrund, Rahmen (Kopf- und Fuß-Zeilen), …
  typische Schrift-Arten, -Größen und -Stile
  corporate design; Firmen- oder Institutions-Vorgaben
  → hoher Wiedererkennungswert, schneller Start
- **Layout (Anordnungs-Hilfe)**
  Hilfs-Objekte (Text- und Bild-Boxen, MediaPlayer-Bedienelemente, …)
  ergonomische Anordnung von Elementen (Präsentations-Objekten)
  allgemeine Vorbelegungen für Objekt-Klassen (z.B. Texte, Tabellen, Bilder, …)
  Angriffs-Elemente für Animationen und Effekte
- **Folie (Inhalt)**
  beinhaltet den eigentlichen Text, die Tabellen, Bilder – also die Daten – einer Präsentation

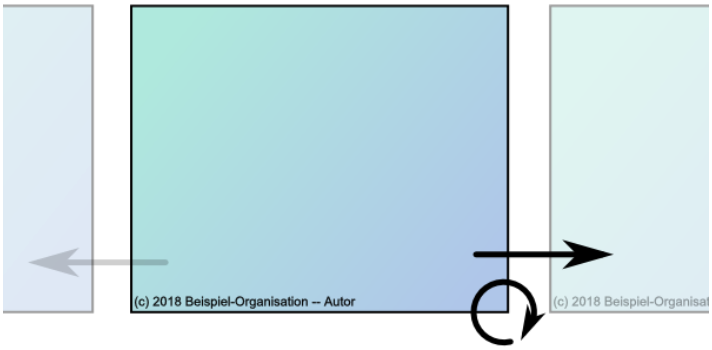
Die wichtige informatische Grund-Idee hinter der Dreiteilung ist die Trennung von Inhalt und Design. Viele Daten sollen möglichst schnell und einfach wieder- oder weiter-verwendet werden können. Den Inhalt – oder Teile davon – möchte man vielleicht in eine andere Präsentation oder einen anderen Präsentations-Typ übernehmen. Da ist ein Zugriff nur auf Inhalts-Daten sehr praktisch. In anderen Fällen will man das Design einer Präsentation immer wieder nutzen. Ein gutes Beispiel ist das corporate design, das den Wiedererkennungs-Effekt fördert. Wir erkennen z.B. bestimmte Werbe-Prospekte schon von weitem an ihrem Aussehen. In diesen Fall werden von einer Präsentation nur die Design-Daten weitergegeben. Die konkreten Inhalte werden jeweils neu ergänzt.
Die zwischen Design und Inhalt liegende Layout-Ebene dient vor allem als Hilfmittel im Bearbeiten-Modus und vereinfacht das Verknüpfen der Daten-Ebenen.

Folienmaster

Folienmaster in LibreOffice-Impress

auch Masterfolien genannt
Grundeinstellungen für die Präsentation
Änderungen wirken sich auf alle Folien aus

Rahmen, Hintergrund, Firmenlogo, corporate design, Folien-Übergänge
also eigentlich das Makro-Layout



bei Einstellungen am Folienmaster möglichst wenig weglöschen sondern besser ändern
ansonsten kann es zu "Folge-Fehlern" bei den abhängigen Ebenen "Layout" und "Folie" kommen

Der Folienmaster vererbt seine Einstellungen an alle nachfolgenden Schichten – wie Layout's und Folien.

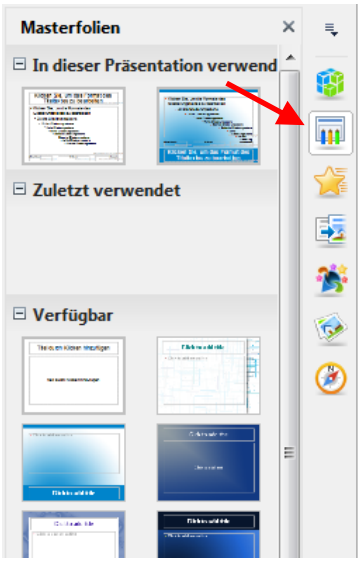
Ändert man nun später für eine spezielle Präsentation den Folienmaster einer eigentlich schon fertigen PowerPoint-Präsentation, dann sehen alle Folien ab der Änderung neu aus. Ausnahmen sind durch spezielle Festlegungen auf den Folien (der Inhalts-Ebene) möglich.

Folienmaster / Masterfolien im LibreOffice (Impress)

Auswahl eines Folienmasters über die Eigenschaften-Leiste

über "Ansicht" "Folienmaster" gelangt man in den Bearbeitungs-Modus

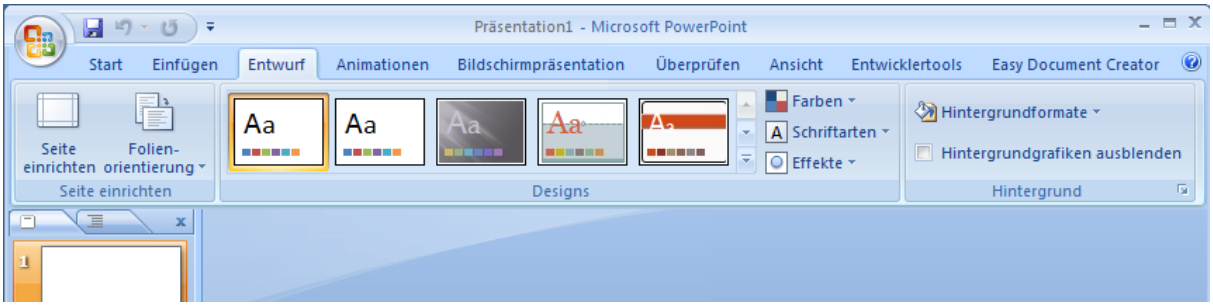
Folienmaster / Masterfolien im Apache OpenOffice (Impress)



Das Impress aus der Apache-OpenOffice-Suite ist dem Impress aus dem LibreOffice recht ähnlich. Beide Suiten entstammen der gleichen Code-Basis. Leider haben sich die Entwickler-Team aufgeteilt und auch der Rechte-Inhaber / Besitzer des OpenOffice hat in der letzten Zeit mehrfach gewechselt. Die Versionen des OpenOffice hinken derzeit hinterher.
Für Normal-Anwender ist das Programm voll nutzbar. Die erstellten Präsentationen (-Dateien) können im LibreOffice weiterverwendet werden. Ob ein zurück zu OpenOffice noch funktioniert, muss ausprobiert werden. Man sollte sich auf ein Programm einschießen.

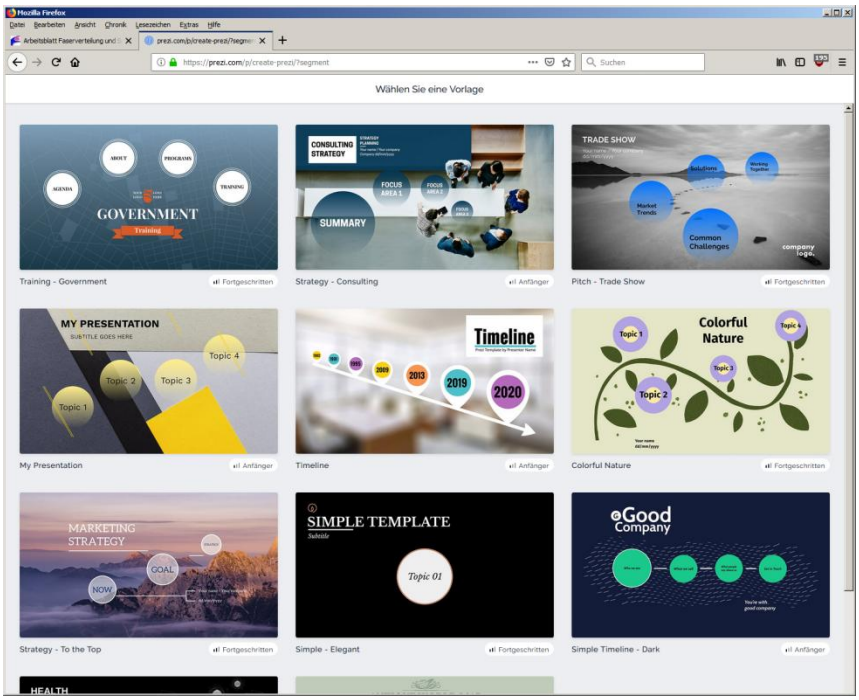
Folienmaster im microsoft® Office (PowerPoint)

Die Folien-Master in PowerPoint findet man unter dem Reiter "Entwurf". Im Bereich "Design" stehen eine Vielzahl von Vorlagen zur Verfügung.



Für das Gesamt-Erscheinungsbild der Präsentation ist auch immer der Hintergrund wichtig. Dabei sollte er – genau so, wie das Design – entweder zum Thema oder zum Ersteller passen (corporate design). Wenn man die Möglichkeit hat, dann sollten hier angenehme Farben in der mittleren Farbtiefe und ohne viele Effekte (Verläufe usw.) gewählt werden. Mittlere Farbtöne sind deshalb gut geeignet, weil man u.U. sowohl mit sehr dunklen (z.B. blau und schwarz) aber auch mit sehr hellen (z.B. weiss und gelblich) Farben im Vordergrund (z.B. die texte) arbeiten kann. Aufdringliche Farben sollten auch nur für Signal-Zwecke eingesetzt werden.
Die Entscheidung für einen Folien-Master sollte immer im Vorfeld erfolgen. Das gilt besonders dann, wenn man seine Presentation in sehr vielen Details ganz individuell gestalten will. Eine nachträgliche Änderung des Masters ist zwar jederzeit möglich – und bei sauberer Trennung von Design und Inhalt sowie wenigen zusätzlichen Veränderungen (im Design) – auch ohne größere Probleme machbar.
Vor der Änderung sollte man die alte Präsentation als Meilenstein speichern, damit man im Zweifelsfall jederzeit auf eine funktionierende Version zurückkehren kann.
Bei nachträglichen oder späten Änderungen des Master's muss mit sehr viel Nacharbeit gerechnet werden.
In anderen – Kosten-frei zu habenen – offline-Präsentationen, wie Presentation (Kingsoft Office Suite Free) oder Presentations (Softmaker Free Office), sehen die Vorlagen ähnlich aus. Die Bezeichnungen sind teilweise etwas anders gewählt. Man findet die richtigen Elemente aber schnell beim ersten Durchstöbern der Software.

Die Auswahl der Master ist in der Kosten-freien "Basic"-Version von prezi etwas eingeschränkt. Aber für erste Versuche sind es sicher genug Vorlagen.



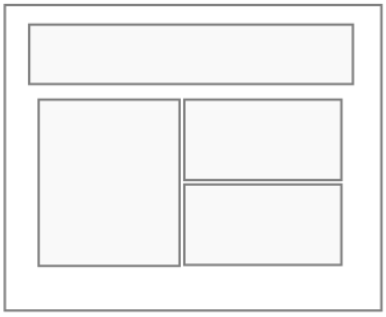
Bei allen Besprechungen darf man auch google drive mit seinem Presentations nicht vergessen. Natürlich muss amn sich an dieser Stelle unbedingt mit dem Datenschutz auseinandersetzen. Aber mit anonymiserten Accounts und dem besonderen Augenmerk auf die Inhalte (ohne Personen-Bezüge und –Daten), dann sind hier tolle Präsentationen machbar.

Layout

Unter dem Layout versteht man die Vorlage für die Anordnung von bestimmten Objekten auf der Folie. So gibt es neben leeren Layout-Vorlagen auch solche, die Textbereiche enthalten. Auf wieder anderen Layouts sind Texte und Bilder gemeinsam vorgeplant.

nur in der Editier- / Bearbeitungs-Ansicht sichtbar

Layouts in LibreOffice-Impress

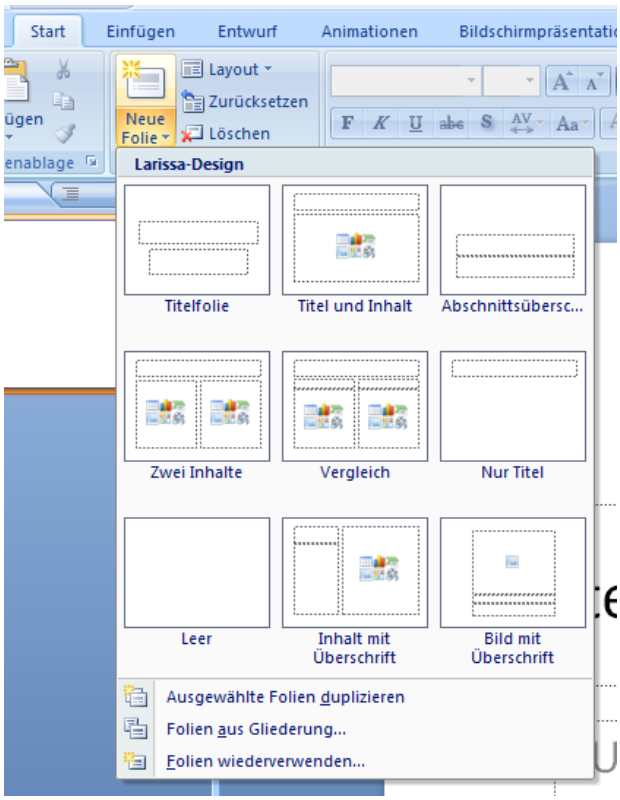


Die Layout's der verschiedenen Präsentations-Programme unterscheiden sich nur geringfügig. Deshalb besprechen wir hier die einzelnen Programme nicht.

Layout's liegen praktisch über den Folienmaster und legen vorgefertigte Ablage-Positionen für die Seiten-Objekte fest. Die Layout's haben meist einfache Namen, die an die gedachte Verwendung erinnern.

Ein falsch gewähltes Layout ist kein Problem, man löscht einfach die überflüssigen Bereiche und fügt sich neue Bereiche dazu. Diese sind mit den typischen Eigenschaften aus dem Folienmaster abgeleitet.

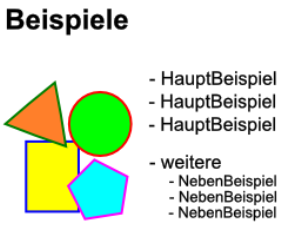
Die meisten Animationen und Effekte beziehen sich auf die Layout-Objekte. So werden z.B. Texte eingeflogen oder Bilder lösen sich auf einmal auf.



Layouts in microsoft PowerPoint

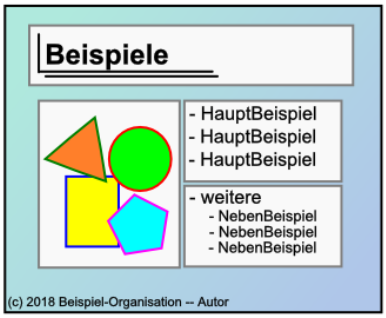
Folie

Die Folie ist die oberste Schicht. Sie ist das fein formatierte Ansicht einer Seite der Präsentation. Die Folien-Ansicht basiert auf dem Folienmaster und den genutzen Layout-Teilen. Die vorgelegten und ev. hinzugefügten Layout-Elemente sind nun mit konkreten Inhalten gefüllt. Hat men z.B. ein Layout-Element nicht genutzt, dann erscheint es auch nicht in der Folien-Ansicht.
Jede Folie als oberstes Gestaltungs-Element kann alle untergeordneten Schichten und deren Elemente ändern. In der Informatik nennt man das Überschreiben.



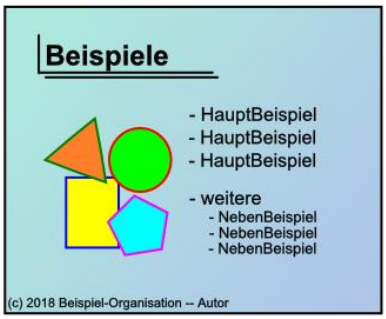
reiner Inhalt der Folie

mehr Mikro-Layout



die drei Ebenen zusammen
im Bearbeiten-Modus

Im Bearbeiten-Modus (Editier-Modus) sind alle drei Ebenen sichtbar. Man kann noch alle Details verändern. Mit dem Erstellen der Bildschirm-Präsentation verschwindet die Hilfs-Ebene Layout und wir sehen die fertige Verschmelzung von Folienmaster und Folie. Im informatischen Sprach-Jargon die Vereinigung von Design und Inhalt.



Ansicht der fertigen Folie
in der Präsentation

3.2.x. Präsentationen beurteilen

Aufgaben:

1. ***Recherchieren und dokumentieren Sie mindestens 5 umfassende (!) Sammlungen von Kriterien / Regel-Systeme od.ä. für gute / erfolgversprechende Präsentationen in einem Textverarbeitungs-Dokument!***
   (Geben Sie neben die Quellen auch das Datum der letzten Aktualisierung der Webseite und den eigentlichen Verantwortlichen für den Inhalt an (siehe z.B. Impressum)!)
   Speichern Sie die Recherche in Ihrem Home-Ordner und kopieren Sie die Datei am Schluss in den Abgabe-Ordner des Fachlehrers!
2. ***Analysieren Sie die gefundenen Regeln / Empfehlungen / Kriterien / … und erstellen Sie eine handschriftliche Sammlung aus 5 bis 10 wichtigen Aspekten zur Beurteilung von guten / erfolgversprechenden Präsentationen! Erläutern Sie die Aspekte kurz! Die Aspekt-Sammlung soll später Ihr Maßstab für die Beurteilung eigener und fremder Präsentationen sein!***

3.2.x. Projekt: Erstellen einer Themen-gebundenen Präsentation (Auftragswerk)

Der Tourismus-Verband "Mein schönes Heimatland e.V." gibt eine Präsentation (online oder offline) in Auftrag mit dem Thema "Mecklenburg-Vorpommern – immer eine Reise wert?!". Die Präsentation soll aus mindestens 6 Inhalts-Folien bestehen, wobei die Titel-Folie und die "Quellen-Verzeichnis / Impressum"-Folie nicht mit eingerechnet sind. Die Folien sollen mindestens 3 der nachfolgenden Themen allgemein vorstellen:

- ***Lage und Geographie***
- ***Natur und Umwelt***
- ***Volksgut und Kultur***
- ***Wellness und Erholung***
- ***Tourismus und Reiseziele***

Die letzte Folie vor dem "Quellen-Verzeichnis / Impressum" soll ein konkretes Angebot für eine Reise / einen Kurzurlaub / ein Hotel / ein Arrangement einschließlich Preis enthalten!
Die Präsentation darf sich an eine spezielle Interessen-Gruppe (z.B. Jugendliche, Eisenbahn- oder Natur-Freunde) wenden, muss sie aber nicht!
Da die Präsentation öffentlich genutzt werden soll, müssen die Urheberrechte unbedingt eingehalten werden (Quellen-Nachweise!) und es dürfen nur freie Bilder (cc-Lizenz, kommerziell nutzbar) benutzt werden!

Abgabe als Präsentations-Datei oder Link in einer Text-Dokument!

Hinweise zur Lizenz-gerechten Bildersuche:

- bei **google** in der Bilder-Suche "Einstellungen" "erweiterte Suche" "Nutzungsrechte" …
- freie Bilder z.B. bei **flickr.com** (nach einer ersten Suche die Lizenz auswählen!)
- …

Mecklenburg-Vorpommern – immer eine Reise wert?!	**Lage und Geographie**	**Natur und Umwelt**
Volksgut und Kultur	**Wellness und Erholung**	**Tourismus und Reiseziele**
■ ■ ■	**Angebot**	**Quellen / Impressum**

Objekte und Strukturen in Präsentationen

Wie auch in der Bild-Bearbeitung kennen wir in der Textverarbeitung verschiedene **Objekte**, denen wieder **Attribute** (Eigenschaften) und **Methoden** (Funktionen, Arbeitsschritte) zugeordnet werden.
Dieses Mal betrachten wir die Objekt-Struktur einmal von oben noch unten, also vom größten zum kleinsten Objekt. Diese Herangehensweise nennt man in der Informatik ""Top down" ("von oben nach unten").

Objekt:			**Bemerkungen / Hinweise**
Präsentation			vom Typ PowerPoint
→ Unterobjekt(e):			
• Folie			
•			
→ Attribut(e):		***Wert***	
• Dateiname	=	*Figuren*	dieses Attribut kann der Nutzer direkt ändern
• Dateityp	=	*PPTX*	
• geändert	=	*wahr*	dieses Attribut wird vom Programm verwaltet
• Schreibschutz	=	*falsch*	
• Passwort	=	*p6$Rq=%*	Passwort wird verschlüsselt abgespeichert
•	=		
→ Methode(n):			
• Neu erstellen			
• Speichern unter …			
• Speichern			
• Öffnen			
• Schließen			
•			

noch nicht angepasst!

Objekt:			**Bemerkungen / Hinweise**
Folienmaster			
→ Unterobjekt(e):			
• Layout-Objekt			
•			
→ Attribut(e):		***Wert***	
• Absatzformat			
○ Ausrichtung	=	*links*	
○ Einzug	=	*hoch*	
▪ 1. Zeile	=	*0,5 cm*	
▪ ab 2. Zeile	=	*0 cm*	
○ Rahmen	=	*falsch*	
○ Zeilenabstand	=	*1,5*	
○ Abstand vor Absatz	=	*3 pt*	
○ Abstand nach Absatz	=	*6 pt*	
○ Tabulatoren	=	*2*	
▪ Tabulator 1	=	*0,5 cm*	
▪ Tab 1 Ausrichtung	=	*links*	
▪ Tabulator 2	=	*12,5 cm*	
▪ Tab 2 Ausrichtung	=	*rechts*	
•	=		
→ Methode(n):			

• Neu erstellen	erzeugt eine neuen Absatz mit dem akt. Format
• Absatzformat setzen	
• Absatz darstellen / anzeigen	
•	

Objekt: **Bemerkungen / Hinweise**

Folie		
→ Unterobjekt(e):		
• Text		
• Tabelle		
• Grafik		
•		
→ Attribut(e):		***Wert***
• Seitenformat		
○ Seitengröße	=	*A4*
○ Orientierung	=	*hoch*
○ Randabstände		
▪ links	=	*2 cm*
▪ rechts	=	*3 cm*
▪ oben	=	*2 cm*
▪ unten	=	*2 cm*
• gefüllt / nächste Seite	=	*falsch*
•	=	
→ Methode(n):		
• Neu erstellen		erzeugt eine neue Seite mit dem akt. Format
• Seitenformat festlegen / ändern		
• Ansicht aktualisieren		
• Druckvorschau erzeugen		
•		

```
Präsentation  ::=  { Folie }

Folie  ::=  { Text  |  Tabelle  |  Grafik }

Text  ::=  { Absatz }
Absatz  ::=  { Zeichen } Zeilenumbruch
```

Objekt: **Bemerkungen / Hinweise**

Absatz		
→ Unterobjekt(e):		
• Zeichen		alle druckbaren, wie auch d. Steuer- Zeichen
•		
→ Attribut(e):		***Wert***
• Absatzformat		
○ Ausrichtung	=	*links*
○ Einzug	=	*hoch*
▪ 1. Zeile	=	*0,5 cm*
▪ ab 2. Zeile	=	*0 cm*

	○ Rahmen	=	*falsch*
	○ Zeilenabstand	=	*1,5*
	○ Abstand vor Absatz	=	*3 pt*
	○ Abstand nach Absatz	=	*6 pt*
	○ Tabulatoren	=	*2*
	▪ Tabulator 1	=	*0,5 cm*
	▪ Tab 1 Ausrichtung	=	*links*
	▪ Tabulator 2	=	*12,5 cm*
	▪ Tab 2 Ausrichtung	=	*rechts*
•		=	
→ Methode(n):			
• Neu erstellen			erzeugt eine neuen Absatz mit dem akt. Format
• Absatzformat setzen			
• Absatz darstellen / anzeigen			
•			

Objekt:

Zeichen		
→ Attribut(e):		***Wert***
• Zeichen-Code	=	*0041h*
• Zeichenformat		
○ Schriftart	=	*Arial*
○ Schriftstil		
▪ fett	=	*wahr*
▪ kursiv	=	*falsch*
▪ unterstrichen	=	*falsch*
○ Schriftfarbe	=	*rot*
○ Hintergrundfarbe	=	*keine*
•		
→ Methode(n):		
• Neu erstellen		
• Schriftart setzen		
• Schriftstil ändern		
• Schriftfarbe setzen		
• Hintergrundfarbe setzen		
•		

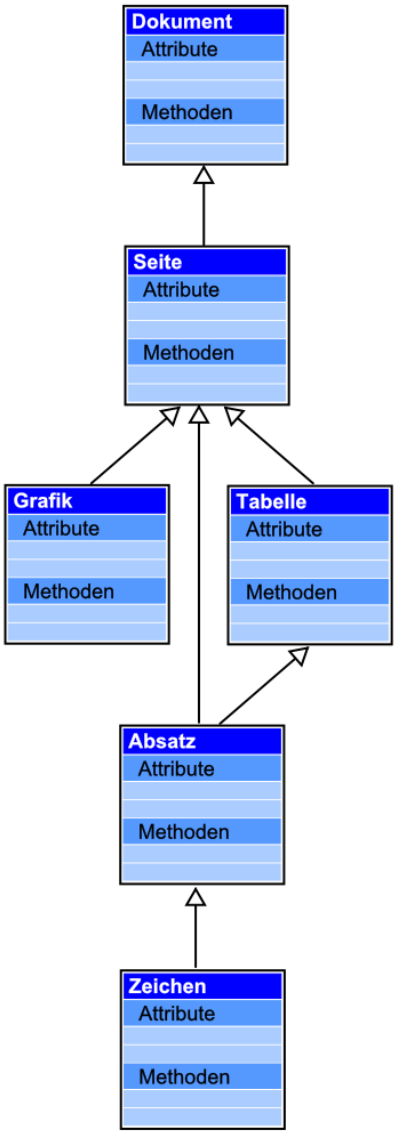
Bemerkungen / Hinweise

meint hier nur druckbare Zeichen

Windows-interner Code für "**A**"

intern auch Nummer: RGB = 255,0,0
RGB = 255,255,255 (weißer Hintergrund)

Übernahme von der Tastatur


Dokument
Attribute
Methoden
Seite
Attribute
Methoden
Grafik
Attribute
Methoden
Tabelle
Attribute
Methoden
Absatz
Attribute
Methoden
Zeichen
Attribute
Methoden

3.x. Präsentation von Zusammenhängen mit MindMap und Co

Vorstellen von MindMap und ConceptMap

Wissens-(Land-)Karten, Gedanken-Landkarten, Gedanken-Modelle, Wissens-Modelle

(freie) MindMap-Programme

-
-
-

(freie) ConceptMap-Programme

- **CmapTools**

  auch als Cloud-App verfügbar
  freie Software
  https://cmap.ihmc.us/

-
-

Vergleich von MindMap und ConceptMap

	MindMap	ConceptMap
Gemeinsamkeiten	Stern-förmige Begriffs-Struktur	
Unterschiede		beliebige Quer-Verbindungen möglich Verbindungen haben einen Namen Verbindungen können Wirkrichtungen haben
Haupt-Verwendungen	Erfassen von Ideen, Assoziationen (z.B. in einem Brainstorming) Darstellung von losen Verbindungen mehr für kreative Übersichten	Darstellung sachlicher Zusammenhänge mehr für sachliche Übersichten
Vorteile	sehr einfach sehr übersichtlich	
Nachteile	nur Stern-förmige Begriffs-Struktur	erfordert immer Festlegung der Beziehungsart

Hinweis auf Verbindung des MindMap-Konzeptes mit dem Präsentations-Konzept von PowerPoint und Co in online-Präsentations-App's wie z.B. Prezi

Objekte und Strukturen in MindMap- und ConceptMap-Dokumenten

A. Grob-Aufbau und Nutzung eines Rechners

Herstellen der Betriebsbereitschaft des Rechners und Interaktion mit dem Gerät

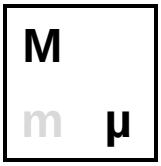
A.1. Maus und Tastatur

Bereiche einer klassischen Tastatur

Funktionen der n

Großbuchstabe wird beim gleichzeitigen Drücken mit der Hochstell-Taste ausgewählt

normale Tasten-Belegung (Kleinbuchstabe)



Graphik-Zeichen mit [Alt Gr]-Taste

spezielle Tasten

[Enter] Ausführungs-Taste / Bestätigungs-Taste (zum Kennzeichnen, dass eine Eingabe jetzt vollständig ist); in vielen Programmen Zeilen-Umbruch oder Eingabefeld-Wechsel

[ESC] Abbruch (z.B. bei Eingaben; Rücksetzen auf den vorherrigen Zustand)

[] Tabulator-Taste (Tab-Taste) zum Springen von Tabellen-Zelle zur nächsten Zelle oder Springen zur nächsten Eingabestelle oder zur nächsten Bedienfläche (Schalter, …)

[] Feststell-Taste / Einrast-Taste für das dauerhafte Eingeben von Großbuchstaben (Zeichen der 2. Tastatur-Ebene

Nummern-Block
damit Ziffern und Rechen-Operatoren benutzt / eingegeben werden können, muss der Nummern-Block aktiviert sein
das erkennt man daran, dass ein Lämpchen leuchtet, dass meist mit einer Nummer (1) beschriftet ist
ansonsten dient der Nummernblock der Bewegungssteuerung (u.U. auch als Maus-Ersatz
es gelten die Tasten-Beschriftungen, die links unten auf der Taste vermerkt sind

erweiterter –Block
diverse Steuerungs- und Bewegungs-Tasten

[Einfg] schaltet zwischen Einfüge- und Überschreib-Modus um

10-Finger-Tippen

Funktions-Tasten-Reihe
diverse Tasten der F-Reihe haben in vielen Programmen ähnliche Bedeutungen / Verwendungen
in neueren Programmen und Programm-Versionen aber auch schnell mal neu belegt

[F1] ruft üblicherweise die Hilfe zum Programm auf

weitere Kombinations-Möglichkeiten mit Sonder-Tasten

[Strg] + [ ] Steuerungs-Befehle (noch in einigen Programmen benutzt; historisch)

[Alt] + [ ] Menü-Befehle (in Programmen zur Funktionsauswahl (statt mit der Maus))
bei Profi's und Vielschreiber gerne benutzt; einige Tasten-Kombinationen sind universell

[⊞] + [ ] Windows-System-Kommandos (z.B. Aufrufen des Start-Menü's usw. usf.)

Tastaturen von Laptop's usw.

oft ist hier die Funktions-Tastenzeile eingespart worden, dafür gibt es eine zusätze [Fn]-Taste, die mit anderen (meist blau gekennzeichneten Tasten) die Funktions-Tasten nachbildet

zusätzliche Steuerungs-Tasten zum Umschalten der angeschlossenen Monitore, der Bildschirm-Helligkeit usw.

A.2. Touch-Pad und Touch-Bildschirme

einfacher Finger-Tipp (nur) auswählen

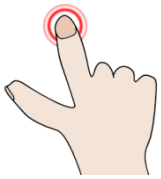
doppelter Finger-Tipp starten / aktivieren

wischen bewegen (z.B. Weiterblättern, kleinere oder größere Werte auswählen usw. usf.)

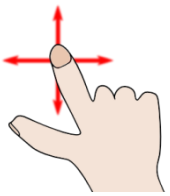
diagonales Spreizen der Finger (üblicherweise von Daumen und Zeigefinger) Vergrößern des Bildschirm-Ausschnittes

diagonales Zusammenziehen der Finger (üblicherweise von Daumen und Zeigefinger) Verkleinern des Bildschirm-Ausschnittes

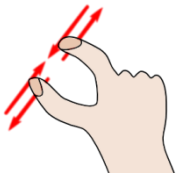
simuliert / bedeutet Maus-"Rechts-Klick"



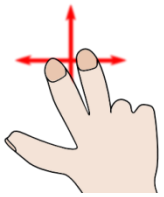
ev. nur TouchScreen



simuliert / bedeutet Maus-"Rechts-Klick"



ev. nur TouchPad



ev. nur TouchPad

ev. nur TouchPad

A.3. Sprach-Steuerung

meist mit Kommando-Wort gestartet (z.B. "google", "aleksa", …
danach dann Ansagen bestimmter Kommando-Worte oder Fragestellungen

Achtung! Damit diese Funktion nutzbar wir, muss das Gerät und die Sprach-Erkennung (also das Mikrophon) ständig auf Empfang sein. Es werden also auch alle anderen Geräusche, Gespräche usw. aufgenommen. Eine missbräuchliche Nutzung durch Dritte ist nicht sicher auszuschließen.

A.4. Gesten-Steuerung

hauptsächlich für die Verwendung in Computerspielen

B. Computer-Netze

Verbund von Rechnern
gemeinsame Nutzung von bestimmten Geräten oder Leistungen
z.B. Drucker, Internet-Zugänge, Datenspeicher

in Schulen sind meist alle Computer – außer die der Schulverwaltung – in einem Computernetz zusammengefasst. Die Verwaltungs-Rechner bilden meist ein Extra-Netz.

besonders deutlich wird die vernetzung der Computer in Computer-Kabinetten. Dort reicht ein Drucker für alle Computer aus. Vielfach kann der Lehrer auch die Schüler-Rechner steuern und z.B. von seinem Rechner aus die Monitore überwachen. Alle Rechner haben auch einen gemeinsamen Internet-Zugang.

nicht das Strom-Netz gemeint, sondern das Daten-Netz
Verbindung über speziellen Netzwerk-Stecker RJ-45, die immer an den Kabeln verbaut sind, die Geräte – auch die Koppel-Geräte – haben immer Buchsen
vorsichte Handhabung der Stecker, vor allem beim Abkoppeln / Herausziehen. Plastik-Nase muss dazu heruntergedrückt werden. Erst dann lässt sich der Stecker leicht aus der Buchse ziehen. Entfernen mit Gewalt zerstört die Nase. Stecker, die defekte Nasen haben und beim Einstecken nicht richtig einrasten haben Wackel-Kontakte und sind im Allgemeinen auch langsamer
Verlängerungen usw. unüblich

Koppel-Gerät für mehrere Computer heißen Switch oder Router. Der Router verbindet die vernetzten Geräte auch noch ins Internet
Zwei einzelne Computer lassen sich auch über ein einzelnes Netzwerk-Kabel verbinden. Dann aber nur Datenaustausch zwischen diesen beiden Computern möglich. Für Internet ist dann ein zusätzlicher Netzwerk-Anschluss notwendig. Diese Anordnung ist aber eher selten.

Gefahren, das ein Nutzer die Geräte-Eistellungen verändert oder ein Gerät austauscht, ohne dass es die anderen Nutzer wissen
versehentliches Löschen fremder Daten
deshalb haben in Computernetzen immer bestimmte Personen extra Aufgaben
allgemein Administratoren (Admin's) genannt, haben praktisch alle Rechte, d.h. sie können alle Arbeiten und Veränderungen ausführen
andere Nutzer haben eingeschränkte Rechte

damit klar ist, wer gerade mit dem Computer muss am sich in einem Netzwerk anmelden – einloggen, wie es im Computer-Jargon heißt.

B.1. Login - Anmeldung in Computer-Netzen

Einhaltung von Konventionen zum Umgang mit dem Schulrechnern

besteht klassischerweise aus einem Nutzernamen und einem Kennwort

manchmal ist das Kennwort durch eine PIN, ein Wischsymbol oder den Fingerabdruck ersetzt

B.1.1. Logout – muss das sein?

Anmeldung noch offen, feindliche Nutzer können u.U. Schaden anrichten

B.2. Verhalten in Computer-Netzen

keinen Schaden anrichten
die Geräte und Ressourcen des Netzes nur solange nutzen wie nötig
keine Ressourcen verschwenden oder mißbrauchen
keine fremden Account's benutzen
fremde Daten sind taboo
Manipulation / Verstellen von Geräten ist nur in Absprache mit dem Netz-Verantwortlichen (Administratoren) zulässig
Unregelmäßigkeiten, ständige Warnmeldungen oder Fehler sofort melden
keine Fremd-Software (keine Installationen!)

C. Grundprinzipien der Informatik

C.x. Hard- und Software

Hardware

Merksätze: Hardware
Unter Hardware verstehen wir alle greifbaren / harten Teile eines Computers.
Hardware sind die materiell-technischen Teile eines Datenverarbeitungs-Systems.

Software

weiche Ware
quasi die Seele des Computers
Programme sagen dem Computer, was er machen soll
Daten sind die Software-Teile, die von den Programmen (und damit dem Computer) verarbeitet werden

Merksätze: Software
Die Software sind die nicht-greifbaren Anteile eines Computers, die aber für dessen funktionieren notwendig sind. Zur Software gehören (Computer-)Programme und Daten.

C.x.2. große und kleine Computer

PC

Notebook und Co

Tablet, Smartphone und Co

Mini-Computer (Raspberry Pi und Co)

der Raspberry Pi als ein erster persönlicher Arbeits-Computer

Ein erster eigener Computer könnte z.B. ein "Raspberry Pi 3" sein. Diese Computer gibt es auch als fertige Laptop's zu kaufen.
Günstiger ist eine Eigenbau-Variante unter Nutzung eines alten PC-Monitor's oder des Fernseher's als Anzeige-Gerät. Dann braucht man nur noch eine Tastatur, eine Maus, ein Netzteil und ein paar Kabel und schon hat man einen richtig guten Schul-Computer.

fertig kaufbare Laptop auf der Basis eines Raspberry Pi
Q: de.wikipedia.org ()

Eine schnelle microSd-Karte benötigt man für das Betriebssystem und die Datenspeicherung. Das Gehäuse zum Schutz kann man sich auch alleine bauen oder 3D-drucken.
Es sei gleich gesagt, zum Spielen eignet sich dieser Computer nur bedingt! Aber man kann sich diesen Computer selbst zusammenbauen, relativ gut hin und her transportieren und die meisten schulischen Aufgaben auf ihm erledigen. Im Internet findet man tausende Projekte, wozu man den Raspberry Pi noch benutzen kann.

Die Informatik-Lehrer, eine Schülerfirma oder schulische Arbeitsgemeinschaft unterstützen Euch bestimmt bei Euren ersten Schritten.
Alternativ gibt es auch noch andere Kleinst-Computer-System. Sie sind nicht ganz so verbreitet. Einige sind deutlich Leistungs-stärker als der Raspbery Pi, andere sind aber auch deutlich schwächer. Die schwächeren sollte man als Schul-Computer nicht unbedingt ins Auge fassen, auch wenn sie vielleicht ein paar Euro billiger sind.

Der "Raspberry Pi zero" und der "Raspberry Pi zero wifi" sind als einfache Kleinst-Rechner auch möglich, aber eben recht langsam beim Arbeiten.
Zu den Leistungs-stärkeren System gehören "UDOO x86", "ODROID", "Banana Pi" oder der "Libre Computer".

Die meisten Kleinst-Computer arbeiten mit Linux – einem kostenlosen Betriebssystem. Dieses muss meist nur auf einer microSD-Karte abgespeichert werden. Wenn diese Karte dann in den Einplatinen-Rechner eingesteckt wird, kann es meist schon direkt losgehen.
Die "Libre Computer" setzen das Konzept der freien Software (→ LibreOffice) auf der Ebene der Hardware fort. "Libre Computer" sind völlig Lizenz-frei und können von jederman gebaut und genutzt werden.

Aufgaben:

1. *Suche bei "Blinde Kuh" nach "Raspberry Pi Projekte"! Welche Projekte findest Du von den Suchergebnissen her interessant?*
2. *Informiere Dich, welche Möglichkeiten und Hilfen es an Deiner Schule zur Nutzung von Raspberry Pi's bzw. ähnlichen Kleinst-Computern gibt!*
3.

C.x. graphische Nutzungs-Oberflächen – der Desktop

Problem-Fragen für Selbstorganisiertes Lernen
Was ist ein Betriebssystem? Welche Betriebssysteme gibt es für welche Geräte? Welches ist das beste Betriebssystem? Kann man Betriebssysteme einfach welchseln?

Wenn man auf dem Bildschirm so etwas sieht, dann fragt man sich schon, wass will der Computer nun von mir und warum macht er nicht weiter?
An dieser Stelle muss ein Computernutzer einen Befehl – auch Kommando genannt – eingeben. Nach der Bestätigung hat der Computer diesen Befehl ausgeführt oder eine Fehlermeldung ausgegeben.
Der nebenstehende Befehl zeigt zum Beispiel an, was genau auf dem Computer gespeichert ist. Mann nennt diese Eingabe-Möglichkeit auch Kommandozeile, Konsole oder Shell.
Sie wird heute noch von Computer-Profi's benutzt, um bestimmte Aufgaben zu erledigen. Ein Beispiel sind Aufräumarbeiten nach einem Viren-Befall.
Im Wesentlichen muss man die Befehle auswendig kennen. Besonders wichtig ist auch die exakte Schreibung und z.B. das genaue Einhalten von Leerzeichen. Die Fehlerhäufigkeit ist recht groß und auch Fehlbedienung sind schnell möglich.
Ein echter Fortschritt waren dann schon kleine Menü-orientierte Programme, mit denen man die wichtigsten Tätigkeiten auswählen konnte. Jetzt braucht man nicht mehr wissen, wie die Kommandos geschreiben werden – man muss sie nur noch aus den verfügbaren Menü's auswählen. Einige dieser Programme sind auch schon mit der Maus bedienbar gewesen.
Heutige Computer sollen im Großen und Ganzen mit der Maus bzw. mit Sprach- oder Touch-Befehlen bedienbar sein.

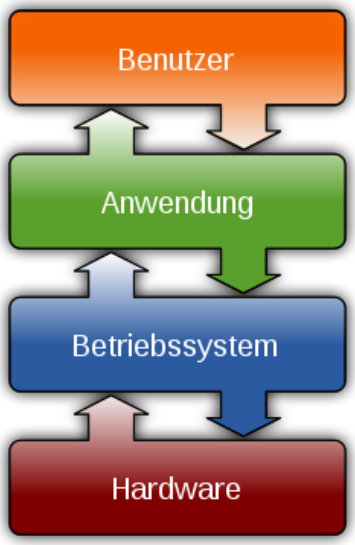
C.x.1. Was sind Betriebssysteme?

effizienter Umgang mit der Bedienoberfläche und den Anwendungen

Schichten innerhalb eines Computer-Systems
hohe Spezialisierung

Vorteile:
jeder kann das machen, was er am Besten kann,
z.B. Hardware produzieren und / oder zusammenstellen
Software entwickeln

Nachteile:
wenn es nicht funktioniert, dann ist Fehlersuche / Fehlerzuweisung schwierig
z.B. bei Garantie-Ansprüchen



Zusammenhang zwischen Betriebssystem, Hardware, Anwendersoftware und dem Benutzer
Q: de.wikipedia.org (Golftheman)

Merksätze: Betriebssystem
Ein Betriebssystem ist die Software, die einen Rechner für einen Nutzer benutzbar macht.
Das Betriebssystem ist die Software, die dem Nutzer eine sinnvolle Verwendung der Hardware und die von Anwender-Programmen ermöglicht.

wichtige Betriebssysteme (ganz-ganz-kurze Geschichte)

C.x.1.1. mircosoft Windows

Maus-Tätigkeiten

Links-Händer können sich ihre Maus passend einstellen. Dann wir eben aus dem "Rechts-Klick" eine "Links-Klick". Da es leider keine allgemeine Bezeichnung für die Klicks gibt, bleiben wir bei der üblichen "Rechtshänder"-Benennung.

wichtige Tastatur-Befehle – nicht nur für den Notfall

C.x.1.2. apple macOS und iOS

C.x.1.3. Linux – die kostenfreie Alternative

microsoft ® Windows ® und die apple-Betriebssystem sind sogenannte proprietäre und kommerzielle Programme. Weil die Software der Firma microsoft ® gehörte war der innere Aufbau des Programm versteckt und eine benötigte Benutzungs-Lizenz kostete Geld.
Der Student Linus TORVALDS hatte nicht das notwendige Geld für ein Betriebssystem und er wollte auch mehr über die Funktionsweise von Betriebssystem wissen. Er interessierte sich für das Betriebssystem UNIX.
Er entwickelte 1991 ein Betriebssystem, das er Linux nannte. Der Quell-Code des Programms wurde von ihm veröffentlicht und jeder konnte es kostenfrei nutzen.
Linux ist ein Betriebssystem, dass sich für kleinste und größte Rechner-System gleichermaßen eignet. Es kann in sehr weiten Zügen angepasst und verändert werden. Jeder kann sich an der Entwicklung beteiligen und die zugehörigen Programme verbessern.

Vertrieb als Distribution
Zusammenstellung verschiedener Programme
Firmen leben vom Support

aktuelle / große / dedeutende Distributionen

C.x.1.4. Android – Linux für Smartphone's

basiert auf Linux
hauptsächlich von google-Konzern entwickelt
ist aber trotzdem frei nutzbar und auch Quellen-offen
orientiert sich an kleineren Geräten, wie Smartphone's, Tablet's, Netbook's und Mediaplayer

praktisch läuft auf dem Bildschirm nur eine App (ein Programm), weitere Programme können im Hintergrund arbeiten; Umschalten zwischen den App's ist jederzeit möglich

Benennung der Versionen nach Süßspeisen, berühmt sind die Versionen "Jelly Bean" und "Lollipop" geworden
Version 7 heißt z.B. "Nougat"

wenige Eingabe-Tasten und ein Touch-Bildschirm; auf neuesten Geräten nur noch Touch-Bedienung

Bedien-Knöpfe:
"zurück"
"Home"
"letzte Applikationen" (App-Liste, Multitasking-Button)
neu jetzt langes Drücken der "App-Liste" führt zum Teilen des Bildschirms und ermöglich gleichzeitige Nutzung von zwei Apps

Touch-Bedienung

Unterscheidung nach der Anzahl der ausgewerteten Berührungs-Punkte
hauptsächlich für die Bedienung mit Finger'n gedacht, oft auch feinere Bedienung mit speziellen Stiften möglich (z.B. Schreib-Notizen)

Multi-Touch

kurzes Drücken / Tippen → auswählen / aktivieren
langes Drücken →
wischen → bewegen / scrollen / verstellen

drücken und wischen → verschieben
zwei-Finger-auseinanderziehen / -zusammenschieben → vergrößern / verkleinern
zwei-Finger-drehen → drehen

neue Touch-Systeme erkennen unterschiedliche Druck-Stärken oder das Drübergleiten eines Fingers

Statuszeile
zumeist zweifarbige Informations-Zeile an der Oberseite des Desktop's
enthälte wichtige Geräte- und Verbindungs-Informationen

hier Informationen zu Bluetooth-Verbindung, Klingel-Modus, Empfangs-Stärke, WLAN-Verbindung, Batterie-Ladezustand, Uhrzeit

ev. auch Anzeige "Flug-Modus" (eingeschränkte Verbindungen)

Statuszeile eines Android-System's
(Version 7 (Nougat))
Q: de.wikipedia.org (google)

wird aber hauptsächlich direkt auf einer Hardware (z.B. einem Smartphone oder einem Tablet) installiert vertrieben
meist sehr stark an die Hardware angepasst und deshalb schwieriger auzutauschen oder zu erneuern (upzudaten)

Software-Vertrieb hauptsächlich über App-Store
neben vielen kostenfreien App's gibt es auch Bezahl-App's
von vielen App's gibt es beide Versionen, die kostenfreie Version enthält meist Werbung oder man "bezahlt" mit seinen persönlichen Daten; zumeist ist der Funktions-Umfang eingeschränkt
App-Store wird von google kontrolliert

Marktanteil z.B. bei Smartphone's über 85 %.
weitere Betriebssysteme für Smartphone's sind iOS, Windows Phone und Blackberry OS

Android gilt als das am schlechtesten gesicherte Betriebssystem im Bereich der Tablet's und Smartphone's
unbedingt zusätzliche Sicherheits-Maßnahmen notwendig
z.B. Installation eines Viren-Scanner's und einer Firewall

C.x. der eigene / die eigenen Arbeits-Rechner

Die meisten von uns haben schnell mit mehreren Rechnern zu tun. Einer zuhause, ein weiterer in der Schule und dann ist da ja noch der, den wir praktisch immer dabei haben – unser Smart-Phone. Das ist hier keine Argumentation, das Smart-Phone ständig bei sich zu haben und immerzu drauf zu schauen. Vielmehr geht es um eine sachgerechte Nutzung, dann, wenn man das Gerät braucht!

C.x.y.1. Welche Software brauchen wir?

C.x.y. portable Software

C.x.y.0. Prinzip der portablen Programme / Apps

Warum ist das eigentlich immer so kompliziert? Erst muss man eine Software downloaden, dann installieren und letztendlich endlich nutzen. Das alles kostet Zeit und ist auch nicht ganz problemlos. Will man die Programme auf einem anderen Rechner nutzen, dann geht der ganze Aufwand u.U. wieder von vorne los.
Die Schwierigkeit liegt in der Zusammenarbeit des Programms mit dem Betriebssystem. Das geht z.B. schon los, wenn man im Windows-Explorer (oder "Arbeitsplatz" / "Computer") eine bestimmte Datei zum Bearbeiten anklickt. Wir erwarten, dass dann das passende Programm automatisch startet und die Datei öffnet. Klicken wir z.B. auf eine DOCX-Datei, dann soll sich automatisch WORD öffnen, wenn es eine XLSX-Datei war, dann eben EXCEL. Damit das klappt, hat das Installation-Programm (auch Setup genannt) dem Betriebssystem genau diese Zuordnung mitgeteilt. Auch andere Informationen zum Programm sind beim Installieren registriert worden. Dazu gehören z.B. die verfügbaren Leistungen für andere Programme oder bestimmte Einstellungen (Farb-Schemata, ein- und ausgeschaltete Funktionen, …).
Den Luxus des unkomplizierten Arbeitens erkaufen wir uns mit dem Aufwand der Installation. Das Kopieren von Programm-Ordnern mit installierten Programmen von einem Rechner auf einen anderen funktioniert deshalb auch nicht. Dem dortigen Betriebssystem müssen auch wieder alle Mitteilungen gemacht werden. Das Programm muss also auf dem anderen Rechner wieder neu installiert werden.
Wäre es nicht toll, die Programme einfach von einem Rechner auf den anderen zu kopieren oder gar auf einem USB-Stick oder einem ähnlichen Datenträger – mitzunehmen?
Genau das haben sich die Programmierer von übertragbarer Software als Ziel gesetzt. Sie verzichten auf die Registrierung ihrer Software beim Betriebssystem. Alle Einstellungen werden nur lokal gespeichert. Damit kann so ein Programm einfach an einen anderen Ort kopiert werden, es funktioniert trotzdem.
Viele Programme gibt es als portable Version. Oft muss man auf den Download-Seiten nur ein bischen weiter scrollen.
Die portablen Versionen kommen meist als ZIP-Datei. Man muss sie nur noch am gewünschten Ort entpacken und die enthaltene EXE starten.

Nachteile sind die fehlende Dateityp-Verknüpfung im Betriebssystem und das u.U. recht aufwändige Suchen der EXE irgendwo auf dem Rechner. (Mit einer Desktop-Verknüpfung kann man sich z.B. helfen.)

C.x.y.1. das "Portable Apps"-Menü-System

Die Macher des "portable Apps"-Menü-System gehen nun noch einen Schritt weiter. Sie bündeln die verschiedenen Programme in einem speziellen Menü. Es ähnelt sehr dem Start-Menü von WINDOWS – ist aber davon unabhängig.
Alle portablen Programme werden – nach verschiedenen Kategorien sortiert – aufgelistet und können aus dem Menü heraus gestartet werden.
Das Menü-System und die portablen Programme (Apps) können gemeinsam auf einem USB-Stick gespeichert werden und an jedem WINDOWS-Rechner benutzt werden. Nur die USB-Anschlüsse müssen funktionieren. Praktisch gehen auch ältere Windows-Versionen. Wegen der Sicherheits-Bedenken sollte man dies aber nur in Ausnahmefällen tun.

Mittlerweile ist das Menü-System so weit entwickelt, dass es beim Start den Apps-Ordner nach neuer Software scannt und die bekannten Programme auf Updates überprüft. Man kann sich gleich entscheiden, ob man Aktualisieren möchte, oder nicht.
Man kann das portableApps-Menü auch einem "normalene" Rechner installieren. Dann lassen sich die Vorteile der schnellen Aktualisierung auch auf dem "normalen" PC nutzen.

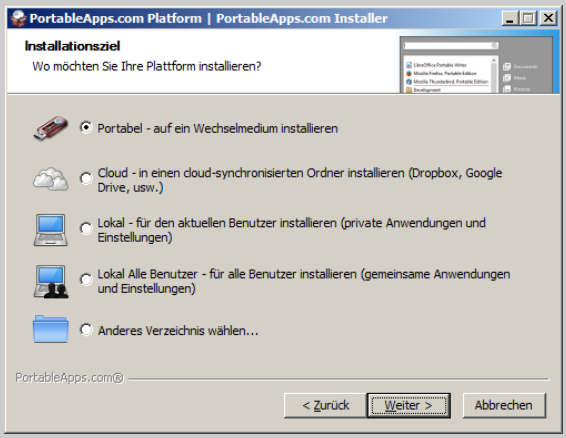
Exkurs: Wie richte ich einen USB-Stick für portableApps ein?

1. Auswahl eines USB-Stick's

Ob USB-Stick, Speicher-Karte oder externe Festplatte – alle mobilen und beschreibbaren Datenträger eignen sich für das Portable-Apps-System. (Auch die eigene Festplatte kann genutzt werden.) Wenn man die Wahl hat, sollte ein möglichst schneller Datenträger genutzt werden. Bei einem USB-Stick sollte es also USB 3.x sein. Alle Größen ab 4 GB sind geeignet. Man sollte hier aber über mindestens 8 – besser aber über 16 oder 32 GB nachdenken. Schließlich sollen neben den Programmen auch noch reichlich Daten gespeichert werden.
Wer schon eine funktionierende Installation hat und diese nun auf einem USB-Stick etc. weiter nutzen möchte, der kann jetzt gleich bei Punkt **5.** weitermachen.

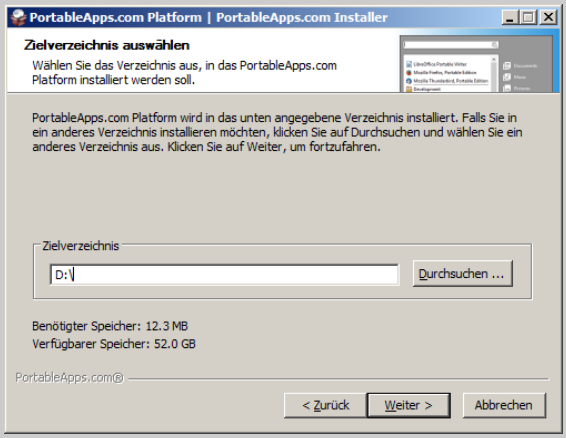
2. Download des portableApps-Installations-Programm

Nun lädt man sich von der Internet-Seite http://www.portableapps.com die aktuelle Version des Installations-Programms herunter. Entweder speichert man die Datei auf dem Desktop oder im Download-Ordner (hierher wird sie oft automatisch gedownloadet).



3. Installation des portableApps-Menü's

Das Installations-Programm führt kleinschrittig durch den Einrichte-Vorgang. Für eine Installation auf dem USB-Stick wählt man die entsprechende Option (Abb. rechts oben). Man kann aber auch später noch den Installations-Ordner angeben und dabei ist auch die Angabe des USB-Stick's kein Problem (Abb. rechts unten).
Die Einrichtung kann praktisch nicht schief laufen. Nach der Installation sollte man den Datenträger im Windows-Explorer od.ä. öffnen und die "Start.exe" auswählen. Wenn das Menü startet, dann ist alles bereit zur Aufnahme / Integration von portablen Programmen.



4. Hinzufügen von Programmen zum portableApps-Menü

a) Hinzufügen von direkt unterstützten Programmen

Beim ersten Starten des Menü-Systems oder über den Menü-Punkt "" kann man sich nun aus einer großen Software-Auswahl bedienen. Alle Software hier ist frei nutzbar. Man wählt die gewünschten Programme einfach aus und startet die Installation. Bei ersten Installationen wird oft nach dem Einverständnis hinsichtlich der Lizenz-Bedingungen gefragt. Da muss man einmal durch.
Es sollte nur solche Software ausgewählt werden, die man wirklich braucht. Der Platz auf dem USB-Stick ist ev. ja begrenzt. Bei ähnlichen Programmen, wie OpenOffice und LibreOffice, sollte man sich für eins entscheiden.
Die über dieses System installierten Programme werden täglich bzw. beim nächsten Start des Menü's auf Updates überprüft.

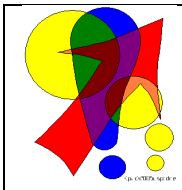
b) Hinzufügen / Integration beliebiger portabler Programme

Andere Programme, die als portable Version verfügbar sind, können direkt auf den USB-Stick kopiert / entpackt werden. Für jedes Programm sollte ein eigener Ordner im Programm-Ordner "portApps" angelegt werden. Das erleichtet späteres Aktualisieren oder Löschen.
Aktualisierungen müssen bei diesen Programmen immer manuell vorgenommen werden.
Beim Starten des Menü-Systems scannt dieses die Unterordner von "portApps" auf EXE-Dateien. Alle gefundenen EXE-Programme werden dann ins Menü-System aufgenommen.

5. Vervielfältigen einer portableApps-Installation

Besitzt man schon irgendwo eine funktionierende portableApps-Installation, dann kann diese direkt auf einen anderen Datenträger kopiert werden. Viele Schulen usw. stellen ihren Schülern schon fertig eingerichtete portApps zur Verfügung. Es empfiehlt sich das gesamte System immer direkt in das Wurzel-Verzeichnis des Datenträgers zu positionieren – notwendig ist das aber nicht. Man findet die "Start.exe" dann aber am Schnellsten.

Hinweise:
Ein zu portableApps ähnliches Programm-System ist **LiberKey**. Für den reinen Schul-Einsatz mit dem Schwerpunkt Informatik bietet sich auch der sogenannte **Io-Stick** von T. HEMPEL an. Hier sind nur mit größerem Aufwand Programme nachträglich integrierbar und eine Aktualisierung gibt es ungefähr einmal jährlich.
Das portable-Apps-Menü und das Io-Stick-System lassen sich parallel auf einem Datenträger nutzen. Man sollte aber dann auch mindestens 16 GB Speicher zur Verfügung haben (absolutes Minimum sind 4 GB, dann ist aber auch kaum noch Platz für weitere Programme und Daten!).
Vom Stick sollte man in regelmäßigen Abständen (mindestens einmal pro Woche, besser täglich) ein Backup machen. Das betrifft vorrangig die eigenen Daten.
Für Backup's fast aller Art empfehle ich das Programm **PersonalBackup** (von J. RATHLEV). Das Programm ist unwahrscheinlich flexibel, sehr schnell und Fehler-tolerant.



Bemerkungen zu "versteckten" / besonderen Start-Programmen:
Einige portablen Programme nutzen Batch-Dateien oder in irgendwelchen untergeordneten Verzeichnissen liegende Start-Dateien. Diese findet das portableApps-Menü nicht.
Als Abhilfe habe ich ein universelles Start-Programm geschrieben. Es heißt **Uni-Starter** und kann unter http://www.lern-soft-projekt.de/ gedownloadet werden.
Das Programm legt beim ersten Starten eine DAT-Datei an, die jederzeit im Editor angepasst werden kann. Die EXE- und die DAT-Datei können jederzeit gleichnamig umbenannt werden.
In der EXE kann man ebenfalls die notwendigen Optionen für die zukünftigen Start's setzen. Geht das nicht, dann reicht das Löschen der DAT-Datei aus. Beim nächsten Start des UniStarter muss man dann die Einstellungen völlig neu vornehmen. Danach gibt es dann wieder eine aktuelle DAT-Datei.
Die beiden Dateien müssen dann im Programm-Odner positioniert werden, damit das Menü-System sie beim nächsten Start findet. Hat man mehrere Programme, die mit UniStarter gestartet werden sollen, dann sollte man unbedingt eine Umbenennung nutzen. Verzichtet man hierauf, hat man den UniStarter mehrfach im Menü (für jedes zu startende Programm einmal).
Ev. erscheint bei einem Start über den UniStarter ein kleines schwarzes Fenster. Das kann man Ignorieren oder Schließen. Am Ende verschwindet es automatisch.

D. Daten – Was ist das?

Öffnen und Speichern einer Datei unter Verwendung eines Ordnungssystems

Suchen unter Vewendung von Techniken zur Eingrenzung der Suchergebnisse

Bewerten der Suchergebnisse

Beachten des Urheberrechts, des Rechts am eigenen Bild und der Persönlichkeitsrechte anderer

Einfügen von Objekten aus Fremdanwendungen in Dokumente

D.1. Was sind Daten?

Definition(en): Informations-Einheit bit

Definition(en): Daten

Definition(en): Information

D.2. Wie wichtig sind meine Daten?

Für die Erzeugung von Daten haben wir immer Arbeitszeit aufgewendet, das heißt auch, die Daten gibt es nicht umsonst. Später bezahlt uns der Arbeitgeber für die erzeugten Daten oder wenn man selbstständig ist, dann zahlen die Kunden. Aber immer kosten Daten etwas.
Solange man nur kleine Texte schreibt oder mal so ein kleines Bildchen malt, der wird einen Verlust noch gut verkraften Wenn man aber viele Stunden an einem besonderen Bild oder einem langen Text gearbeitet hat, dann ist ein Verlust dieser Daten nicht mehr zu akzeptieren. Wir müssen die Daten unbedingt speichern.

Besonders wichtige Daten sollte man auch mindestens zweimal speichern. Wenn ein Speicher-Gerät kaputt geht, dann sollten wir noch irgendwo eine nutzbare Kopie der Daten haben. Wie man Daten recht sicher speicher, schauen wir uns gleich an.

D.3. Wie werden meine Daten gespeicher!

Prinzip der Datenspeicherung auf Computern: digital
es gibt nur die Zustände (für die Elektrik / Elektronik): Spannung vorhanden bzw. keine Spannung vorhanden → AN oder AUS (ON / OFF)
universeller ist die Bezeichnung der beiden Zustände mit O und I oder heute fast ausschließlich nur noch 0 und 1 benutzt

ein Speicher-Element / ein Verarbeits-Element kann immer nur 2 Informationen speichern, eben entweder 0 oder 1

0

1

Eine Speicher-Einheit wird bit genannt. Als Abkürzung für die Einheit nach Zahlenwerten wird das kleine b verwendet. Da es aber häufig zur Verwechslung mit der Einheit B für Byte kommt, schreibt man lieber bit aus.

durch Kombination von Speicher- / Verarbeitungs-Elementen kann man größere Datenmengen verarbeiten kombiniert man 2 Speicher- / Verarbeitungs-Elemente, dann gibt es schon 4 Belegungs-Möglichkeiten

0	0
0	1
1	0
1	1

Aufgaben:

1. *Wieviele verschiedene Belegungs-Möglichkeiten gibt es bei drei Speicher- / verarbeitungs-Elementen! Notiere sie alle!*
2. *Erstelle jeweils eine Tabelle für die Belegungs-Möglichkeiten von vier und von fünf Speicherzellen!*
3. *Ermittle, wieviele Belegungs-Möglichkeiten es bei 4, 5, 6, 7, 8, 9 und 10 Verarbeitungs- / Speicher-Elementen gibt!*
4. *Ein Mini-Computer in einem Elektronik-Baustein arbeitet mit 6 bit. Wieviele unterschiedliche Informationen kann der Mini-Computer verarbeiten / speichern?*
5. *Ein Hersteller von Elektronik-Bausteinen muss 29 verschiedene Informationen (Zustände) verarbeiten. Welche bit-Zahl empfiehlst Du ihm für sein System? Begründe Deine Wahl!*

für Experten und zum Knobeln:

6. *Wieviele Speicher-Belegungen sind für 16 bit-Systeme möglich?*
7. *Nach welchem Rechen-Schema lässt sich die Zahl der Speicher-Möglichkeiten für jede beliebige Bit-Länge berechnen?*

Früher benutzte man einige wenige Jahre 4-bit-Systeme. Man ging dann schnell auf 8 bit über, weil man einfach mehr Daten verarbeiten wollte als nur Zahlen. Acht Bit sind bis heute einer der magischen Größen in der Datenverarbeitung. Acht Bit werden auch als ein Byte bezeichnet.

Modernen Systeme enthalten immer 8-bit-Verarbeitungs-Einheiten oder Mehrfaches dieser Größe. So findet man je nach Ansprüchen der Anwender 16-, 32-, 64- oder 128 bit-Systeme. PC's arbeiten fast ausschließlich mit 64 bit. Ältere verfügen nur über 32-bit-Verarbeitungs-Systeme. Dazu gehören auch viele Tabletts.

Alle Systeme sind prinzipiell abwärts-kompartibel. D.h., dass z.B. ein 32-bit-System auch mit Daten und Programmen aus der 8- und 16-bit-Welt arbeiten kann. U.U. ist ein 32-bit-System sogar in der Lage vier 8-bit-Programme parallel laufen zu lassen.

Irgendwie muss man nun diesen Bit-Mustern Bedeutungen zuordnen. Neben einem rein nummerischen Wert (→) gibt es noch eine alphanummerische Bedeutung. In dieser sind diversen Bit-Mustern auch Buchstaben und besondere (Steuer-)Zeichen zugeordnet. Als Bedeutungs-Tabelle hat sich vor vielen Jahrzehnten die **ASCII**-Kodierung durchgesetzt. ASCII steht dabei für American Standard Code of Information Interchange (Amerikanischer Standard-Code für den Informations-Austausch). In der ursprünglichen Form (1963) wurden nur 7-Bit-Muster verwendet, so dass die Tabelle nur 127 Zeichen aufnehmen konnte. Später folgte dann eine Erweiterung auf 8 bit. Damit ließen sich dann auch viele Sondersymbole codieren. Heute verwenden wir – um auch die vielen anderen Buchstaben und Schriftzeichen in einer internationalen Tabelle eine 16-Bit-Kodieren – den **Unicode**.

ASCII-Tabelle

dual	ASCII	dez.	dual	ASCII	dez.	dual	ASCII	dez.	dual	ASCII	dez.
00000000	**NUL**	0	01000000	@	64	10000000		128	11000000		192
00000001	**SOH**	1	01000001	A	65	10000001		129	11000001		193
00000010	**STX**	2	01000010	B	66	10000010		130	11000010		194
00000011	**ETX**	3	01000011	C	67	10000011		131	11000011		195
00000100	**EOT**	4	01000100	D	68	10000100		132	11000100		196
00000101	**ENQ**	5	01000101	E	69	10000101		133	11000101		197
00000110	**ACK**	6	01000110	F	70	10000110		134	11000110		198
00000111	**BEL**	7	01000111	G	71	10000111		135	11000111		199
00001000	**BS**	8	01001000	H	72	10001000		136	11001000		200
00001001	**HAT**	9	01001001	I	73	10001001		137	11001001		201
00001010	**LF**	10	01001010	J	74	10001010		138	11001010		202
00001011	**VT**	11	01001011	K	75	10001011		139	11001011		203
00001100	**FF**	12	01001100	L	76	10001100		140	11001100		204
00001101	**CR**	13	01001101	M	77	10001101		141	11001101		205
00001110	**SO**	14	01001110	N	78	10001110		142	11001110		206
00001111	**SI**	15	01001111	O	79	10001111		143	11001111		207
00010000	**DLE**	16	01010000	P	80	10010000		144	11010000		208
00010001	**DC1**	17	01010001	Q	81	10010001		145	11010001		209
00010010	**DC2**	18	01010010	R	82	10010010		146	11010010		210
00010011	**DC3**	19	01010011	S	83	10010011		147	11010011		211
00010100	**DC4**	20	01010100	T	84	10010100		148	11010100		212
00010101	**NAK**	21	01010101	U	85	10010101		149	11010101		213
00010110	**SYN**	22	01010110	V	86	10010110		150	11010110		214
00010111	**ETB**	23	01010111	W	87	10010111		151	11010111		215
00011000	**CAN**	24	01011000	X	88	10011000		152	11011000		216
00011001	**EM**	25	01011001	Y	89	10011001		153	11011001		217
00011010	**SUB**	26	01011010	Z	90	10011010		154	11011010		218
00011011	**ESC**	27	01011011	[	91	10011011		155	11011011		219
00011100	**FS**	28	01011100	\	92	10011100		156	11011100		220
00011101	**GS**	29	01011101	]	93	10011101		157	11011101		221
00011110	**RS**	30	01011110	^	94	10011110		158	11011110		222
00011111	**US**	31	01011111	←	95	10011111		159	11011111		223
00100000	**SPACE**	32	01100000	-	96	10100000		160	11100000		224
00100001	**!**	33	01100001	a	97	10100001		161	11100001		225
00100010	**"**	34	01100010	b	98	10100010		162	11100010		226
00100011	**#**	35	01100011	c	99	10100011		163	11100011		227
00100100	**$**	36	01100100	d	100	10100100		164	11100100		228
00100101	**%**	37	01100101	e	101	10100101		165	11100101		229
00100110	**&**	38	01100110	f	102	10100110		166	11100110		230
00100111	**'**	39	01100111	g	103	10100111		167	11100111		231
00101000	**(**	40	01101000	h	104	10101000		168	11101000		232
00101001	**)**	41	01101001	i	105	10101001		169	11101001		233
00101010	*****	42	01101010	j	106	10101010		170	11101010		234
00101011	**+**	43	01101011	k	107	10101011		171	11101011		235
00101100	**,**	44	01101100	l	108	10101100		172	11101100		236
00101101	**-**	45	01101101	m	109	10101101		173	11101101		237
00101110	**.**	46	01101110	n	110	10101110		174	11101110		238
00101111	**/**	47	01101111	o	111	10101111		175	11101111		239
00110000	**0**	48	01110000	p	112	10110000		176	11110000		240
00110001	**1**	49	01110001	q	113	10110001		177	11110001		241
00110010	**2**	50	01110010	r	114	10110010		178	11110010		242
00110011	**3**	51	01110011	s	115	10110011		179	11110011		243
00110100	**4**	52	01110100	t	116	10110100		180	11110100		244
00110101	**5**	53	01110101	u	117	10110101		181	11110101		245
00110110	**6**	54	01110110	v	118	10110110		182	11110110		246
00110111	**7**	55	01110111	w	119	10110111		183	11110111		247
00111000	**8**	56	01111000	x	120	10111000		184	11111000		248
00111001	**9**	57	01111001	y	121	10111001		185	11111001		249
00111010	**:**	58	01111010	z	122	10111010		186	11111010		250
00111011	**;**	59	01111011	{	123	10111011		187	11111011		251
00111100	**<**	60	01111100	--	124	10111100		188	11111100		252
00111101	**=**	61	01111101	}	125	10111101		189	11111101		253
00111110	**>**	62	01111110	~	126	10111110		190	11111110		254
00111111	**?**	63	01111111	DEL	127	10111111		191	11111111		255

Daten werden auf einem Computer immer als sogenannte Datei gespeichert. Selbst viele Geräte werden als Datei betrachtet. Das macht ein universelles Arbeiten – auch mit modernsten oder zukünftigen Geräten – möglich.

Definition(en): Datei

D.3. Wo kann ich am Besten speichern? Wo finde ich meine Daten?

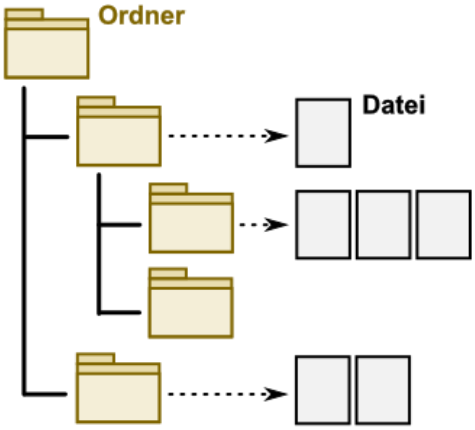
Aufgaben:

1. ***Im Laufe der letzten Schuljahre sind die nachfolgend aufgezählten Zettel mit Kurz-Überschriften zusammengekommen. Nun soll endlich mal Ordnung in die ganze Sammlung gebracht werden. Überlege Dir ein Ordnungs-System, in dem man mit möglichst wenigen Schritten schnell zu den einzelnen Zetteln kommen kann! Auch das Prüfen, ob einer bestimmter Zettel (z.B.*** Biologie_Kl5_Fische***) überhaupt vorhanden ist, so über das System schnell möglich sein! Das System sollte auch ausbaubar / erweiterbar sein!***

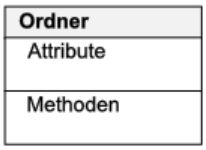
Klasse3_Heimatkunde_Kuh	Kl4_Mathe_1x1	Bio_Kl5_Lebewesen
Deutsch_Gedicht_Klasse4	Heimatstadt_Kl4_Heimatk.	Bio_Kl5_Tiere_am_See
Musik_Meine_Heimat	Zahlen_Mathematik_Klasse5	Plus_Klasse4_Ma
Kl5_Mathe_Zahlen	Klasse4_Heimatkunde_Wetter	Tiere_Bio_Klasse5
Kunst_Kl4_Laubblätter	Oh_Tannebaum_Musik	Kl5_D_Ostergedicht
Ma_Klasse4_Rechentürme	Großes1x1_Ma_Klasse4	Mu_Osterlieder

2. ***Tausche Deinen Vorschlag für eine Ordnungs-System mit Deinem Nachbarn aus! Diskutiert darüber, welches System besser ist!***

D.3.0. Laufwerke, Ordner und Dateien



Das Prinzip der Ordner (Verzeichnisse) und Dateien ist in allen Betriebssystem gleich. In Windows (wie auch im alten DOS) wurde zusätzlich noch Laufwerke definiert. Diese entsprechen üblicherweise den verschiedenen technischen Geräten, wie z.B. der Festplatte, dem DVD-Laifwerk, einem USB-Stick oder einem Netzwerk-Laufwerk.
Bei Linux sind die technischen Geräte spezielle Unterordner in einem Ordner-System mit nur einem einzigen Wurzel-Verzeichnis.
Da in Computer-Systemen meist sehr viele Ordner vorkommen, überlegt man sich ein einheitliches Aufbau-Schema. Das Aufbau-Schema wird **Klasse** genannt. Alle Ordner gehören zu einer Klasse von informatischen Dingen.
Zum graphischen Darstellen von Klassen werden üblicherweise Rechtecke benutzt, in denen oben der Klassenname verzeichnet ist.
Zum Aufbau-Schema – auch **Klassen-Definition** genannt – gehören zuerst einmal die Eigenschaften eines Ordners, wie der Name oder ein Schreibschutz-Kennzeichen. Solche Eigenschaften werden in der Informatik **Attribute** genannt.



Klasse:	**Bemerkungen / Hinweise**
Ordner	auch "Verzeichnis" genannt
→ Attribut(e):	
• Wurzel-Verzeichnis	
• Name	
• Attribute	
○ Schreibschutz	
○ System	
○ Versteckt	
• Unterordner	

Neben den Atrributen gibt es auch noch Tätigkeiten, die mit einem Ordner gemacht werden können. Dazu gehören z.B. das Neuerstellen oder das Kopieren. Die Tätigkeiten werden **Methoden** genannt. Eine Klassen-Definition beinhaltet also immer Attribute und Methoden.

Klasse:	**Bemerkungen / Hinweise**
Ordner	auch "Verzeichnis" genannt
→ Attribut(e):	

(c,p) 2015 – 2020 lsp: dre

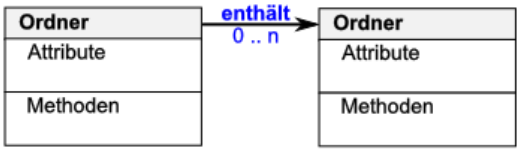
• Wurzel-Verzeichnis	
• Name	
• Attribute	
○ Schreibschutz	
○ System	
○ Versteckt	
○ Archiv	
• Unterordner	
→ Methode(n):	
• Neu erstellen	erzeugt einen neuen Ordner
• Umbenennen	der aktuelle Ordner wird neu benannt
• Löschen	der aktuelle Ordner wird (mit Inhalt) gelöscht
• Ausschneiden	zum Arbeiten mit der Zwischenablage
• Kopieren	zum Arbeiten mit der Zwischenablage
• Einfügen	zum Arbeiten mit der Zwischenablage
•	

Aus Effektivitäts-Gründen werden bestimmte Attribute und Methoden nicht bei jeder Klasse neu definiert und realisiert (programmiert). Besonders die Methoden hat man meist schon vorher einmal in ähnlicher Form bereitgestellt. Deshalb erben Klassen solche Methoden und / oder Attribute von anderen Klassen. Diese werden als Unterklassen bezeichnet.
Viele der Attribute werden auch schon mit bestimmten Werten vorbelegt, die als Standard verwendet werden. Der Nutzer oder das Betriebssystem kann diese Vorbelegungen dann überschreiben.

Klasse:			**Bemerkungen / Hinweise**
Ordner			auch "Verzeichnis" genannt
→ Unterklasse(n):			
• Datei			
•			
→ Attribut(e):		***Vorbelegung***	
• Wurzel-Verzeichnis	=	*falsch*	
• Name	=	*Neuer Ordner*	
• Erstell-Datum	=	*01.01.2018 10:22:09*	Datum wird Sekunden-genau gespeichert
• Änderungs-Datum	=	*01.01.2018 10:23:21*	
• letzter Zugriff	=	*12.03.2018 17:32:42*	
• Attribute			
○ Schreibschutz	=	*falsch*	
○ System	=	*falsch*	
○ Versteckt	=	*falsch*	
○ Archiv	=	*wahr*	
• Unterordner	=	*[ ]*	Liste der Unterordner ist leer
→ Methode(n):			
• Neu erstellen			erzeugt einen neuen Ordner
• Umbenennen			der aktuelle Ordner wird neu benannt
• Löschen			der aktuelle Ordner wird (mit Inhalt) gelöscht
• Ausschneiden			zum Arbeiten mit der Zwischenablage
• Kopieren			zum Arbeiten mit der Zwischenablage
• Einfügen			zum Arbeiten mit der Zwischenablage
•			

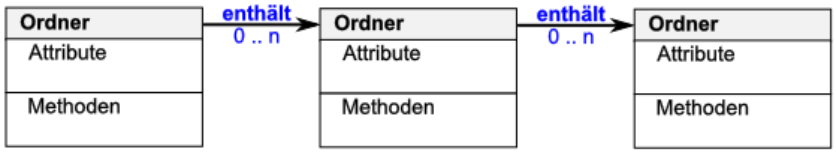
In modernen Betriebssystemen werden noch weitere Attribute verwaltet. Dazu ghören z.B. der "Besitzer" oder auch "Zugriffs"-Rechte.
Innerhalb eines Ordners können sich nun

wiederum weitere Ordner befinden. Es muss keiner sein, aber es können auch sehr viele sein. In Klassen-Diagrammen steht man die Beziehung zwischen den Klassen mittels eines Pfeils dar.

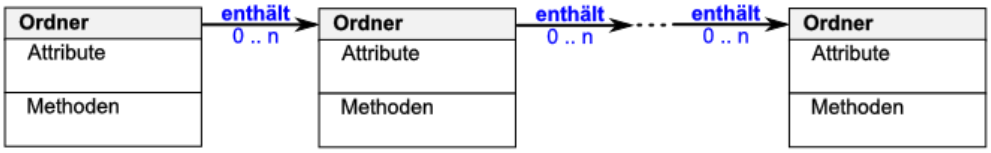


An den Pfeil notiert man die Art de Beziehung – hier ist das "enthält" – und die sogenannte Kardinalität. Die Kardinalität beschreibt die Zahlen-Verhältnisse zwischen beiden Klassen. In unserem Beispiel ist die Kardinalität von "enthält" ein Bereich von 0 – also keinem – bis n – also unendlichen vielen – Ordnern. Die enthaltenen Ordner werden häufig auch als Unter-Ordner bezeichnet. Dieser Name bezieht sich aber nur auf die für uns sichtbare Hierrarchie – nicht auf eine andere Art von Ordner.

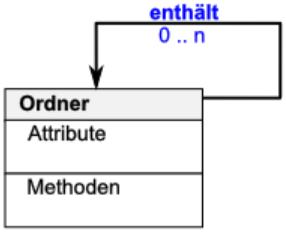
Die "enthält"-Beziehung existiert jetzt auch für den nächsten Unter-Ordner. Er kann wieder 0 bis n Unter-Ordner enthalten.



Die Unterordnung lässt sich theoretisch beliebig fortsetzen. In vielen Betriebssystemen gibt es aber Begrenzungen.



Da diese Art der Klassen-Diagramme iregndwie unklar ist – schließlich weiss ja keiner im Voraus, wieviele Unterordner es geben kann – benutzt man eine effektivere Darstellung.

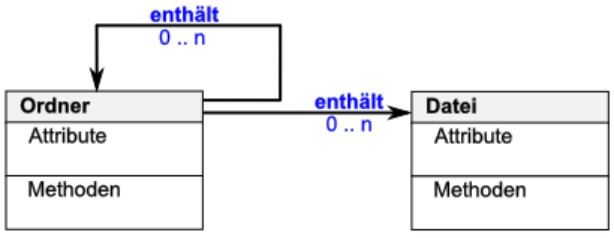


Wir haben ja schon gesagt, dass alle Ordner und Unterordner zur gleichen Klasse gehören. Praktische verweist also die Klasse auf sich selbst. Das zeichnen wir in Klassen-Diagrammen durch einen selbst-bezüglichen Pfeil.

Dieses Klassen-Diagramm ist nun wesentlich kürzer und enthält trotzdem immer noch alle notwendigen Informationen.

Neben den Ordnern können nun auch noch Dateien in den Ordner enthalten sein. Dateien sind eine Klasse für sich. Sie haben ihre eigenen Attribute und Methoden.

Schaut man sich diese allerdings genauer an, dann wird man viele Gemeinsamkeiten entdecken.



Klasse:

Datei		
→ Unterklasse(n):		
•		
→ Attribut(e):		***Vorbelegung***
• Name	=	
• Erstell-Datum	=	*12.03.2018 13:32:42*
• Änderungs-Datum	=	*12.03.2018 13:33:15*
• letzter Zugriff	=	*12.03.2018 13:35:00*
• Attribute		
○ Schreibschutz	=	*falsch*

Bemerkungen / Hinweise

auch "File" genannt

	○ System	=	*falsch*
	○ Versteckt	=	*falsch*
	○ Archiv	=	*wahr*
→ Methode(n):			
• Neu erstellen			erzeugt eine neue Seite mit dem akt. Format
• Umbenennen			
• Löschen			
• Ausschneiden			
• Kopieren			
• Einfügen			
•			

Beim Erstellen eines Ordners – mit der Methode "Neu erstellen" – wird nun ein echter Ordner auf dem Datenträger erzeugt. Man spricht jetzt von einem Objekt. Dieser könnte z.B. "Hausaufgaben" heißen. Das Objekt (der Ordner) "Hausaufgaben" ist eine konkrete Ableitung aus der Klasse "Ordner". Objekte sind immer reale, individuelle Dinge. Klassen sind die Typen oder Gruppen zu denen die Objekte zugeordnet werden.

Objekt:			**Bemerkungen / Hinweise**
Ordner			auch "Verzeichnis" genannt
→ Unterobjekt(e):			
• Ordner			meint Unter-Ordner
• Datei			
•			
→ Attribut(e):		***Wert***	
• Seitenformat			
○ Seitengröße	=	*A4*	
○ Orientierung	=	*hoch*	
○ Randabstände			
▪ links	=	*2 cm*	
▪ rechts	=	*3 cm*	
▪ oben	=	*2 cm*	
▪ unten	=	*2 cm*	
• Wurzel-Verzeichnis	=	*falsch*	
•	=		
→ Methode(n):			
• Neu erstellen			erzeugt eine neue Seite mit dem akt. Format
• Seitenformat festlegen / ändern			
• Ansicht aktualisieren			
• Druckvorschau erzeugen			
•			

Objekt:			**Bemerkungen / Hinweise**
Datei			Bitmap-Typ
→ Unterobjekt(e):			
• Pixel			Voll-Liste
•			
→ Attribut(e):		***Wert***	
• Dateiname	=	*Haus*	dieses Attribut kann der Nutzer direkt ändern
• Dateityp	=	*BMP*	
• geändert	=	*wahr*	dieses Attribut wird vom Programm verwaltet
• Breite	=	*30 px*	

• Höhe	= *20 px*
•	=
→ Methode(n):	
• Neu erstellen	
• Speichern unter …	
• Speichern	
• Öffnen	
• Schließen	
•	

Exkurs: Was sind Objekte und Klassen?

Lösen wir uns mal ganz vom Computer und mit ihm zusammenhängenden Dingen.
Objekte sind – einfach gesagt – Dinge aus der realen Welt. Dazu gehören Euer Familien-Auto, Dein Fahrrad, Dein Frühstücks-Teller, Dein Wohnhaus, Deine Eltern usw. usf.
Euer Familien Auto ist wahrscheinlich ein typisches Auto. Es gehört wie viele andere zur Klasse der Autos. Der Begriff Auto meint meist nicht ein konkrtes Auto (- als Objekt -), sondern eher irgendein Auto. Es handelt sich mehr um die allgemeine Beschreibung eines Fahrzeugtypes. Solche allgemeinen Beschreibungen nennen wir in der Informatik **Klassen**.
Dein Fahrrad gehört also zur Klasse der Fahrräder. Autos und Fahrräder gehören zur gemeinsamen Klasse der Fahrzeuge. Im Allgemeinen können wir Klassen also irgendwelchen übergeordneten Klassen zuordnen. Diese nennen wir auch **Oberklassen**.
Häufig können wir auch noch Unterklassen finden. So lassen sich die Fahrräder z.B. in die Unterklassen Kinderräder, Sporträder, Freizeiträder, E-Bikes und Padilac's unterteilen.
Aber wie können wir die Dinge nun sicher zu Klassen zuordnen? Dazu brauchen wir die Eigenschaften oder Merkmale der Objekte. Dazu gehören z.B. die Größe, die Farbe, die Reifen-Größe, Sattelart usw. usf. Diese Merkmale bezeichnet man in der Informatik als **Attribute**. Jedes Objekt hat eine einmalige Kombination aus Attributen. So erkennen wir z.B. unser eigens Fahrrad und können es einer speziellen Klasse zuordnen.

Aufgaben:

1. *Betrachte Deine Schultasche! Überlege Dir, welche Eigenschaften und Merkmale sie hat!*
2.
3. *Was haben die Klassen in einer Schule eigentlich mit den Klassen in der Informatik zu tun? Oder gibt es da gar keine Beziehung? Stelle Deine Meinung vor und argumentiere ev. gegen andere Meinungen aus Deinem Kurs!*

D.3.1. lokale Datenträger

Unter den lokalen Datenträgern versteht man üblicherweise die Speicher-Geräte eines Computers. Dazu gehören z.B. Festplatten, DVD-Laufwerke, USB-Stick's – aber auch die veralteten Disketten-Laufwerke.
In Betriebssystemen der DOS/Windows-Welt sind den Geräten Laufwerks-Buchstaben zugeordnet. Einige der Zuordnung sind historisch mehr oder weniger festgelegt. Andere können sich ändern.
Zu den klassischen Laufwerken gehört z.B. die Festplatte, die meist den Buchstaben C erhält. Als Laufwerks-Kennzeichen wird ein Doppelpunkt verwendet. Häufig gibt es ev. noch eine zweite Festplatte mit der Kennung D:. Jeweils der nächste Buchstabe ist dann für das DVD-Laufwerk das E: oder eben D:.
In den meisten Fällen sind aber gar keine zwei Festplatten eingebaut. Hier wurde durch einen Trick (- der nennt sich Partitionierung) die Festpaltte in zwei (oder mehr) Bereiche aufgeteilt. Diese nennen sich Partitionen und erhalten Laufwerks-Buchstaben, wie echte Laufwerke. Der einfache Nutzer kann den Trick oft gar nicht erkennen.
Die Unterteilung wird meist vorgenommen, um die Betriebssysteme und Programme von den viel wichtigeren Daten zu trennen.
Die Laufwerke A: und B: waren früher für die Disketten-Laufwerke reserviert. Heute kann man sie – mangels solcher Geräte – auch z.B. für USB-Stick's verwenden. Ansonsten werden USB-Stick's meist die nächsten freien Buchstaben hinter D: bzw. E: zugeordnet.

Vorteile
volle Kontrolle über die Daten

Nachteile
Daten nur auf dem konkreten Rechner verfügbar

üblicherweise bekommen die Datenträger kurze Bezeichnungen.
weil früher die Datenträger imme was mit bewegten Teilen zu tuen hatten nannte man sie Laufwerke
solche alten / klassischen Laufwerke waren Disketten-Laufwerke, Streamer (Band-Laufwerke) oder
aus historischen Gründen ist man beim Laufwerks-Begriff geblieben

Laufwerken werden in vielen Computer-Systemen Buchstaben zur schnelleren Erkennung zugeordnet. Damit man jederzeit erkennen kann, das es sich um einen Laufwerks-Buchstaben handelt, folgt dahinter ein Doppelpunkt.

Klassisch ist das Laufwerk C: das Speichergerät, auf dem das Arbeits-System (Betriebs-System) zu finden ist. Aber auch viele Programme und nicht selten auch die Daten finden hier Platz.

Um bei den vielen Daten und Programmen auf einem Computer den Überblick zu behalten wurde ein Ordnungs-System entwickelt. Die Dateien werden – je nach Verwendungs-Zweck – in bestimmte Ordner (Verzeichnisse) gesteckt. Die Ordner auf dem Computer kann man sich wie Schnellhefter (Mappen) vorstellen, in denen die Daten als Zettel gespeichert werden.
Die Daten sind gedruckte oder geschriebene Zeichen auf dem Papier.
Papier ist Datenträger
Sortierungen z.B. durch Akten-Dulli's oder Register-Blätter entsprechen den Verzeichnissen
Laufwerke sind mit Aktenordnern gleichzusetzen.

Bei größeren Datenmengen können Regale als Laufwerke betrachtet werden und dann sind die Regal-Böden und auch die Ordner Verzeichnissen zuzuordnen.

Vorteile elektronischer Ordner:

- lassen sich praktisch ohne Speicherbedarf realisieren
- beliebig viele Ebenen / Unterteilungen / Hierrarchie-Ebenen
- lassen sich einfach umsortieren / umorganisieren

Datenträger aus der Geschichte der Informatik

Disketten / Floppy-Disk

noch heute das Symbol für "Speichern"
der! Datenträger der ersten Personal-Computer (PC) und Heim-Computer (HC)
geringe Speicher-Kapazitäten,
bei den ersten Systemen reichte die Speicherkapazität noch für Betriebssystem (Konsolen-System), Programme und Daten
später mit graphischen Betriebssystemen (wie z.B. microsoft® Windows®) wurden dann Festplatten gebraucht (zuerst nur für Betriebssystem und Programme, später auch für die Daten)

mobil, praktisch jeder Computer hatte früher so ein Laufwerk;
Daten relativ universell austauschbar

Exkurs: Speicher-Kapazität

Wir haben schon die Speicher-Größen Bit und Byte kennen gelernt. Irgendwann werden die Zahlen für die Speicher-Einheiten sehr groß. Wie in der physikalischen Welt arbeitet man mit Vorsätzen, um Zahlen sinnvoll zu verkürzen.

Eintausend Gramm kann man noch als Zahl mehrfach für Rechnungen usw. aufschreiben. Schöner ist aber die Zusammenfassung zu einem Kilogramm. Dabei bedeutet das Kürzel k in kg genau 1'000 bzw. 10^3.
Niemand mag aber sehr große Werte – wie z.B. eine Milliarde – als Zahlenwert aufschreiben.

$$1'000\ g = 1\ kg$$

$$1'000\ g = 1 \cdot 1'000\ g$$
$$1'000\ g = 1 \cdot 10^3\ g \qquad 10^3 \equiv k$$

Dafür gibt es weitere Äquivalente in Form von Vorsätzen (Präfixe) für die Einheiten. Dazu gehört z.B. M für 106 (= 1'000'000) oder G für 109 (= 1'000'000'000). M wird als Mega und G als Giga gesprochen.
Bei den physikalischen Größen sind auch Verkleinerungen, wie z.B. m, c, d und µ gebräuchlich. Die brauchen wir in der Informatik nicht, da es kleine Speicher-Größen als Eins nicht geben kann.
Weiter vorne haben wir schon festgestellt, dass mit 10 Speicher- / Verarbeitungs-Einheiten 1'024 Zustände speicherbar sind. Eigentlich bräuchten wir für ein kbit nur 1'000. Da würden 24 übrig bleiben, die der nächsten Menge zugeordnet werden müssten. Das ist für die Praxis viel zu kompliziert. Deshalb hat man sich für die Informatik auf eine leicht geändertes System

geeinigt. In der Informatik verwendet man den Faktor 1'024 statt 1'000 (Dezimal-Präfix). Somit sind 1'024 bit ein Kilobit. Zur Unterscheidung der beiden Faktioren benutzt man in der Datenverarbeitung das große K für den Faktor 1'024 (Binär-Präfix). Für exakte Berechnungen muss man den kleinen Unterschied unbedingt beachten. Auch bei den höheren Faktoren blieb man bei 1'024.

Somit ergeben sich die nebenstehenden Beziehungen.

Bei den größeren Vorsätzen gibt es keine Korrektur der Buchstabenschreibung.

Das G steht für Giga, T für Tera und P für Peta.

Der anfänglich kleine Unterschied von 24 bit multipliziert sich bei größeren Faktoren zu einem merklichen Fehler.

1'024 bit = 1 Kbit
1'024 Kbit = 1 Mbit
1'024 Mbit = 1 Gbit
1'024 Gbit = 1 Tbit
1'024 Tbit = 1 Pbit

Um auch bei höheren Werten und größeren Vorsätzen exakt bleiben und die Präfixe eindeutig zu können, hat man sich auf neue Binär-Präfixe festgelegt. Diese bekommen neue Namen und Symbole.

Für das Kilo sieht das z.B. so aus: Aus Kilo (K) wird jetzt Kibi (Ki). Somit heißen 1'024 bit jetzt eindeutig und Zahlensicher 1 KibiBit.

Entsprechend wurde mit den anderen Präfixen verfahren. Sie heißen nun Mebi (Mi), Gibi (Gi), Tebi (Ti) und Pebi (Pi).

Zum besseren Ausprechen wurde also auch noch einige Vokale angepasst.

1'024 bit = 1 Kibit
1'024 Kbit = 1 Mibit
1'024 Mbit = 1 Gibit
1'024 Gbit = 1 Tibit
1'024 Tbit = 1 Pibit

Es ist heute schon abzusehen, dass für die Datenmengen der Zukunft, die obigen Einheiten-Vorsätze nicht mehr reichen werden.

Nun kommen neue / weitere Vorsätze hinzu.

Diese sind für unseren Sprachgebrauch sehr ungewöhnlich. Die Vorsätze sind im SI-Einheitensystem definiert, werden aber für naturwissenschaftliche Zwecke nur sehr selten gebraucht.

1'024 Pbit = 1 Exbit
1'024 Exbit = 1 Zebit
1'024 Zebit = 1 Yobit
1'024 Yobit = 1 Brbit
1'024 Brbit = 1 Gebit

Peta = 10^{15}
Exa = 10^{18}
Zetta = 10^{21}
Yotta = 10^{24}

Einige Vorsätze wurden als Ergänzungen mit in das SI-System aufgenommen:

Bronto = 10^{27}
Geob = 10^{30}

Aufgaben:

1. *Berechne in einer Tabelle exakt, wieviele Speicherzellen für 1 Kb, 1 Mb und 1 Gb gebraucht werden! (Lasse in der Tabelle noch 3 Spalten frei!)*
2. *In der ersten leeren Spalte berechne nun, wieviele kb, Mb und Gb herauskommen würden, wenn man den Umrechnungs-Faktor 1'000 benutzen würde!*

3. *Die Hersteller von Festplatten usw. benutzen lieber den Faktor 1'000 bei der Größen-Angabe für ihre Produkte. Warum tun sie dies?*
4. *Berechne in der nächsten freien Tabelle den prozentualen Fehler zwischen der exakten Berechnung und der vereinfachten Rechnung mit dem Faktor 1'000!*
5. *Ermittle nun für die letzte freie Spalte die exakten Binär-Vorsätzen!*
6. *Übernimm die folgende Tabelle und ermittle die fehlenden Inhalte! Kennzeichne die vorgegebenen Zellen durch leichtes Ausmalen mit einem Bleistift!*
   (Wenn Zahlen erst hinter der 4. Dezimalstelle einen Ziffernwert haben, dann können sie getrichen werden! Große Zahlen über 1 Mrd. (= 1'000'000'000) dürfen als 10er Potenz (hier 10^9) notiert werden!)

	Binär-Präfixe			Dezimal-Präfixe		
bit	Kibit	Mibit	Gibit	kbit	Mbit	Gbit

für Experten und zum Knobeln:

7. *Unser Gehirn hat eine geschätzte Speicherkapazität von 2,5 Petabytes. Notiere die Speichergröße in Byte!*
8. *Wieviele Festplatten mit einer Kapazität von 5 Terabyte braucht man, um alle "Daten" aus einem Gehirn zu speichern? Rechne vor!*
9. *Wie groß ist der Gehirnspeicher, wenn ein passender Binär-Präfix verwendet werden soll!*

D.3.1.2. moderne Datenträger

Festplatten

ältere Festplatten enthalten Magnet-Platten, auf denen die Daten gespeichert sind
Speicherung der Information durch unterschiedliche Richtung des Magnetfeldes an einer Speicher-Stelle
mechanische Arbeitsweise bewirkt relativ hohe Empfindlichkeit gegenüber Stoß und Schlag; Gefahr der inneren Zerstörung durch Fallenlassen
relativ langsam; Grenzen der Speicherkapazitäten (bei magnetischen System) fast erreicht

moderne Festplatten – die heißen dann SSD-Platten – speichern die Daten in Micro-Chips
kleine elektronische Bau-Elemente, die Informationen auch nach dem Abschalten des Stroms speichern können
auch hier Minituarisierung der Chip-Strukturen fast erreicht, aber in der Fläche ausdehnbar

Problem hier ist die Fehler-Freiheit und Arbeits-Sicherheit der Chip-Strukturen (Speicherzellen)

CD / DVD / Blue-ray

Datenspeicherung erfolgt als kleine Löcher im Daten-Träger
werden optisch gelesen (und geschrieben)
ein Laser-Strahl (stark gebündeltes Licht) tastet Oberfläche ab
die normale Reflexion wird von der Nicht- oder Anders-Reflexion bei den Löchern unterschieden
Zuordnung beider Zustände zu 0 und 1

USB-Stick's

die gleichen Chip's sind auch in den USB-Sticks's verbaut. Sie sind mit einem speziellen Stecker ausgestattet, für den auf praktischen allen modernen Computern eine passende Buchse vorhanden ist.
Somit kann man seine Daten – wenn sie denn auf dem Stick gespeichert sind – an einen anderen Computer mitnehmen.

Speicher-Karten

derzeit vorherrschend SD- oder microSD-Karten

ehemals als Speicher für digitale Foto-Apperate entwickelt, finden wir die Speicherkarten heute in vielen MP3-Playern, Smartphone's und Tablet's. Da diese Geräte allesamt sehr klein sind, haben dort so große Buchsen – wie das USB-System – keinen Platz mehr. Deshalb nutzt man lieber die sehr kleinen und flachen microSD-Karten.

die Welt der Kunden-Karten

Karten mit echter Speicher-Funktion

beeinhalten außer den einfachen Kunden-Daten auch spezielle Speicher
diese werden z.B. zum Speichern weiterer (oft verschlüsselter) Daten benutzt
spezielle Anwendung

- Aufladen von Geld- oder Guthaben-Beträgen

Krankenkassen-Karte

Sim-Karte für Handy / Smartphon / …

Kundenkarten von Schwimmbädern / Fitness-Centern / …

Karten ohne Speicher-Funktion / Identifikations-Karten

beeinhalten nur wenige Kunden-Daten
dient nur der Identifizierung des Kunden
Daten liegen auf großen Servern
nur online Verarbeitung der Daten möglich
Kunden-Bindung und individualisierte Werbung

payback

Deutschland-Card

D.3.2. Netzwerk-Laufwerke

D.3.3. die Cloud (Speichern in der Wolke)

Nachteil der lokalen Speicher-Laufwerke ist, dass man sie nur begrenzt mit irgendwo hin nehmen kann. Ärgerlich ist es auch, wenn man seinen USB-Stick gerade zuhause vergessen hat.
Schön wäre es, wenn man z.B. über das Internet auf seine Daten zugreifen könnte. Genau das ermöglicht uns die Cloud. Früher wurde in Abbildungen oft das Internet als Wolke abgebildet, weil man nicht genau weiss, wo welcher Computer genau steht. So ist es auch mit den modernen Datenspeichern – die nun Cloud genannt werden. Die Computer, auf denen die Daten auf speziellen Laufwerken gespeichert werden, stehen irgendwo im Internet. Mit speziellen Programmen greift man auf seine "Internet"-Daten zu und kann mit ihnen genauso arbeiten, wie mit Daten auf den lokalen Laufwerken.

Problematisch ist aber, das die Daten im Internet auch für andere nutzbar sein könnten. Hacken die unsere Cloud oder bekommen Fremde unsere Anmelde-Daten für die Cloud in die Hände, dann können sie auch unsere Daten sehen.
Das Hacken eines lokalen Speichers ist deutlich schwerer.

D.3.4. Sicheres Speichern von Daten

Öffnen und Speichern einer Datei unter Beachtung des Speicherortes und eines zweckmäßigen Dateinamens

Auf den privaten Computern sollte der Speicherort zuerst einmal immer der Ordner "Eigene Dateien" ("Eigene Dokumente" od. so ähnlich) bzw. "home" (auf Linux-Rechnern) sein.
In Computer-Netzen wird jedem Nutzer ein spezielles Laufwerk zugewiesen, was meist auch "Home" heißt und vielleicht auch den Laufwerks-Buchstaben **H:** bekommen hat.

Aufgaben:

1. *Erkunde, wo auf Deinem Computer die Daten gespeichert werden sollen! Frage dazu bei Deinen Eltern und / oder dem Lehrer nach!*
2. *Notiere Dir Deine Speicherorte gut sichtbar auf Deinem Hefter od.ä.! Unterscheide ev. zwischen Zuhause und Schule!*

Manche Schulen benutzen ein System, bei dem die für die Schule notwendigen Programme und auch die Daten auf einem USB-Stick gespeichert sind. Mit diesem Stick kann man dann in der Schule und Zuhause gleichermaßen arbeiten. Das System nennt sich "portableApps" und es gibt verschiede Varianten davon. Ein spezielles für die Schule heißt IoStick ("Informatik on Stick").
Die portableApps-Systeme haben einen Dokumente-Ordner auf dem Stick, in dem dann auch gespeichert wird.

Beim Speichern gibt es drei Dinge festzulegen, den Speicher-Ort, den Datei-Typ und den Datei-Namen.
Die ersten beiden Angaben sind oft automatisch vorgegeben und brauchen nur kontrolliert werden. Natürlich können wir sie bei Bedarf auch ändern.
Was wir aber immer genauestens festlegen müssen ist der Datei-Name. In jedem Ordner kann immer nur eine Datei mit einem Namen vorkommen. Wir müssen also immer einen anderen Namen vergeben, wenn mehrere Dateien im gleichen Ordner gespeichert werden sollen.
Es ist notwendig, jeder Datei einen passenden und informativen Namen zu geben. Vielleicht wissen wir noch nach einer Woche, was isn der Datei "123" gespeichert war, aber nach einem Jahr ??? Benutzt man dagegen Namen wie "Bewerbung Firma Müller", "Berechnung Kontostand 2018" oder "Vorstellung Projektarbeit", dann wird schnell klar, was in den Dateien nethalten sein wird.
Auf Angaben vom Speicher-Datum und -Uhrzeit kann man auch in einem Dateien-Namen verzichten, weil diese automatisch gespeichert werden. Das macht der Computer.
Im Allgemeinen haben alle modernen Programme zum Speichern ganz ähnlich aufgebaute Dialoge. Sie führen uns Nutzer durch das Speichern.
Diese Dialoge können – je nach Computer-System etwas unterschiedlich aussehen – der Grobaufbau ist aber fast immer gleich.

Für Deine Zwecke musst Du Dir nun die richtige Variante aussuchen. Unter Umständen unterscheiden sich das Speichern zuhause und in der Schule, die Lehrer und Deine Eltern werden Dir aber sicher helfen, die richtige Variante auszuwählen:

Speichern auf einem klassischen Schul-Computer (Windows):

Speichern auf einem heimischen Windows-Computer:

Speichern auf einem portableApps-System:

Speichern auf einem Linux-Rechner (z.B. Raspberry Pi):

Speichern auf einem Android-Tablett:

Speichern auf einem iOS-Gerät (MacBook, iPhone,

Wenn wir nun Daten von Zuhause mit in die Schule bringen wollen / müssen, dann bietet sich vorrangig ein USB-Stick als Speicher-Ort an:

Speichern auf einem USB-Stick (Windows-System):

D.3.y. Schutz vor Daten-Verlusten

3-2-1-Strategie

3 identische Dateien auf **2** verschiedenen lokalen Datenträgern (Festplatte + z.B. USB-Stick / NAS) und **1** externen Cloud (od.ä.)

minimalistisch wären 2 identische Dateien auf 2 Datenträgern, wovon einer eine Cloud od.ä. ist

D.4. Arbeit mit der Zwischenablage

Verwenden der Techniken zum Markieren, Kopieren, Einfügen und Ausschneiden von Objekten

Marktplatz für Daten(-Stücke)

D.5. Wem gehören die Daten?

Problem-Fragen für Selbstorganisiertes Lernen
Welche Daten werden geschützt? Woher weiss mein Browser immer, wofür ich mich interessiere / interessiert habe?

Arbeits-Daten
gehören ersteinmal immer dem Erzeuger

Die Daten anderer als seine eigenen auszugeben, ist Betrug!

Die Kopierbarkeit von Daten ist kein Zeichen für eine freie Nutzung durch Jedermann.

Urheberrecht
Grundzüge

für die Beachtung vor dem Gesetz ist eine besondere Schöpfungshöhe notwendig
nicht jede Krakel-Zeichnung ist urheberrechtlich zu schützen
eine Krakel-Zeichnung eines Maler kann es aber sein
Maß von allgemeinen Regelungen umrissen, im Spezialfall muss es zur Klärung vor Gericht kommen

Recht am eigenen Bild

Vorsicht bei freizügigen, intimen – sehr privaten – Foto's. Sexting ist das einvernehmliche Tauschen von intimen / erotischen Texten, Foto's oder Video's. Zumeist werden sicher geglaubte Nachrichten-Kanäle benutzt. Also eher WhatsApp, Snatshat uns Skype.
Es besteht immer die gefahr, dass Foto's in falsche Hände kommen. Z.B. durch versehentliches Verschicken an eine falsche Adresse. Ein enttäuschter / zurückgewiesener / nicht mehr aktueller (Liebes-)Partner nutz die Foto's als Rache-Mittel. Ein Account wird geknackt oder das Handy gelangt in falsche Hände (Diebstahl, Verlust).

Sonderfall: Personen-Daten

Datenspuren

Wenn wir uns durch die Welt bewegen hinterlassen wi Spuren. Am Besten kann man das z.B. am Strand im feuchtem Sand oder im Schnee erkennen. Aber auch von den Geruchsspuren, die durch Polizei-Hunde verfolgt werden können habt ihr schon gehört. Allen diesen Spuren ist gemeinsam, dass sie irgendwann von ganz alleine verschwinden. Irgendwann ist der Schnee geschmolzen oder die Spuren am Strand wurden von einer Welle weggespült.
Die Zuordnung unserer Person zu den Spuren ist nur teilweise sicher möglich. Auch andere Personen können ähnliche Schuhe mit ähnlichen Sohlen gehabt haben.
Obige Datenspuren von uns kann man auch nicht so einfach woanders hin transportieren oder kopieren. Von Spuren im Sand müss-

ten dazu vielleich schwere Gips-Abdrücke gemacht werden.
Auch – und ganz besonders – in der digitalen Welt hinterlassen wir Spuren. Die andere Welt nennt man übrigens die analoge Welt.
Beim Arbeiten mit Computern und Kommunikations-Geräten ist die Anzahl der Spuren die wir hinterlassen viel größer. Spätestens, wenn wir uns bei einem System anmelden, wird unsere Präsenz gespeichert. Oft passiert das schon automatisch. Dein Smartphone meldet sich – und damit praktisch ja auch Dich – am nächstgelegenen Sende-Mast Deines Mobilfunknetzes an. Sonst wärest Du nicht erreichbar und es könnten auch keine Daten übertragen werden.
Bei den heutigen Preisen für große Datenspeicher (Festplatten, Cloud-Speicher) ist das Aufheben von Daten auf unbestimmte Zeit überhaupt kein Problem mehr. Schließlich weiss man ja auch nicht, wann man diese Daten vielleicht mal (wieder) braucht.
Jede Datei wird mit einer Vielzahl von zusätzlichen Informationen gespeichert. Schon in einem Date-Manager (z.B. Arbeitsplatz od. Windows-Explorer) sind viele Nutzungs-Informationen abrufbar.

Digitale Daten können praktisch beliebig oft – und praktisch auch ohne großen Aufwand – kopiert werden.

Bei der Installation von Apps gestatten wir oft den Zugriff auf irgendwelche Daten. Aber welche Daten genau werden nun von wem genutzt?

Cookies

Aufgaben:

1. *Überlege für einen normalen Unterrichtstag, welche Spuren Du in der "analogen" Welt hinterlässt!*
2. *Betrachte nun einen Tag, an dem Du Informatik-Unterricht hast! Überlege Dir, welche Datenspuren Du in der digitalen Welt hinterlassen hast!*
3. *Vergleiche Datenspuren der analogen Welt mit denen in der digitalen Welt!*

"Daten sammeln" als Geschäfts-Modell

Viele glauben die großen Internet-Firmen oder Rabatt-Systeme (DeutschlandCard, Payback) sind Leistung, die nur zu Gunsten der Nutzer erbracht werden. Jede Firma will Geld verdienen, auch Facebook, youtube oder google. Ihr Geschäft basiert nicht nur auf der Vermarktung von Werbung. Ein ganz wichtiger Ertragsteil wird aus dem gewonnen, über das diese Firmen lieber nicht so laut reden. Sie vermarkten die Daten ihrer Nutzer. Aus dem reinen Sammeln kann man kaum Gewinn erzielen, aber aus dem Verkauf von Daten schon. Zum Einen eben für Werbung, die auch schon gezielt ablaufen kann. Wer das letzte Mal vielleicht ein Fahrrad bei einem großen Händler gekauft hat, bekommt nun auf einmal massiv Werbung für den Kauf von Fahrrädern und ev. auch von Zubehör. Kauft man einen anderen Artikel, dann geht es mit der Werbung in diese Richtung weiter. Ganz offensichtlich hat hier irgendwer Daten abgegriffen und sie weitergegeben. Unentgeldlich machen das auch die großen Firmen nicht untereinander – schließlich will jeder mitverdienen. Das Geschäfts-Modell müsste also besser "Daten sammeln und verkaufen" heißen.
Und der Begriff der Werbung wird von einigen Firmen sehr frei interpretiert. So kann es schon sein, dass man gezielt Nachrichten erhält, die eher politischer Natur sind. Auch Fake-News gehören in diesen Bereich. Die Beeinflussung des Wahl-Verhaltens der Bürger nimmt immer größere Maße an.

Aufgaben:

1. *Erstelle eine Übersicht, welche Rabatt- und Club-Karten Ihr in der eigenen Familie benutzt!*
2. *Übernimm die folgende Tabelle und fülle sie durch Interview's mit den Karten-Besitzern aus!*

Karten-Name	erwartete Leistung / Vorteile / … für den Karten-Inhaber	Vorteile / Nutzen für die Betreiberfirma (Vermutungen / bekannte Nutzung)

Vermeiden von Datenspuren

Bei der amerikanischen Suchmaschine yahoo sind über 50% der Suchanfragen "google".

Nicht alle Datenspuren lassen sich verhindern. Beim Smartphone muss ein Ein- und Auslog-gen des Gerätes bei den Sendemasten schon sein, sonst wären wir nicht erreichbar. Auch die monatliche Abrechnung oder die Berechnung des verbrauchten Daten-Volumens benö-tigt das Mitprotokollieren von Nutzungs-Daten.
Anders sieht das vielleicht beim Surfen im Internet aus. Die meisten Browser führen eine Historie (Verlauf) zu den besuchten Seiten über einen bestimmten Zeitabschnitt. Das sind oft mehrere Tage oder Wochen. Will man wissen, welche Seite man noch mal vorgestern be-sucht hat, dann ist die Historie eine feine Sache.
Für fremde Personen ist die Historie auch sehr informativ. Welche moralisch verwerflichen Seiten hat jemand besucht, wer wurde gegoogelt, war man auf der SocialMedia-Seite einer heimlich geliebten Person usw. usf.? Da steckt viel Streitpotential dahinter.

anonymes Surfen

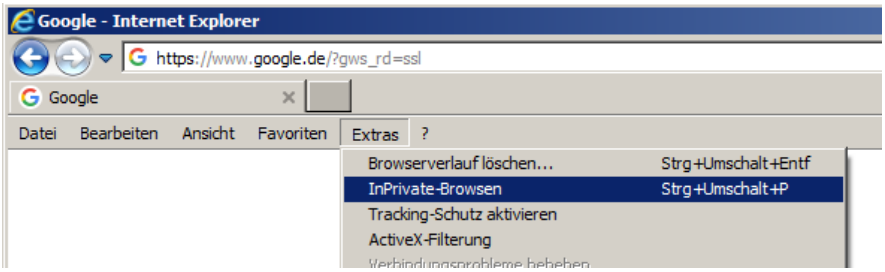
Außer technischen Vorkehrungen und Einstellungen muss aber auch sein eigenes Surf-Verhalten anpassen. Wer goggle als Startseite hat und dort jede andere Webseite als Such-Begriff eingiebt, der hinterlässt ausreichend Datenspuren. Da kann man sich dann andere Maßnahmen gleich sparen.

Nicht alles im Internet ist für jeden zugänglich. Firmen und Organisationen haben ihre eige-nen Kanäle zum Datenaustausch. Diese Daten gehen ja eigentlich auch keinem anderen etwas an. Aber es gibt auch das sogenannte Darknet (dark net). Dieses Netz-Teil ist über die gängigen Suchmaschinen nicht zugänglich. Selbst die Server bzw. deren Internet-Adressen sind bei Eingabe in den Browser nicht erreichbar.
Das liegt daran, dass es bestimmte Rechner im Internet gibt, die Namen und Adressen der verschiedenen Server kennen. Durch diese Rechner können Anfragen an eine bestimmte Adresse auch an den richtigen Server weitergeleitet werden. Der Service, der Namen von Servern und deren Adressen vermittelt heisst DNS (Domain-Name-Service). Entweder man muss diese DNS-Server vor dem ersten Surfen auf dem Gerät oder einem Router einstellen. in modernen Sytemen wird der Service automatisiert vergeben. Man kann seinen aktuellen DNS-Server immer abfragen.
Server des Darknet sind bei den regulären DNS-Servern nicht gelistet. Das Darknet hat sei-ne eigenen DNS-Server und sogar eigene Suchmaschinen.

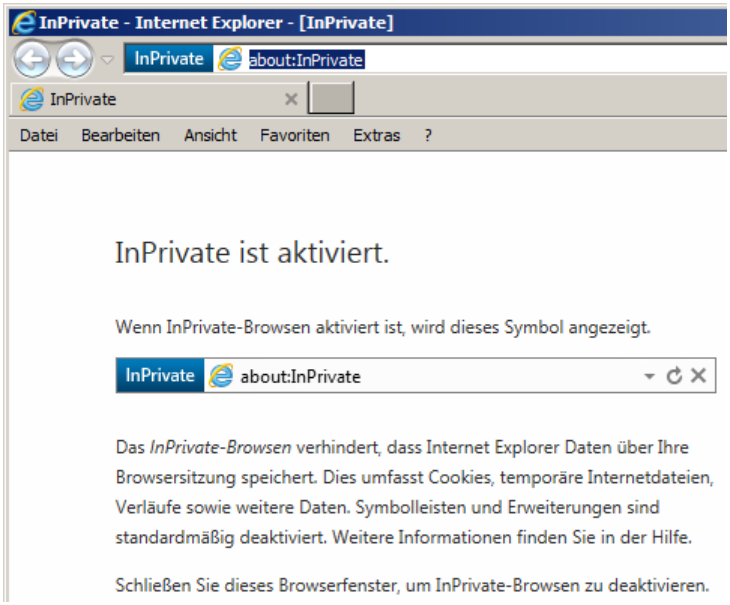
Browser für anonymeres Surfen einrichten / nutzen

microsoft ie / Internet-Explorer

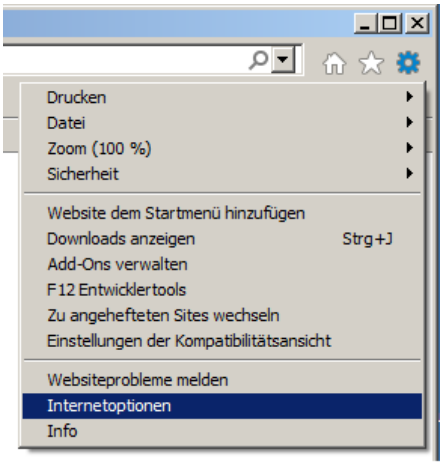
Die einfachste Variante ist es, den Internet-Explorer in einem speziellen Modus zu starten. Dieser heisst "InPrivate".



Beim Starten dieses Modus über "Extras" "InPrivate Browsen" wird eine neue Browser-Sitzung geöffnet.Vor der Adresszeile steht nun ein spezieller Eintrag, der über den Modus informiert. Beendet man den Internet-Explorer ist der InPrivate-Modus beendet und die alten Internet-Optionen gelten wieder.



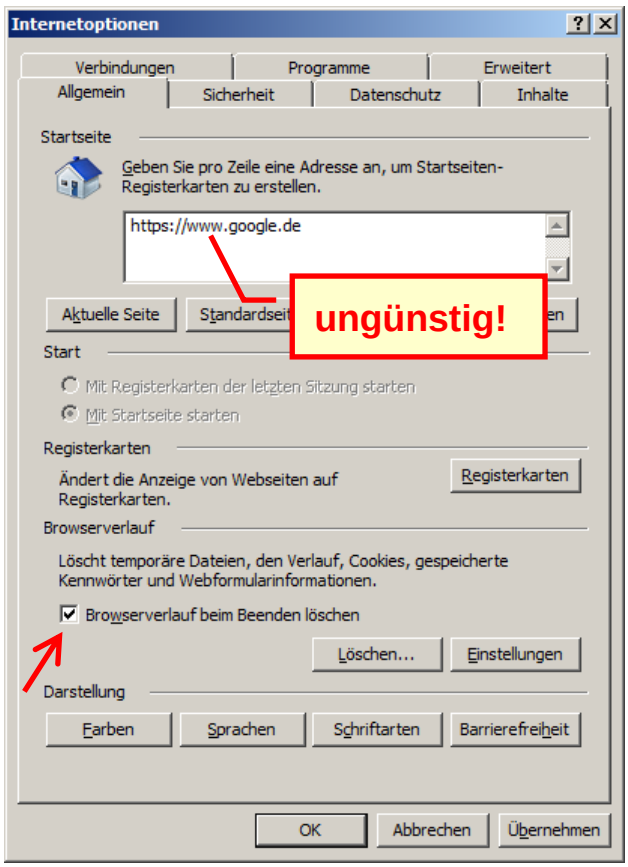
Wer keine Menüleiste eingestellt hat, kann das "Extras"-Menü auch schnell über das Zahlrad links oben im Browser-Fenster aufrufen.



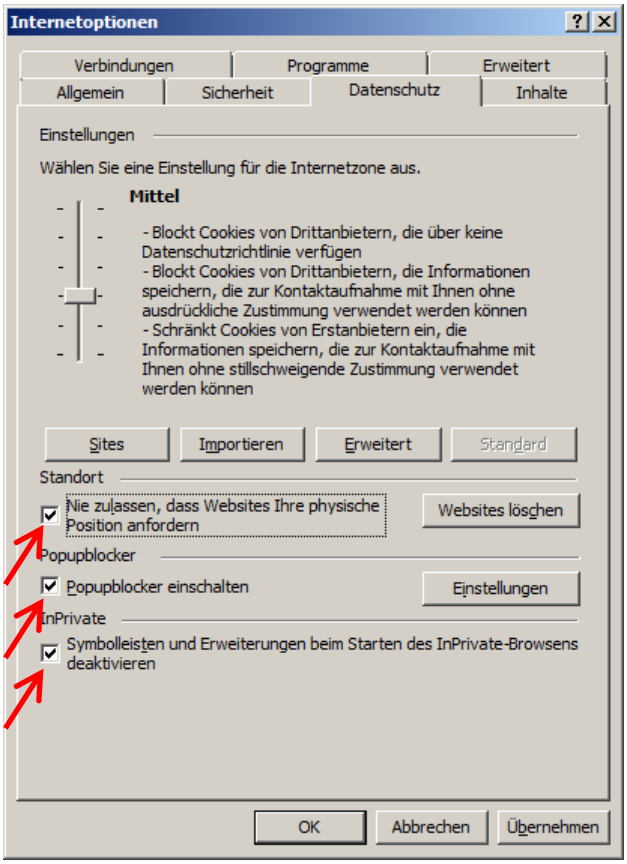
Viele der Menü-Punkte sind allerdings hinter den oberen Menü-Einträgen versteckt. Zum händischen " Browserverlauf löschen …" braucht man den Menü-Eintrag "Sicherheit".

Im folgenden Fenster kann man nun auswählen, was genau gelöscht werden soll.
Die hier gezeigte Auswahl ist eine Empfehlung. Jeder Nutzer muss für sich entscheiden, welche Daten auf dem PC gespeichert bleiben sollen.
Wer z.B. mit seinen Passwörtern nicht klar kommt, sollte sich das Löschen der Passwörter gut überlegen.
Will man den Internet-Explorer dauerhaft sicher(er) einstellen, dann sind mehr Einstellungen notwendig.
Die Feineinstellungen sind unter "Extras" "Internetoptionen" verfügbar.

Die Einstellungen sind z.T. sehr speziell und schränken teiweise auch den "normalen" Browser-Betrieb ein. Am Besten nimmt man die Einstellungen jeweils einzeln vor und probiert dann die Wirkung auf die selbst am Meisten genutzten Internetseiten.
Wenn alles gut läuft, lässt man die Einstellung, ansonsten nimmt man sie ev. zurück.
In vielen Fällen ist die Hilfe von Erwachsenen oder älteren Geschwistern hilfreich.
Einige der Optionen besprechen wir hier kurz. Dabei zeigen wir auch ungünstige Einstellungen auf.
Vielfach ist z.B. schon google als Startseite eingestellt. Dadurch wird schon die erste Spur gelegt. Da man die aktuelle Seite von google abruft, weiss goggle jetzt auch darüber Bescheid, das an einem bestimmten Rechner unter einer bestimmten Internet-Adresse ein ev. mit einem Namen angemeldeter Nutzer sitzt.
Bei der Startseite sollte man also besser seine eigene Homepage oder eine leere Seite einstellen.
Eine gute Vermeidung von Datenspuren auf dem eigenen Rechner ist die Option "Browserverlauf beim Beenden löschen". Solange man das Browser-Fenster nicht schliesst, kann man alle Vorteile der internen Speicherung von Datenspuren



nutzen – mit dem Beenden des Internet-Explorers werden die Datenspuren gelöscht.



Wer nicht möchte, dass fremde Internetseiten immer schon "wissen" an welchem Ort man sich befindet, der sollte "Nie zulassen, dass Websites Ihre physische Position anfordern". Ganz verhindern lässt sich das aber nicht, weil eben schon über die Internet-Adressen der Internet-Anbieter (Provider) eine gewisse Lokalisierung möglich ist.

Der Popupblocker verhindert die lästige Werbung in kleinen Zusatzfenstern. Man kann aber auch spezielle Einstellunge für bestimmte Internetseiten vornehmen. Dazu nutzt man dann den "Einstellungen"-Button neben der Option.

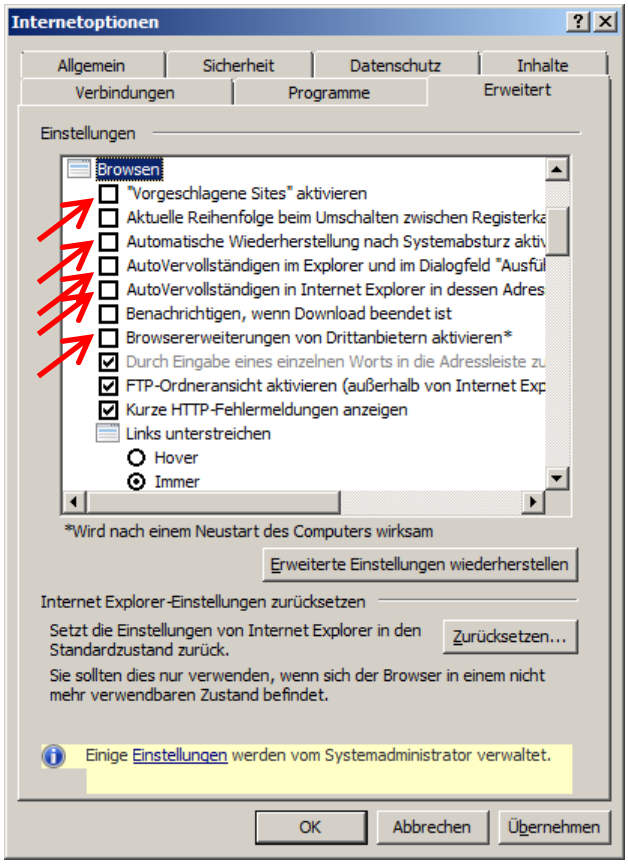
Vor allem zusätzliche Symbolleisten – wie z.B. die "google"- oder "WEB.DE Mailcheck"-Leiste – liefern und speichern Informationen des eigen Computers an die Leisten-Anbieter. Deshalb sollte man diese unbedingt während "InPrivate"-Sitzungen deaktivieren. Es nützt nichts, wenn man auf dem heimischen PC, den man kontrollieren kann, alles sauber hält, im Internet aber die Datenspuren mit der Gießkanne streut.

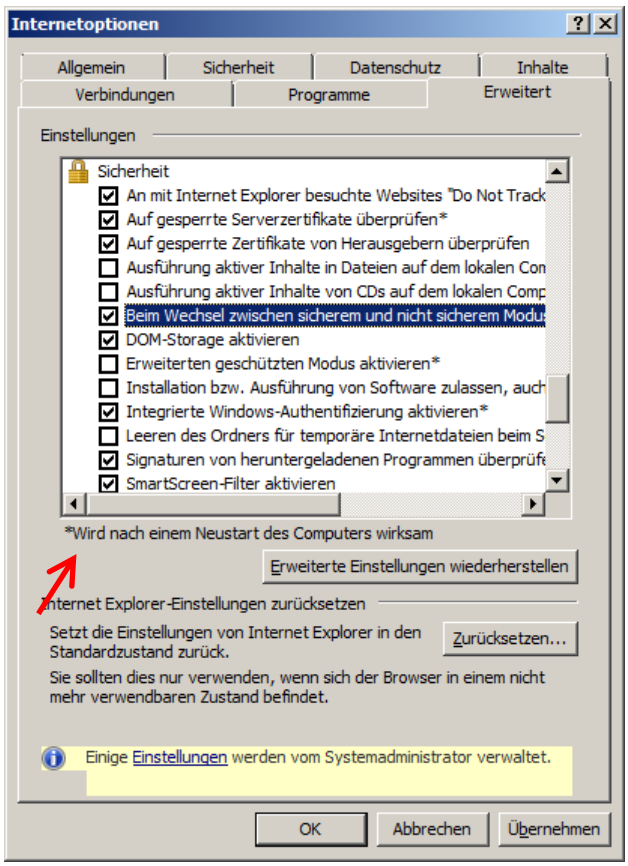
Das Vermeiden von Datenspuren ist ein sehr vielschichtiges Problem. Leider kann man nicht immer nachvollziehen, welche Daten mit welchen fremden Symbolleisten-Erweiterungen ausgetauscht werden. Auch der Hersteller des Browsers kann das nicht, also bleibt nur ein Abschalten während privater Sitzungen.

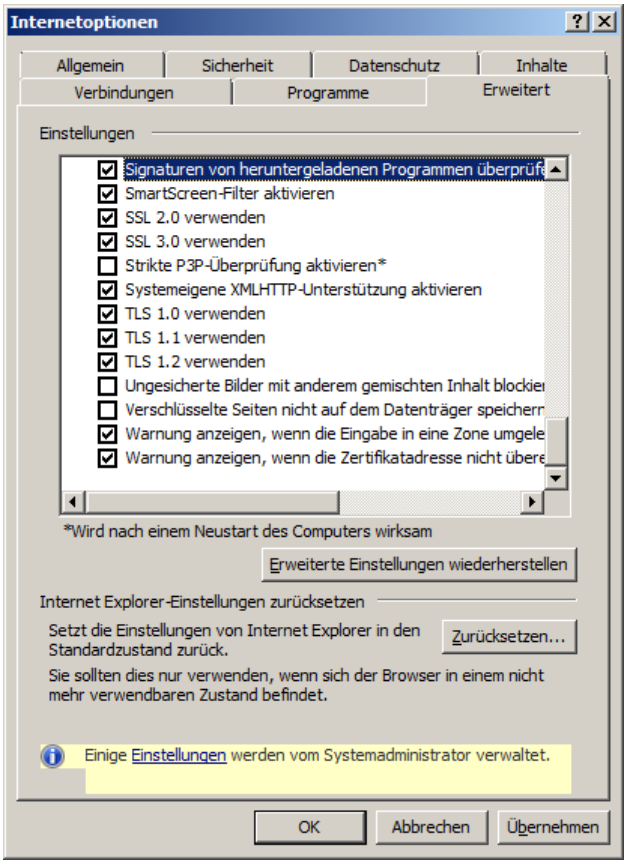
Das sieht man auch bei einigen speziellen Optionen unter dem Reiter "Erweitert".

Die Option ""Vorgeschlagene Sites" aktivieren" einzuschalten, bedeutet, dass auf irgendeinem Server im Internet die gemachten Adresse-Einträge mit Fremd-Vorschlägen abgeglichen werden. Datenspuren sind da gar nicht zu verhindern. Desweiteren muss man damit rechnen, auf eigentlich nicht gewollten Seiten zu landen.

Ähnlich verhält es sich mit den "AutoVervollständigen"-Funktionen. Entweder im Internet oder auf dem eigenen PC müssen die zu vervollständigenden Texte gespeichert sein – Datenspuren über Datenspuren.


Internetoptionen
Allgemein
Sicherheit
Datenschutz
Inhalte
Verbindungen
Programme
Erweitert
Einstellungen
Browsen
"Vorgeschlagene Sites" aktivieren
Aktuelle Reihenfolge beim Umschalten zwischen Registerka
Automatische Wiederherstellung nach Systemabsturz aktiv
AutoVervollständigen im Explorer und im Dialogfeld "Ausfül
AutoVervollständigen in Internet Explorer in dessen Adres
Benachrichtigen, wenn Download beendet ist
Browsererweiterungen von Drittanbietern aktivieren*
Durch Eingabe eines einzelnen Worts in die Adressleiste zu
FTP-Ordneransicht aktivieren (außerhalb von Internet Exp
Kurze HTTP-Fehlermeldungen anzeigen
Links unterstreichen
Hover
Immer
*Wird nach einem Neustart des Computers wirksam
Erweiterte Einstellungen wiederherstellen
Internet Explorer-Einstellungen zurücksetzen
Setzt die Einstellungen von Internet Explorer in den Standardzustand zurück.
Zurücksetzen...
Sie sollten dies nur verwenden, wenn sich der Browser in einem nicht mehr verwendbaren Zustand befindet.
Einige Einstellungen werden vom Systemadministrator verwaltet.
OK
Abbrechen
Übernehmen


Internetoptionen
Allgemein
Sicherheit
Datenschutz
Inhalte
Verbindungen
Programme
Erweitert
Einstellungen
Sicherheit
An mit Internet Explorer besuchte Websites "Do Not Track
Auf gesperrte Serverzertifikate überprüfen*
Auf gesperrte Zertifikate von Herausgebern überprüfen
Ausführung aktiver Inhalte in Dateien auf dem lokalen Con
Ausführung aktiver Inhalte von CDs auf dem lokalen Comp
Beim Wechsel zwischen sicherem und nicht sicherem Modus
DOM-Storage aktivieren
Erweiterten geschützten Modus aktivieren*
Installation bzw. Ausführung von Software zulassen, auch
Integrierte Windows-Authentifizierung aktivieren*
Leeren des Ordners für temporäre Internetdateien beim S
Signaturen von heruntergeladenen Programmen überprüfe
SmartScreen-Filter aktivieren
*Wird nach einem Neustart des Computers wirksam
Erweiterte Einstellungen wiederherstellen
Internet Explorer-Einstellungen zurücksetzen
Setzt die Einstellungen von Internet Explorer in den Standardzustand zurück.
Zurücksetzen...
Sie sollten dies nur verwenden, wenn sich der Browser in einem nicht mehr verwendbaren Zustand befindet.
Einige Einstellungen werden vom Systemadministrator verwaltet.
OK
Abbrechen
Übernehmen



mozilla Firefox

google chrome

opera

microsoft edge

Datenschutz – welche Daten müssen geschützt werden?

Der Begriff suggeriert, dass hier Daten geschützt werden. Der Schutz von Daten gegen z.B. versehendliches Löschen oder Manipilation durch Dritte wird unter dem Begriff Datensicherheit (→) betrachtet.

aktuelle Problemkreise
starke (soziale) Vernetzung
hohes Überwachungs-Interesse (bei Behörden, Firmen , …)

Der Datenschutz beschäftigt sich mit dem Rechten und Pflichten sowie der Verwendung von persönlichen Daten.

Definition(en): Datenschutz
Der Datenschutz umfasst alle Regelungen, Verfahren und zur Verhinderung der missbräuchlichen Nutzung von persönlichen Informationen in Datenverarbeitungssystemen.

Inhalte des Datenschutzes:

Schutz des rechtes auf informelle Selbstbestimmung
Schutz der Persönlichkeitsrechte
Schutz der Privatsphäre
Verhinderung des freien Datenaustausches zwischen Behörden, Firmen und Organisationen
Verhinderung von Daten-Monopolen bei Firmen, Behörden, …
Einschränkung und Begrenzung von Überwachungs-Maßnahmen
Einschränkungen hinsichtlich des Speicher-Zeitraumes (Recht auf Vergessen)

Bundesdatenschutzgesetz (BDSG)
Grundidee des Gesetzes ist die Schaffung einer Machtgleichheit zwischen den Interessen einer Person und denen von Organisationen (Behörden, Firmen, …)

in Zeiten superschneller Rechner und praktisch unbegrenzter Daten-Speicher ist eine unendliche Speicherung von Daten kein Problem mehr

basiert / begründet sich auf der Datenschutz-Grundverordnung der Europäischen Kommission (historisch geht es auf die Datenschutzrichtlinie von 1995 zurück
die legte Mindeststandards für alle Mitgliedsländer der EU fest
heute sehr umfangreiche Regelungen festgelegt, die auch den Umgang mit Personen-Daten im Datenverarbeitungs- und Kommunikations-Sektor mit einschliessen

Landesdatenschutzgesetz (für Mecklenburg-Vorpommern:)

Datenschutz-Gesetze bzw. –Verordnungen für spezielle Bereiche
z.B. Datenschutz-Regelungen für den Umgang mit Personen-Daten im Schulbereich (z.B. für Meckelnburg-Vorpommern:)

Gesetze: Datenschutz (augewählte Auszüge)

Die Datenschutz-Gesetze gelten übrigens nicht nur für Firmen oder Organisationen, sondern sind für alle verbindlich.

Hauptprinzipien des Datenschutzes:

- **Verbot mit Erlaubnisvorbehalt** — Daten dürfen nicht ohne Rechtsgrundlage und Erlaubnis (durch die betroffene Person (od. deren gesetzliche Vertreter) verarbeitet werden
- **Datensparsamkeit / Datenvermeidung** — es sollen möglichst wenig Daten erfasst, verarbeitet und gespeichert werden
- **Erforderlichkeit** — es sollen nur die Daten erfasst und verarbeitet werden, die für die Aufgabe / Funktion notwendig sind
- **Zweckbindung** — die erfassten Daten dürfen nur für den erlaubten Zweck benutzt werden, eine Weitergeabe an Dritte ist nicht erlaubt

Definition(en): persönliche Daten

aktuelle Probleme / Grauzonen / …

- Anschauen von Pornographie
- Gefahren für Dritte
- Drohnen (mit Kamera's)
-

Doxing - Mißbrauch von Daten

Veröffentlichung von Klar- oder Voll-Namen, Adressen, Telefonnummer bei anonymisierten Accounts
Krankheiten,

dazu gehören auch das politsch oder moralisch motivierte "Leaken" bzw. "Whistleblowing"

CyberMobbing / Hate Speech

Bedingung in modernen Kommunikations-Systemen:
Das Netz vergisst nie!

Das Netz vergisst nie!

Daten werden in Netzen mehrfach gespeichert, verteilt und auf externen Systemen gesichert.

Festlegung allgemeingültiger Regeln

- nur die notwendigsten persönlichen Daten veröffentlichen
- auf den Kommunikations-Partner abgestimmte und eingeschränkte Freigabe von persönlichen Daten
- Abstellen von automatischen Upload-Funktionen (für Bilder- oder Video-Ordner)
- die gleiche Cloud / Tauschböse / App niemals für private und schulische / dienstliche Kommunikation gemeinsam benutzen
- Foto's / Video's mit intimen Inhalten – wenn überhaupt – ohne persönliche Erkennungs-Merkmale
- keinem Anderen Schaden (egal welche Art) entstehen lassen
- Nutzung fremder Daten nur mit dem Einverständnis des Besitzers / Urhebers
- keine Persönlichkeits-verletzende Veröffentlichung oder Komposition von Daten
- sich beim Verfassen von Texten / Kommentaren in die Situation / Lebenswelt des Anderen hineindenken und überlegen, wie man sich selbst fühlen würde
- bei Problemen / Beleidigungen / Daten-Missbrauch sofort die Eltern oder andere Vertrauens-Personen informieren / kein übereiltes Löschen (Beweise erhalten)

- einem Kommunikations-Partner, der sich daneben benommen hat, kurz und ganz deutlich machen, dass man dieses nicht möchte, er dass sofort unterlassen soll sowie problematische Daten sofort löschen soll
- nicht auf inhaltliche Streits einlassen, solche Kommunikationen arten immer aus!
- Straftaten (Betrügerein, Beleidigungen, Lügen, …) sofort melden (ansonsten macht man sich ev. der Mittäterschaft schuldig)

Links:
http://lfmpublikationen.lfm-nrw.de/index.php?view=product_detail&product_id=442 (Info-Mat. für Lehrer und Eltern)
https://no-hate-speech.de/ (Mat. zur Kampagne des Europarates)
https://www.hanisauland.de (Kinderseite der Bundeszentrale für politische Bildung) ☆☆☆☆☆

Sicherheitstipps für das Internet

(von K. ROSENTHAL (Q: www.hanisauland.de), Bundeszentrale für politische Bildung)

1. Suche mit Kinder-Suchmaschinen!
2. Richte dir eine E-Mail-Adresse für Kinder ein!
3. Triff dich nicht mit Cyber-Freunden!
4. Halte dein Passwort geheim!
5. Schütze deine Daten!
6. Chatte fair – kein Mobbing!
7. Lade nie allein etwas auf den Computer!
8. Kaufe nichts im Internet!
9. Zeige nicht alles von dir!
10. Hole Hilfe, wenn nötig!

Jahrgangsstufe 6

4. Entscheidungen treffen und Spiele gestalten

Daten verarbeiten

- Variablen verwenden
- das Prinzip Eingabe-Verarbeitung-Ausgabe von Daten erläutern

Hierzu gehören das Ein- und Ausgeben von Daten, die Zuweisung eines Wertes zu einer Variablen und das Operieren mit Daten.

Algorithmische Grundbausteine verwenden

- *Bedingungen für die Steuerung des Programmablaufs formulieren*

Bedingungen erzeugen einen Wahrheitswert und entstehen z.B. durch die Verwendung von vergleichsoperatoren (größer als, kleiner als, ist gleich) oder von logischen Operatoren (nicht, und, oder)

- *Verzweigungen interpretieren und verwenden*
- *kopfgesteuerte Schleifen interpretieren und verwenden*

Die SuS wählen entsprechend der Aufgabenstellung zwischen kopfgesteuerten Schleifen mit selbstdefinierten Bedingungen und Zählschleifen.

Algorithmen verstehen

- *Programme im Hinblick auf Korrektheit beurteilen*
- *die Ursache von Programmfehlern systematisch erkunden*

Die SuS testen das Verhalten des Programms für unterschiedliche Eingaben. Sie nutzen Techniken der Nachverfolgung des Programmablaufs und der Variablenwerte.

Vorschlag zur inhaltlichen Vertiefung:
Erhöhung der Komplexität der Bedingungen oder Vergrößerung der Schachtelungstiefe der Strukturen

Problem-Fragen für Selbstorganisiertes Lernen

Machen Computer immer exakt das Gleiche? Können Programme unterschiedliche Dinge tun?
Kann der Computer Entscheidungen fällen? Kann der Computer doch denken?

Wie kann man Algorithmen beurteilen?
Arbeitet jeder Algorithmus exakt? Ist jeder Algorithmus immer korrekt?
Machen Computer Fehler?

4.1. das EVA-Prinzip

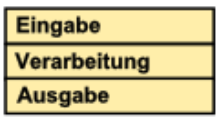
EVA-Prinzip der Informatik

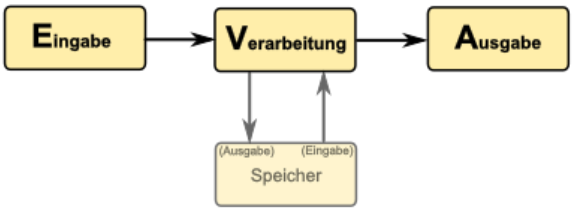
gleiches Prinzip auch bei uns Menschen
z.B.: sehen → denken → handeln

oder in der Industrie
Rohstoff → Herstellung → Produkt

Block-Schreibweise

oder auch Struktogramm genannt





4.2. Erstellen kleiner Spiele

das Variablenkonzept verwenden

Verzweigungen interpretieren und implementieren

Kopfgesteuerte Schleifen interpretieren und implementieren

Programme in Hinblick auf Korrektheit beurteilen und Ursachen von Programmfehlern systehmatisch erkunden

moderne ikonisch-symbolische Programmierung (Block-Programmierung)

Snap!
Scratch
Blockly
MakeBlock
…

verbunden mit physical computing (Benutzung von IO-Bausteinen)
Snap4Arduino (→ http://snap4arduino.rocks/; online Version: http://snap4arduino.rocks/run/) od. S4A (→ http://s4a.cat/) od. ModKit (→ http://www.modkit.com/)

6.x.1.1. Programmieren mit Scratch

entwickelt vom MIT-Media-Lab () von Mitch RESNIK
als Programmier-Lern-Umgebung
Nummerierung der Versionen ist abenteuerlich, am Besten von der Webseite leiten lassen und die passenden Versionen dort anwählen (nicht um die Versionsnummern kümmern!)

Hinweis: die msi-Downloads können direkt gestartet werden oder für spezielle Fälle von der Komandozeile mit:

```
Scratch2_MSI.exe /s /x /b"c:\test" /v"/qn"
```

wobei c:\test der Ziel-Ordner der Installation ist

Links:
http://www.inf-schule.de/programmierung/scratch/scratchwelt (Programmier-Anleitung)
http://schuelerlabor.informatik.rwth-aachen.de/sites/default/files/dokumente/Modulhandbuch-Scratch.pdf (Handbuch zu Scratch)
https://scratch.mit.edu/projects/editor/?tutorial=getStarted (Tutorial des MIT; engl.)
http://ilearnit.ch/de/sb.html (Sammlung von Projekt-Ideen zu Scratch)
https://appcamps.de/unterrichtsmaterial/scratch/ (Scratch-Kurs bei AppCamps)
https://csfirst.withgoogle.com/de/home (CS First-Projekt von google)

6.x.1.1.0. Wege zu Scratch

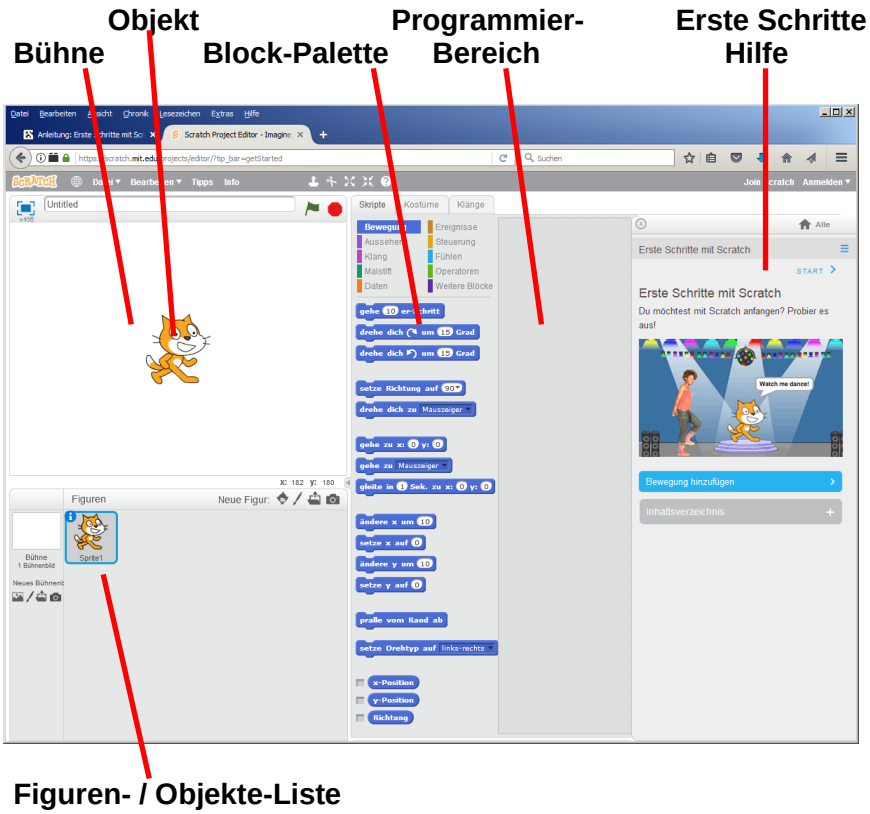
Scratch online
online: https://scratch.mit.edu

Vorteile
praktisch immer verfügbar, wenn man eine Internet-Verbindung hat
immer aktuell
keine Installation notwendig

Nachteile
Abhängigkeit vom Internet
bei Volumen- oder Zeit-gebundenen Internet-Verbindungen nicht günstig
es wird eine aktelle Version des Flash-Players (von Adobe®) benötigt

Hinweise / Bedingungen
im Internet-Explorer müssen TLS 1.0 bis 1.2 eingeschaltet werden

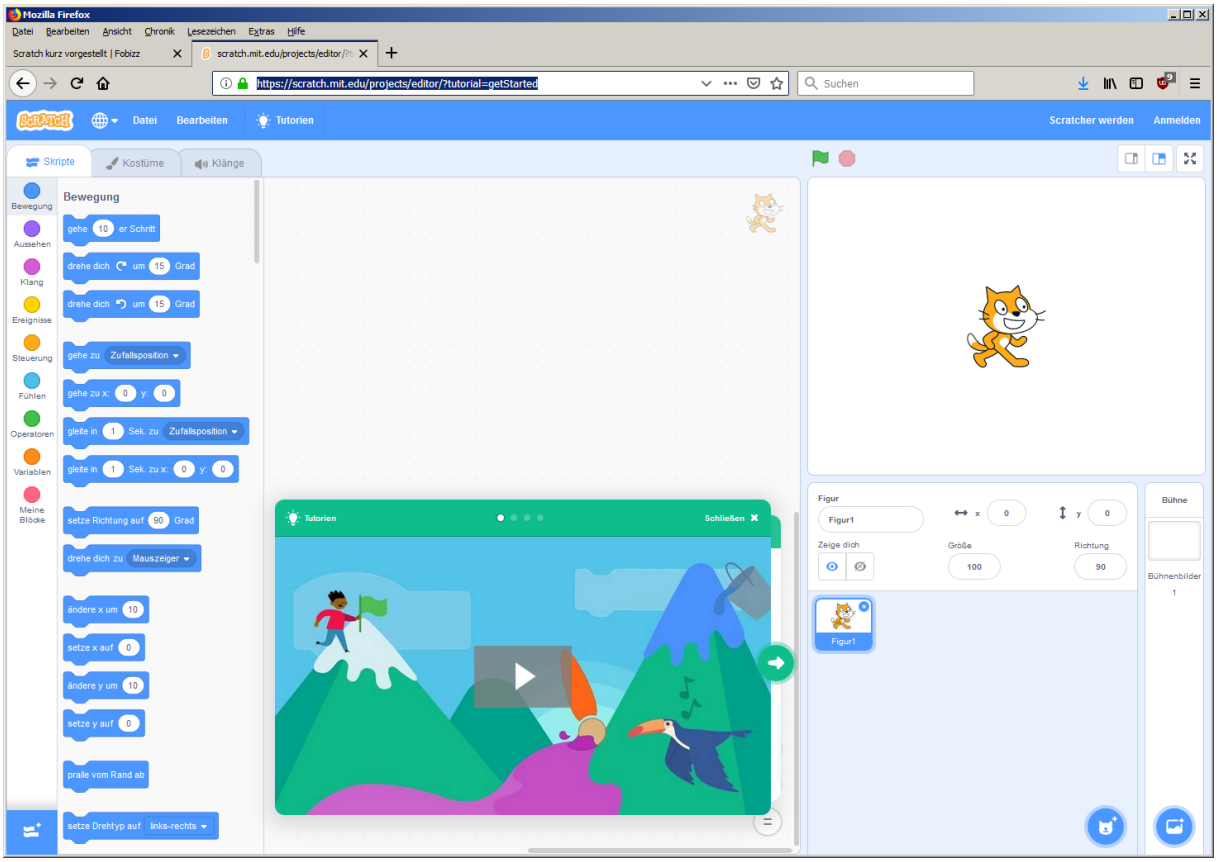
für Scratch 2 ist ein Flash-Player notwendig



Scatch in der modernen Version 3

Direkt-Start → https://scratch.mit.edu/projects/editor/?tutorial=getStarted

übersichtlicher, moderner, heller
auch gut für Touchscreen's geeignet
setzt aber Win 10 oder MacOS 10.1 voraus

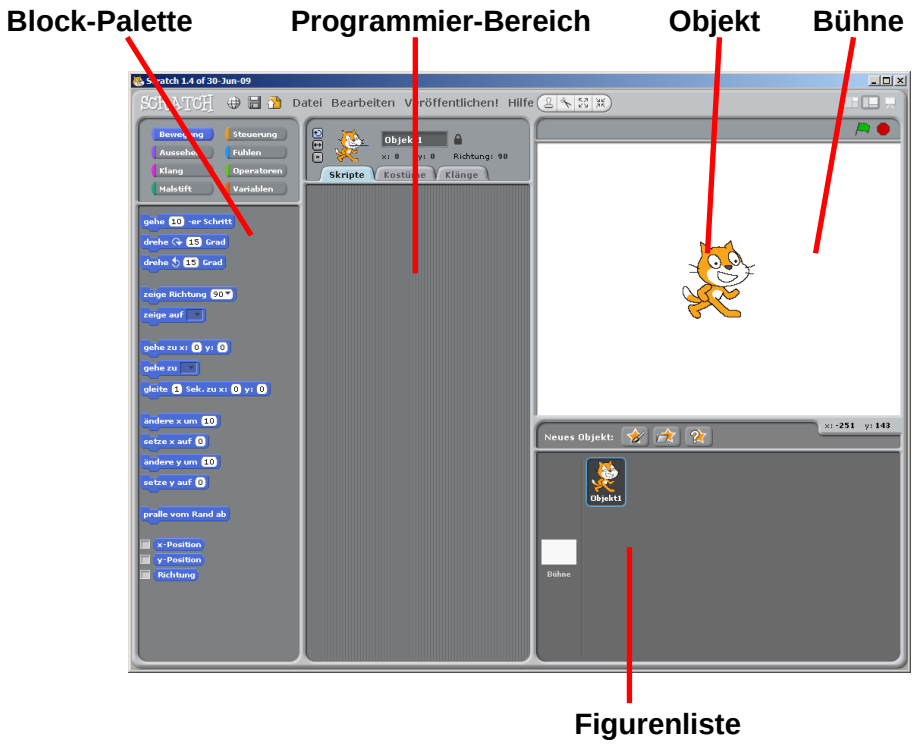


Scratch installieren / Scratch offline
downloaden von https://scratch.mit.edu/download; dort ist ein Link zum Download von Adobe AIR und man findet hier auch einige Materialien
(weiterhin verfügbar eine ältere Version für ältere Rechner oder bei problematischen Installationen)

Vorteile
relativ neue Version
unabhängig vom Internet

Nachteile
muss installiert werden (Administrator-Rechte notwendig)
Adobe AIR als Laufzeit-Umgebung muss vorher installiert werden

Für die offline-Version von Scratch – z.B. vom IoStick – ist nur die Verteilung der Bereiche anders. Es fehlt aber auch der Hilfe- / Erste-Schritte-Bereich.



Aufgabe:

Prüfe, welche Scratch-Variante auf Deinem Arbeitsplatz am Besten funktioniert und verwende diese dann!

neue Version 3 ist auch als Desktop-Version für die offline-Programmierung verfügbar
ev. wegen Datenschutz-Bestimmungen die unkompliziertere Variante
allerdings ist Win 10 die Mindest-Voraussetzung, was in den Schulen eher weniger vorhanden ist, hier dominiert Win 7

downloaden von https://scratch.mit.edu/download;

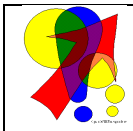
Eventuell ist auch die Benutzung von Snap! eine bessere Alternative. Wir besprechen das Programm und die Programmierung mit Snap! in einem nachfolgenden Kapitel (→ 5.x.1.2. Programmieren mit Snap!).
Vor allem für ist Snap! besser geeignet.

Scratch vom Rasperry Pi mit Pixel-Desktop od.a. ähnlichen Linux-Distributionen

bei den meisten Distributionen ist Scratch schon fertig installiert
ansonsten ist ein Nachinstallieren meist sehr leicht möglich

Vorteile
alles fertig / Wohlfühl-Umgebung

Nachteile
man braucht einen Raspberry Pi



Tipp für Scratch-Besessene / echt Interessierte:
Wenn Dir das Programmieren wirklich Spass macht und Du gar nicht aufhören möchtest, Du noch hunderte von Ideen hast und sowieso noch einen Wusch zu Weihnachten oder zum Geburtstag frei hast, dann wünsch Dir doch mal so einen Raspberry Pi.
Zeig einfach mal Deine ersten Skripte und erzähle dem "Weihnachtsmann" oder Deinen Eltern, was Dich am Programmieren und am Rasp Pi so begeistert. Du wirst sehen, das wirkt Wunder.

Scratch vom IoStick / Scratch mobil

passende Version herunterladen und auf einem USB-Stick installieren (lassen)
der USB-Stick enthält viele Schul-relevante Programme und kann auf jedem Windows-Rechner benutzt werden, auch aus eines Gast-Anmeldung heraus

Vorteile
man hat Programm und Daten immer zusammen auf einem USB-Stick

Nachteile
bei Beschädigung / Verlust des USB-Stick ist alles weg

Block-Gruppen	Farbe	englische Bezeichnung	enthält z.B. …	
Bewegung		Motion		
Aussehen		Looks		
Klang		Sound		
Malstift		Pen		
Daten		Data		
Ereignisse		Events		
Steuerung		Control		
Fühlen		Sensing		
Operatoren		Operators		
weitere Blöcke		More Blocks		

Scratch-Sprache ändern:

- auf Globus-Symbol klicken
- Sprache auswählen

***Programm / Skript speichern** (offline-Version):*

- auf Disketten-Symbol klicken
- Speicher-Ort auswählen
- Datei-Namen festlegen
  - "Projektautor" und "Über dieses Projekt:" (Projekt-Beschreibung) sind optional
- "OK"

Programm / Skript speichern (online-Version) / herunterladen:

- bei "Datei" "Herunterladen auf deinen Computer" anklicken
- Speicher-Ort auswählen
- Datei-Namen festlegen
- "Speichern"

***Projekt / Skript öffnen** (offline-Version):*

- bei "Datei" "Öffnen …" Speicher-Ort auswählen
- Datei aus Liste auswählen
- "OK"

Projekt / Skript öffnen (online-Version) / hochladen:

- bei "Datei" "Hochladen von deinem Computer" Speicher-Ort auswählen
- Datei aus Liste auswählen
- "Öffnen"

Aufgabe:

Erkunde die Bühne!

Wo beginnt die Bühne (Koodinaten-Ursprung)? Wie groß ist die Bühne ungefähr (Breite / Höhe)?

Bühne erkunden:

- aktuelle Position des Maus-Zeigers steht rechts unten an der Bühne (x- und y-Position)
- Maus-Zeiger oder Objekt (z.B. Katze auf der Bühne) mit der Maus anklicken und über die Bühnen-Fläche schieben → Positions-Anzeige beobachten!

Programm Start-fähig machen / Start-Symbol einfügen:

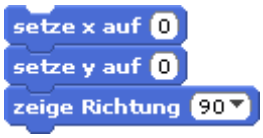
- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- obersten Block "Wenn angeklickt" in den Programmier-Bereich ziehen

Objekt bewegen:

- oben in der Block-Palette "Bewegung" auswählen
- passenden Bewegungs-Block anklicken und in den Programmier-Bereich ziehen
  - ev. gleich mit anderen Blöcken verbinden / an die richtige Position ziehen
- Wert für Schritte, Drehwinkel oder Positionen eingeben (weiße Felder)

Ausgangs-Situation wieder herstellen / zurücksetzen:

- oben in der Block-Palette "Bewegung" auswählen
- Block "setze x auf 0" als ersten Block nach dem "Grüne Fähnchen"-Block
- danach Block "setze y auf 0" einfügen
- Richtung des Objektes mit Block "zeige Richtung 90" festlegen



eigenes Programm / Skript starten / ablaufen lassen:

- das grüne Fähnchen rechts oben über der Bühne anklicken

eigenes Programm / Skript in Einzelschritten laufen lassen (Block für Block):

- im Programm-Menü "Bearbeiten" "Ausführung in Einzelschritten"
  - ev. "Einzelschritte einstellen …" z.B. auf "Bausteine hervorheben (langsam)"
- dann zum Starten das grüne Fähnchen rechts oben über der Bühne anklicken

eigenes Programm / Skript beenden / unterbrechen:

- auf den roten Kreis rechts oben über der Bühne klicken

ODER

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- einen der Blöcke "stoppe dieses Skript" (gilt nur für ein Objekt) ODER "stoppe alles " (gilt für alle Objekte / das Gesamt-Programm) in das Skript einbauen

Aufgaben:

1. *Erstelle ein neues Programm (Skript) mit dem Datei-Namen "FeldAblaufen" in Deinem eigenen Ordner!*
2. *Kombiniere jetzt die verschiedenen Blöcke so, dass die Katze (/das Objekt) einmal am Rand um das gesamte Feld (/ die Bühne) läuft!*
3. *Speiche das fertige Programm am Ende ab!*

für Experten und zum Knobeln:

4. *Lasse das Objekt auch noch die beide Diagonalen der Bühne ablaufen!*

einen Klang / Sound abspielen:

- oben in der Block-Palette "Klang" auswählen
- passenden Klang-Block auswählen und zur Abspiel-Position in den Block-Stapel ziehen
  - ev. Klang ändern
  - ev. Geschwindigkeit anpassen (extra Block vor den Abspiel-Block einfügen)
  - ev. Lautstärke anpassen (extra Block vor den Abspiel-Block einfügen)
  - ev. als einen de letzten Blöcke "stoppe alle Klänge" einbauen

 (c,p) 2015 – 2020 lsp: dre

Aufgaben:

1. *Speichere das fertige Programm (Skript) vom vorherigen Aufgaben-Block für verschiedene Erweiterungen unter einem neuen Datei-Namen "FeldAblaufenExtra" in Deinem eigenen Ordner!*
2. *Das Objekt (/ die Katze) soll sich in den Bühnen-Ecken langsam einmal um mindestens 360° drehen (Richtung ist egal)! Dabei soll das Objekt mindestens einmal einen Klang abspielen (z.B. "Miau")!*
3. *Lasse am Ende des Feldablaufes einen Schlagzeug-Wirbel spielen!*

für Experten und zum Knobeln:

4. *Das Objekt (die Katze) soll beim Drehen (Aufgabe 2) zwischen den kleinen Dreh-Bewegungen die Klänge von laut nach leise abspielen!*

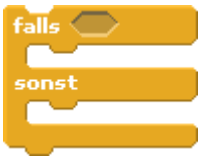
bedingte Ausführung von Programm-Teilen:

- oben in der Block-Palette "Steuerung" auswählen
- den "falls"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
- Bedingung (aus Bereich "Operatoren" oder "Fühlen") einfügen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Bedingung festlegen:

- vorher einen geeigneten "Steuerung"s-Block (mit Waben-förmiger Bedingungs-Stelle) in das Skript einbauen
- oben in der Block-Palette "Operatoren" auswählen
- einen Waben-förmigen Bedingungs-Block auswählen
- variable Felder ausfüllen
  - ev. Bedingungen durch logische Verknüpfungen kombinieren

alternative Wege erstellen:



- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "falls .. sonst"-Doppel-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
- Bedingung (aus Bereich "Operatoren" oder "Fühlen") einfügen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die obere Klammer ziehen
  - die alternativ / sonst auszuführenden Blöcke / Programm-Teile in die untere Klammer ziehen

Programm-Abschnitt eine bestimmte Anzahl oft wiederholen:

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "wiederhole … mal"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
  - die abgezählt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Programm-Abschnitte fortlaufend / immerzu wiederholen:

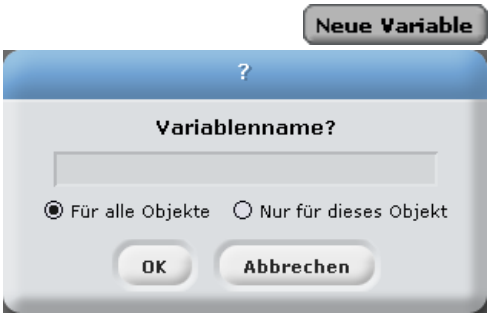
- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "wiederhole fortlaufend"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Programm-Abschnitte bedingt fortlaufend wiederholen:

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "wiederhole fortlaufend, falls …"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
- Bedingung (aus Bereich "Operatoren" oder "Fühlen") einfügen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Variable definieren (bekanntgeben):

- oben in der Block-Palette "Variablen" auswählen
- die "Neue Variable"-Schaltfläche klicken und im nachfolgenden Dialog den Namen der Variable eingeben



Variable verwenden:

- oben in der Block-Palette "Variablen" auswählen
- die Variable (hier z.B.: laenge) auswählen und in einen geeigneten Block hineinziehen

neues Objekt einfügen:

- in der Kopfzeile der Objekt-Liste "Neues Objekt aus Datei laden" auswählen
- in einen passenden Ordner wechseln
- Objekt auswählen

Achtung! *Jedes Objekt muss einzeln programmiert werden!*

Aufgaben:

1. *Erstelle ein neues Programm, in dem eine Maus (als Objekt) durch Drücken der Pfeil-Tasten gesteuert wird!*
2. *Verändere das Programm soweit, dass beim Erreichen eines Randes ein Klang ausgegeben wird!*

für Experten und zum Knobeln:

3. *Steuere die Hoch- und Runter-Bewegung des Objekt's durch die Umgebungslautstärke! (Mikrofon bzw. Head-Set notwendig!)*

Objekt-Größe ändern:

- Objekt auf der Bühne anklickenund mit rechter Maus-Taste Kontext-Menü aufrufen
- "Die Größe dieses Objektes ändern" auswählen und am Objekt mittels Skalierer die Ausdehnung kleiner oder größer ziehen

Objekt-Spur zeichnen (beginnen):

- oben in der Block-Palette "Malstift" auswählen
- Block "senke Stift ab" an Start-Position (im Skript) für Zeichnen setzen
  - ev. Farbe setzen
  - ev. Stiftdicke setzen

Objekt-Spur zeichnen beenden:

- oben in der Block-Palette "Malstift" auswählen
- Block "hebe Stift an" an Ende-Position (im Skript) setzen

Berührung von Objekten testen:

- vorher einen geeigneten "Steuerung"s-Block in das Skript einbauen
- oben in der Block-Palette "Fühlen" auswählen
- passende Berührungs-Bedingung in den Steuerungs-Block hineinziehen

Zufallszahlen verwenden:

- vorher einen geeigneten Block mit einem variablen Bereich (rundlich, weiß) in das Skript einbauen
- oben in der Block-Palette "Operatoren" auswählen
- Block "Zufallszahl von … bis …" an in den Block mit variablen Bereich hineinziehen

Aufgaben:

1. *Erstelle ein Programm, bei dem sich die Maus immerzu zwischen den Rändern hin und her und auch hoch und runter bewegt!*
2. *Speichere das Programm von Aufgabe 1 nochmals unter einem neuen Namen ab! Erweitere das Programm um eine Katze, die durch Tasten-Steuerung die Maus fangen soll! Immer, wenn die Maus einmal erwischt wurde (Berührung), soll ein Zähler hochzählen und die Maus zufällig an eine neue Position springen!*
3. *Öffne das Programm von Aufgabe 1 und speichere es wiederum unter einem neuen Namen ab! Verändere das Programm nun so, dass die Maus eine Spur hinterlässt!*

für Experten und zum Knobeln:

4. *Lege auf der Bühne einige Hindernisse an! Diese müssen nun von den Figuren auch umgangen werden.*

6.x.1.1.1. fortgeschrittenes Programmieren mit Scratch

Das Austauschen von Nachrichten dient der Kommunikation von Skripten / Objekten untereinander. Damit kann einem anderen Objekt etwas mitgeteilt werden oder ein Skript kann auf eine spezielle Nachricht eines anderen Skriptes / Objektes reagieren.

senden einer Nachricht:

- oben in der Block-Palette "Ereignisse" bzw. "Events" auswählen
-
  -
  -

empfangen von Nachrichten:

-
-
  -
  -

senden einer Nachricht und warten (*z.B. auf eine Reaktion / Antwort****):***

-
-
  -
  -

6.x.1.2. Programmieren mit Snap!

Nachfolger von Scratch / neu entwickelt
von Brain HARVEY und Jens MÖNIG
gleiches graphisches Programmier-Prinzip
für etwas ältere Schüler (nicht ganz so verspielt)
erste Zielgruppe waren die Studenten der Berkeley Universität in Californien
basiert neben Scratch auch auf der Programmiersprache Scheme
Scheme ist eine Ausbildungssprache für Informatiker
Prinzip der Block-Programmierung, Blöcke sind Bausteine, die nur in bestimmter Art- und Weise zusammenpassen bzw. ineinanderpassen. Dieses Passen entspricht den Syntax-Regeln, die hier quasi nebenbei betrachtet wird. Oft sind es gerade diese unendlichen vielen Regeln, die Anfänger verzweifeln lassen. Was darf miteinander kombiniert werden, was nicht?

Start im Browser oder über eine lokale Installation möglich.

Im Browser ruft man die Adresse https://snap.berkeley.edu/snapsource/snap.html auf.

Quellen:

https://snap.berkeley.edu

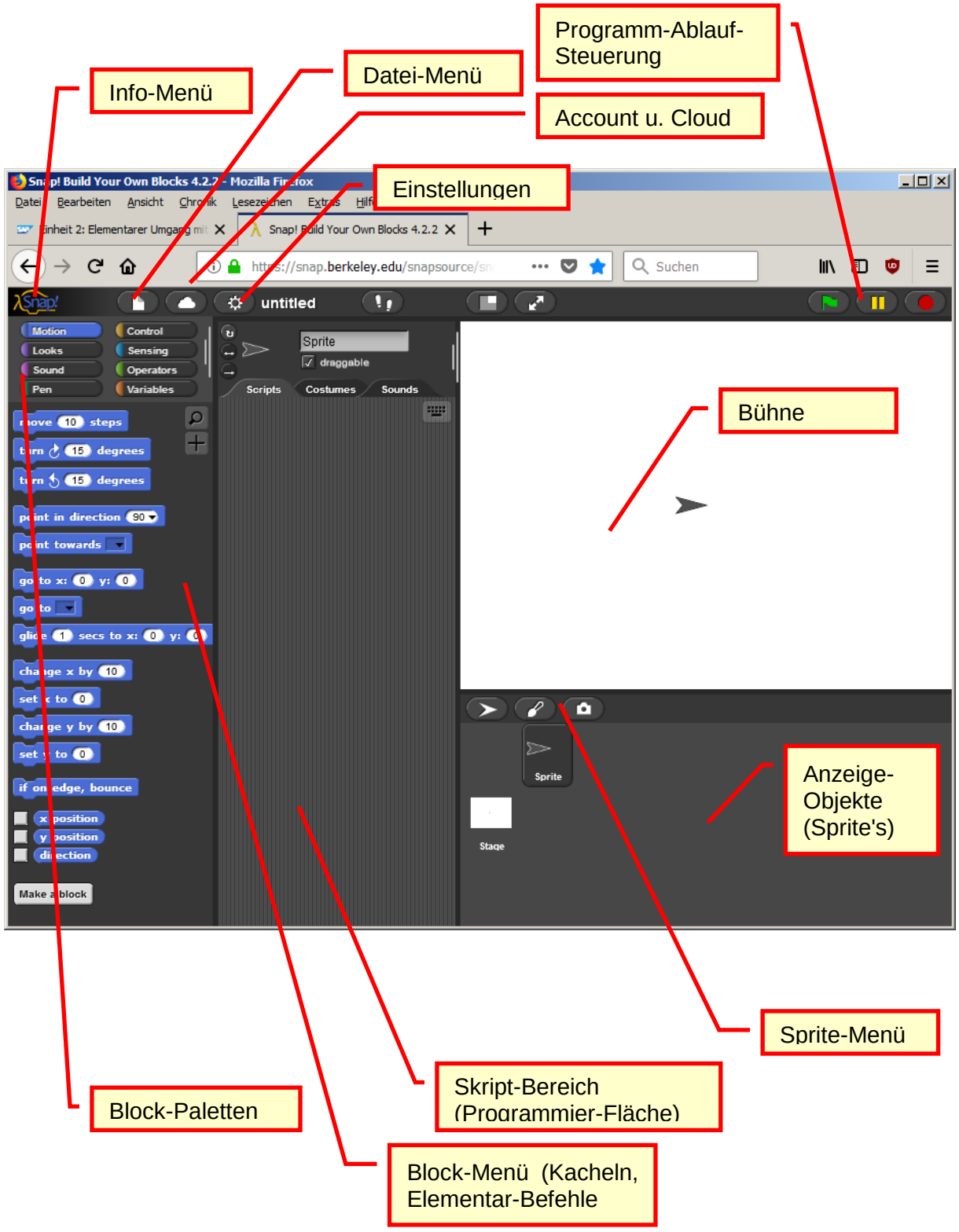
Mirror's:

https://media.mit.edu/~harveyb/snap
http://cs10.org/snap
http://bjc.edc.org/snapsource/snap.html

http://ddi-mod.uni-goettingen.de/InformatikMitSnap.pdf (freies Buch: MODROW: "Informatik mit Snap!)

http://ddi-mod.uni-goettingen.de/ProjekteZuInformatikMitSnap.zip (Programme zum MODROW-Buch)

Übersicht über die Programmier-Oberfläche



Für jeden Sprite gibt es eigenständige Programme und damit auch eigene Programmier-Flächen.

Block-Gruppen	Farbe	englische Bezeichnung	enthält z.B. …	
Bewegung		Motion		
Aussehen		Looks		
Klang		Sound		
Stift		Pen		
Steuerung		Control		
Fühlen		Sensing		
Operatoren		Operators		
Variablen		Variables		
zusätzliche Blöcke / Erweiterung				

Scratch-Sprache ändern:

- auf Globus-Symbol klicken
- Sprache auswählen

Benutzen einzelner Blöcke

Jeden Block aus den Kategorien können wir durch einfaches Klicken einmal ablaufen lassen. Wir wollen uns aber gleich daran gewöhnen, alles über die Programmier-Fläche zu steuern. Diese wird später dann unsere vollständigen Programme enthalten.
Die Befehls-Blöcke zum Benutzen aus dem Block-Menü auf die Programmierfläche gezogen werden. Einen Block, den man nicht mehr braucht kann man mit der Maus anklicken und dann in die Kategorie zurückschieben. Er ist dann gelöscht. Eine andere Möglichkeit ist das Löschen aus dem Kontext-Menü heraus. Dieses erhält man – wie üblich – durch klicken mit der rechten Maustaste auf den Block.
Alle Blöcke, die sich auf der Programmier-Fläche befinden können ebenfalls durch einfaches Klicken jeweils einmalig ausgeführt werden. Besonders gut ist das bei den Bewegungs-Befehlen zu beaobachten.

Aufgaben:

1. ***Positioniere auf der Programmier-Fläche die folgenden Befehls-Blöcke! Durch Klicken auf die Befehle kann dieser jeweils ausgeführt werden. Bewege den Sprite (Pfeil-Spitze) in die obere, rechte Ecke!***

2. ***Setze den Sprite auf die Ausgangs-Position! Zähle nun die Klicks, die Du brauchst, um die linke, obere Ecke zu erreichen! Wer schafft es im Kurs mit möglichst wenigen Klicks?***
3. ***Diskutiert im Kurs darüber, welches Vorgehen am Besten geeignet ist, um das Ziel zu erreichen!***
4. ***Notiert einen Vorschlag für einen Algorithmus!***

Kombination von Blöcken

Befehle lassen sich durch Verbinden zu einem neuen Block kombinieren. Das geht natürlich nur auf der Programmier-Fläche. Dazu bewegt man den einen Block so, dass er in den anderen einrasten kann. Eine heller Strich zeigt die mögliche Verbindung an.

Befehle-Block

Gleiche Blöcke muss man sich nicht jedes Mal aus den Katagorien herausholen. Man kann sie auch duplizieren. Dazu klickt man einen Block mit der rechten Maustatste an und wählt im sogenannten Kontext-Menü "duplizieren" aus. Der duplizierte Block erscheint auf der Programmier-Fläche und kann an geigneter Stelle verbaut werden.

Umringter (Groß-)Block

Die eingerasteten Blöcke bilden nun einen gemeinsamen Block. Will man den ganzen (großen) Block duplizieren, muss man ihn vorher markieren. In Snap! heißt das Umringen. Das lässt sich ebenfalls über das Kontext-Menü erreichen. Man erkennt das Umringen deutlich am grauen Rand um die Blöcke.

Nun kann der gesamte Befehls-Block wie gehabt, dupliziert werden.

Am Schluss muss der Block dann auch wieder entringt werden

Damit wir den Weg unseres Sprite's besser nachverfolgen können, soll dieser immer eine Spur hinterlassen.

Dazu eignet sich der Block "Stift runter" aus der "Stift"-Kategorie.

Der Befehls-Block muss auf der Programmier-Fläche nur einmal ausgelöst werden. Ab sofort hinterlässt der Sprite eine dünne Linie.

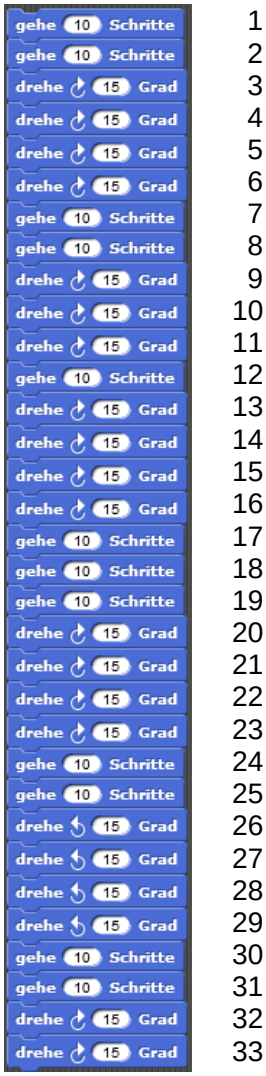
Das Gegenstück ist der Befehl "Stift hoch". Der Sprite hinterlässt dann bei einer Bewegung keine Spur mehr.

Mit dem Befehl "wische" lässt sich die Bühne von den Spuren befreien.

Aufgaben:

1. *Positioniere den Sprite in der Mitte der Bühne! Zeichne nun ein Rechteck mit einer Länge von 60 Schritten und einer Höhe von 40 Schritten! Alle notwendigen Befehle sollen als ein großer Block auf der Programmier-Fläche stehen!*
2. *Notiere die Befehle zuerst in Deinem Hefter! Suche nun eine effektivere Schreibweise für den Algorithmus, ohne die einzelnen Befehls-Blöcke zu ändern! (Ein Block "gehe 60 Schritte" ist also z.B. nicht zugelassen!)*
3. *Mit dem Block "wische" lässt sich die Bühne wieder von Spuren befreien. Nun soll eine Gruppe von zwei Quadraten gezeichnet werden, die einen gemeinsamen Startpunkt haben und deren Kante 10 und 30 Schritte gross sind!*
4. *Im nebenstehenden Programm ist dem Programmierer mindestens ein Fehler unterlaufen. Eigentlich soll das Programm ein regelmäßiges Sechseck zeichen und der Sprite wieder in der Ausgangs-Stellung stehen. Es entstand aber das folgende Bühnen-Bild:*

Korrigiere das Programm! Für nicht korrekte Befehle gib immer die Nummer des falschen Blockes und den richtigen Block an!
Fehlende Blöcke werden z.B. mit 2a, 2b usw. ergänzt. Bei Blöcken, die gelöscht werden sollen, wird einfach die Blocknummer gestrichen.



5. *Was macht eigentlich das nebenstehende Programm? Versuche die Leistung des Programm's zu ergründen, ohne es praktisch auszuprobieren!*

für Experten und zum Knobeln:

6. *Lasse den Sprite ein Parallelogramm mit einer Länge von 50 bzw. 30 Schritten und einem Winkel-Paar von 30 und 60 Grad zeichnen! Vergleiche Dein Parallelogramm mit dem von anderen Kursteinehmern! Sehen alle gleich aus?*

 (c,p) 2015 – 2020 lsp: dre

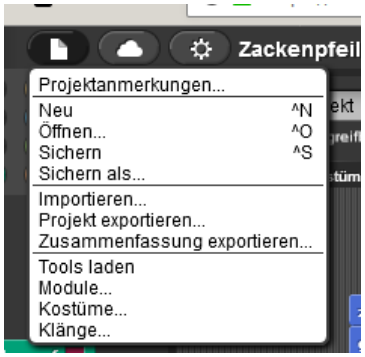
In den letzten Programmen haben uns die ewigen Wiederholungen und auch die kleinen Befehls-Schritte genervt. Es ist sicher jedem schon in den Sinn gekommen, die Zahlen in den Befehls-Blöcken anzupassen. Und das geht auch, wir nennen die Zahlen oder Ausdrücke, mit den ein Block etwas tut Argumente. Diese kann man üblicherweise ändern. Dazu gleich mehr ().
Trotzdem tauchen auch immer wieder gleiche Sequenzen auf. Für solche Wiederholungen gibt es ebenfalls vorgefertigte Blöcke, die in der Informatik auch Schleifen (Schlaufen,) genannt werden. Auch diesen werden wir einen größeren Abschnitt widmen (→).

Änderung der Argumente

Taucht in einem Programm eine Sequenz (Folge) von gleichen Blöcken auf, dann kann man prüfen, ob bei diesen die Argumente zusammengefasst werden können. Macht der Sprite fünfmal hintereinander 10 Schritte, dann können wir das zu einem Schritt von 50 Schritten zusammenfassen. Unser Programm wird dadurch deutlich kürzer.
Außerdem steigert sich die Übersichtlichkeit. Fehler in falschen Längen-Angaben usw. lassen sich jetzt schneller erkennen und korrigieren.
Ein weiterer Vorteil ist auch, dass wir nun jede beliebige Zahl als Argument verwenden können. Ein Rechteck mit einer Kantenlänge von 27 x 43 Schritte ist auf einmal kein Problem mehr.
Genau wie bei den Geh-Schritten können wir auch bei den Grad-Angaben für die Dreh-Befehle vorgehen. Aus sechsmal drehen um 15 Grad wird jetzt einmal 90 Grad. Da versteht doch auch gleich jeder, was gemeint ist.
Ein Zeichen-Programm für ein Rechteck (50 x 35) könnte dann so aussehen:

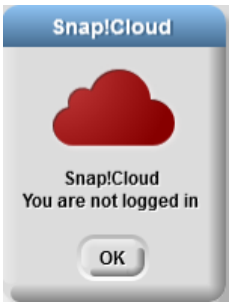
Die Schrittfolge ist sehr übersichtlich. Wenn nun jemend die Schritt-Länge für die Rechteck-Kanten anders gestalten möchte, dann findet er sehr schnell die passenden Stellen dafür.
Genauso leicht wird jetzt auch die Erstellung von speziellen Vierecken, wie Rhomben, Drachen-Vierecken oder Parallelogrammen.
Ein richtig gutes Programm sorgt auch für klare Start-Bedingungen. In unserem Fall soll der Sprite die Ausgangs-Position einnehmen und die Arbeitsfläche vorher einmal säubern. Zur Sicherheit senken wir auch den Stift ab.
Zum Abschluß unseres Programm's wird der Stift dann wieder abgehoben.
Das fertige Programm ist rechts zu sehen.
Argumente können also irgendwelche Zahlen oder Daten sein. In die abgerundeten Kästchen (Slot's) passen aber auch andere Funktionen. Auch sie sind an den aberundeten Seiten zu erkennen.
Im Folgenden nennen wir sie deshalb auch Argument- oder Funktions-Blöcke, um sie auch von den großen Befehls-Blöcken zu unterscheiden.

Programme speichern

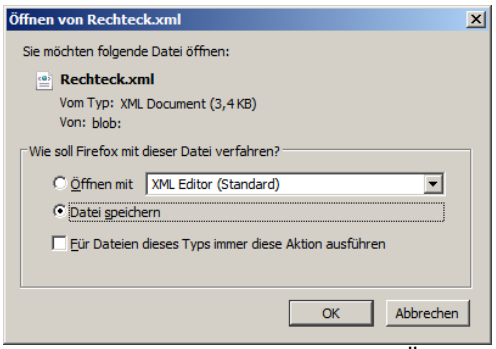


Dieses Programm wollen wir nun auch unbedingt für die Nachwelt aufheben. Wir müssen es also speichern. Dafür gibt es im Wesentlichen drei Möglichkeiten.
Als Erstes können wir in unserem Browser speichern. Das hat allerdings den Nachteil, dass das nächste Mal dieses Programm nur am aktuellen Rechner und mit dem aktuellen Browser benutzen können. Den Internet-Speicher dürfen wir auch nicht löschen oder löschen lassen.
Um im Browser zu speichern wählt man im Datei-Menü den Punkt "Sichern als …"
und im nachfolgenden Dialog die Schaltfläche "Browser".

Der Name des Programm sollte immer passend sein. Unter solchen Namen, wie "Programm_4" oder "Michael 8" kann man sich später nichts vorstellen.
Die zweite Variante besprechen wir hier für die Nutzer, die einen Account bei Snap! haben. Die Anlage eines Accounts ist für minderjährige Kursteilnehmer immer mit den Erziehungsberechtigten abzusprechen. Das sollte separat erfolgen.
Wer also einen Account hat, der kann sich über das Cloud-Menü "Anmelden …" und dann seine Datei speichern. Klappt das Einloggen nicht, oder gibt es noch keinen Account bei der Snap!Cloud, dann gibt es eine Fehler-Meldung (s.a. Abb. rechts).



Die sicherste Methode zum Speichern von Snap!-Programmen ist sicher das Sichern auf ein eigenen USB-Stick. Natürlich geh auch jedes andere Speicher-medium, wie z.B. die eigene Festplatte oder ein Netz-Laufwerk (z.B.: Home).
Jetzt wählen wir "Projekt exportieren …" im Datei-Menü. Je nach Browser und Browser-Einstellungen folgen noch weitere Dialoge. Dabei sollte man auf "Datei speichern" gehen und, wenn möglich einen geeigneten eigenen Ordner zum Speichern auswählen. Ansonsten landet die Datei im Download-Ordner des Rechners.

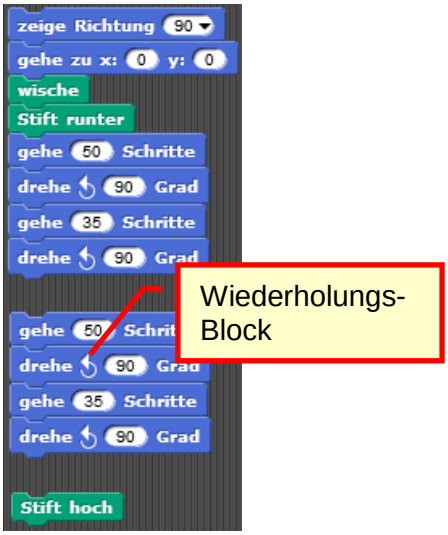


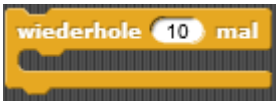
Nicht von der Beschriftung "Öffnen" irritieren lassen – es wird gespeichert!

Öffnen von abgespeicherten Programmen

Wiederholungen

Nun wenden wir uns aber wieder dem eigentlichen Programmieren zu. Wenn wir uns das Rechteck-Programm genau ansehen, dann wiederholen sich bestimmte Abschnitte zweimal.
Die Weiderholung ist in der nebenstehenden Abbildung durch Auftrennung der Sequenz dagestellt.
Informatiker sind der Inbegriff von "Schreib-Faulheit". Sie suchen immer möglichst kurze Programme.
In der Programmierung gibt es zum Verkürzen von sich wiederholenden Blöcken die sogenannten Schleifen (auch Schlaufen od. genannt).
Die Schleifen-Blöcke finden wir in der Kategorie "Steuerung". Der für uns interessante Block heißt "wiederhole 10 mal". Das wir das Argument aus 2 ändern können wissen wir schon.





Um den Wiederholen-Block an die richtige Stelle zu bekommen, trennen wir zuerst die zu wiederholende Sequenz heraus. Dann fügen wir den Wiederholungs-Block an die richtige Stelle und können anschließend die vier zu wiederholenden Blöcke in das "Maul" der Schleife schieben.
Die nicht mehr gebrauchte doppelte Sequenz löschen wir.

Als letzte Korrektur verändern wir noch die Anzahl der Wiederholungen auf 2 und schon haben wir ein Programm erstellt, auf dass jeder Informatiker Stolz sein kann.
Wie wir gleich sehen werden, gibt es noch einige weitere Verbesserungen, die wir gleich erklären (→).

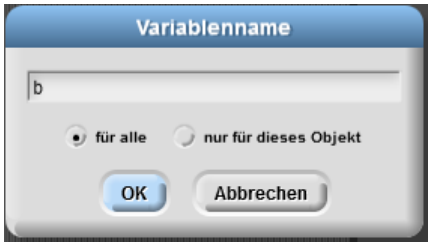
Aufgaben:

1. *Schreibe ein Programm zum Zeichnen eines regelmäßigen Sechsecks unter verwendung einer Schleife! (Der Sprite soll am Anfang und am Ende in der Ausgangs-Stellung sein.)*
2. *Erstelle ein Programm, dass ein beliebiges Flächen-Objekt mit mindestens einer Schleife erzeugt! Speichere es unter dem Namen "Fläche.XML" ab zu tausche es mit Deinem Nachbarn! Finde ohne einen Programm-Start heraus, welche Fläche das Programm des Nachbarn erzeugt! Prüfe dann nach!*

Arbeiten mit Variablen

Dass wir ein Programm brauchen, welches ein Rechteck genau mit den Maßen 50 x 35 Schritte zeichnen kann, wird wohl recht selten auftreten. Dagegen wird es sicher häufiger vorkommen, irgendwelche Rechteck zu zeichnen. Um nun ein anderes Rechteck zu zeichnen, müssten wir uns das alte Programm genau anschauen und rausbekommen, wo die Schritt-Maße eingetragen werden müssen. Bei kleinen, übersichtlichen Programmen ist das kein Problem. Wenn aber auch noch gleiche Zahlen (Argumente) für unterschiedliche Zwecke vorkommen, dann wird das Ändern schon zur Streß-Aufgabe. Man muss das Programm verstehen, um nicht an der falschen Stelle zu korrigieren. Da ist es oft einfacher, schnell ein neues Programm zu schreiben. Das wollen wir natürlich nicht.
Die Lesbarkeit von Programmen wird deutlich verbessert, wenn man mit Variablen arbeitet. Wir brauchen für ein Rechteck die Höhe und die Breite. Wir werden sie hier nur kurz h und b nennen.
Eine neue Variable muss man in Snap! zuerst einmal definieren, d.h. dem Programm bekanntgeben. Dazu gibt es in der Kategorie "Variablen" den Menü-Punkt "Neue Variable". Nach dem Klicken darauf öffnet ein Dialog zur definietion der Variable.

Besonders wichtig ist ein eindeutiger Name. Hier sollen zuerst einmal nur h und b reichen.
Nach dem "OK" erscheint die Variable als oranger Block in der Variablen-Kategorie. Auf der Zeichen-Fläche (Bühne) sehen wir links oben ebenfalls kleine Anzeigen für die Variablen. Hier wird uns nun ständig der aktuelle Wert angezeigt.
Was auffält ist die andere Form des Variablen-Blockes.



Ein Variablen-Block lässt sich nicht direkt in die Sequenz eines Programm's einbauen. Aber die Form passt exakt zu unseren Argument-Stellen. Ein Variablen-Block ist also mehr ein Stellvertreter-Block oder besser noch ein Ausdrucks-Symbol.
Somit können wir nun die Variablen h und b an die geeigneten Stellen schieben. Die ursprünglichen Werte (Argumente) verschwinden dann.
Die Variablen haben zum Zeitpunkt der Definition den Wert 0. Diese Anzeige auf der Bühne vedeutlicht uns das ja auch. Unser Rechteck-Programm mit einer Höhe und Breite von 0 aufzurufen, macht nicht wirklich sinn. Deshalb müssen wir im Programm vor der Benutzung der Variablen eine Wert-Zuweisung machen. Das muss nur einmal erfolgen. Der beste Platz ist dafür vor der Schleife.

Wir ergänzen unser Programm um zwei Blöcke vom Typ "setze ? auf 0".
Im Auswähler suchen wir uns die passende Variable aus uns tragen dann hinten noch den richtigen Wert ein.
Der so präparierte Block kann jetzt an eine Stelle vor der Schleife eingefügt werden. Man kann den Block abe auch nach dem Einfügen mit Variable und Wert versehen, die Reihenfolge bestimmt der Programmierer.
Man darf nur nicht die einzelne Arbeitsschritte vergessen. Entweder das Programm funktioniert dann nicht richtig oder es gibt eine Fehler-Meldung.

Das fertige Programm ist nun nahezu perfekt. Die echten Informatiker bemängeln jetzt nur noch, dass man zum Zeichnen eines bestimmten Rechtecks immer noch ins Programm eingreifen muss. Aber es ist jetzt nur noch an einzelnen Stellen notwendig und trotzdem recht gut verständlich.
Die Variablen h und b stellen für das gesamte Programm zwei symbolische Werte bereit. Veränderlich – wie es der Name Variable eigentlich suggeriert – müssen h und b eigentlich gar nicht sein. Man müsste eigentlich eher von Konstanten sprechen. Einmal festgelegt gelten sie im gesamten Programm. Ein direktes Konstanten-Konzept hat Snap! nicht. Man kommt auch gut so ein Konzept aus. Nur ist es manchmal besser vor einer versehentliche Veränderung einer Konstante gewarnt zu werden.
Neben dem einfachen Bereitstellen eines Wertes können Variablen noch viel mehr. Man kann mit ihnen rechnen (Operationen ausführen), man kann sie jederzeit neu belegen und viele lassen sich auch nur kurzzeitig benutzen und dann gleich wieder verwerfen.

Aufgaben:

1. *Verbessere das Rechteck-Programm noch weiter, so dass auch der Winkel als Variable eingebaut ist!*
2. *Kann man das Programm von 1. nun auch verwenden, um ein Viereck mit einem anderen Winkel als 90 Grad zu zeichnen? Erläutere Deine Meinung! Demonstriere ausgewählte Fälle!*
3. *Erstelle ein möglichst kurzes Programm (aber mit den Vor- und Nachbereitungen), dass ein Quadrat zeichnet!*
4. *Jemand braucht ein Programm, dass ein gleichseitiges Dreieck zeichnet. Kannst Du ihm da helfen, wenn er Dir die Maße für die Basis (b) und die Schenkel (s) sowie den Basis- und den Schenkel-Winkel (bw, sw) angibt? Wie würde Dein Programm dann aussehen?*

für Experten und zum Knobeln:

5.

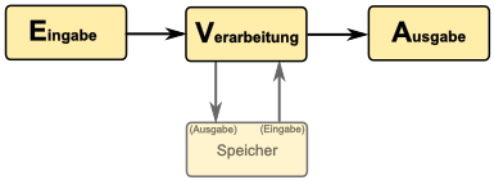
Eingaben

Unser Problem mit dem Rechteck-Programm war ja, dass wir immer noch in das Programm selbst eingreifen müssen, um ein Rechteck mit einer anderen Ausdehnung zu erhalten. Schön wäre es, wenn das Programm uns fragt, welche Maße es benutzen soll.

Dieses Abfragen ist de erste Teil im vielgenannten EVA-Modell. Nach dem EVA-Prinzip unktioniert die gesamte Datenverarbeitung.

Das **E** steht dabei für **E**ingabe, das **V** für **V**erarbeitung und das **A** für **A**usgabe. Diese drei Grundbausteine müssen sachlich immer in dieser Reihenfolge ablaufen. Zuerst werden Daten **e**ingegeben, diese werden dann **v**erarbeitet und letztendlich **a**usgegeben. In modernen Programmen werden die Bausteine aber auch mal gemischt.

Weiterhin wird bei diesem einfachen Modell nicht genug beachtet, dass Daten zur Verarbeitung imme auch gespeichert werden müssen (Das genau haben wir ja mit h und b getan!). In aktuelleren Versionen des EVA-Prinzip's spricht man deshalb auch lieber vom EVAS-Modell.



Auf den Speicher werden wir gleich noch eingehen.

In einer anderen Darstellungs-Form – dem Struktogramm - werden die Blöcke schön untereinander angeordnet. Das erinnert uns doch schon sehr an unser Snap!-Programm. Diese Kompatibilität von Struktogrammen und Block-Programmierung ist auch das Erfolgs-Modell solcher Programmier-Systeme, wie Snap!, MakeCode, Blockly usw. usf.

Kommen wir aber zu den Eingaben zurück. Um den Nutzer zu fragen, mit welchen Werten das Programm arbeiten soll, kann man den "frage … und warte"-Block verwenden.

Mit einer passenden Frage, die am als Argument formuliert, sieht die Bühne dann so aus.

Am Sprite erscheint eine Sprechblase mit unserem Aufforderungstext. Die eigentliche Eingabe machen wir unten in dem langen Eingabe-Feld. Ob man die Eingabe mit dem Häkchen oder mit der [Enter]-Taste beendet ist Geschmackssache.

Ein eingegebener Wert erschein dann links oben in der Variable Antwort.

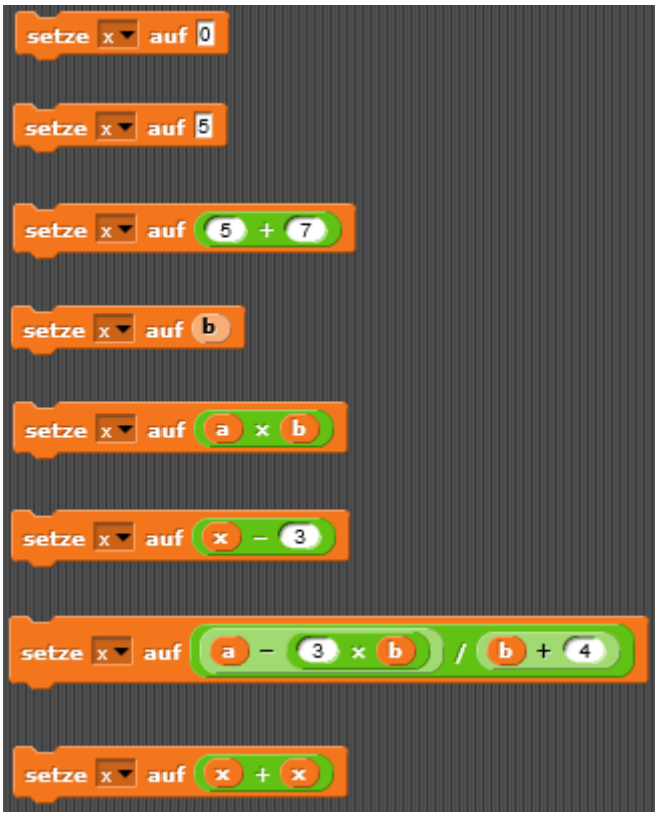
Genau diese Variable können wir nun als Argument in unserem Programm benutzen.

Diesen Antwort-Ausdruck schieben wir nun in den Festlegungs-Block für b.
Genau so verfahren wir dann auch für die Höhe h.
An diesem – nun wirklich fertigen Programm – kann kein Informatiker mehr was aussetzen.

Operationen mit Variablen und Zahlen

Variablen lassen sich nun lesen und schreiben. Für viele Programme braucht man berechnete Werte. Variablen lassen praktisch wie in der Mathematik benutzen. Die Argument-Schlitze in den Blöcken deuten es teilweise auch schon an.

In Snap! stehen uns die üblichen mathematischen Operatoren, wie **+** und **–** zur Verfügung. Die meisten Opratoren brauchen zwei Argumente (Operanden). Die verschiedenen Opratoren finden wir in der Operatoren-Kategorie. Die Operatoren-Blöcke passen in die Argument-Positionen.
Nebenstehend sind eine prinzipielle Mglichkeiten aufgezeigt. Dabei gehen wir immer davon aus, dass die Variable x mit einem neuen Wert belegt werden soll.
Es lassen sich also Werte direkt abspeichern (obere zwei Beispiele).
Man kann Zahelen / Werte mittels Operatoren verknüpfen oder andere Variablen nutzen. Diese werden zur weiteren Verarbeitung entsprechend ausgelesen.
Jedwege Kombination ist möglich. Selbst mehrfach geschachtelte Operationen lassen sich mit den Blöcken zusammenstellen. Besonders interessant ist das letzte Beispiel.
Solche Zuweisungen sind mglich, weil zuerst die rechte Seite – also x+x berechnet wird.



Dazu wird die Variable x ausgelesen und verarbeitet. Erst wenn die rechte Seite vollständig berechnet und nur noch ein Zuweisungs-Wert übrig ist, wird dieser in die Variable x gespeichert. Der alte x-Wert geht dabei verloren.

Aufgaben:

1. *Überlege Dir welchen Wert die benutzen Variablen nach jedem Schritt a bis haben!*

a)	a = 3; b = 4; x = 0	b)	x = a + 4	c)	x = b + a
d)	a = x	e)	b = x + a	f)	x = 3 * x + b
g)	x = x + x	h)	x = x + x + x	i)	x = x * 2 + x

2. *Erkunde wie die Operation "x modulo y" funktioniert! Gib dann die Lösung für die folgenden Aufgaben an! (mod ist die Kurzform von modulo)*

a)	4 mod 2	b)	5 mod 2	c)	6 mod 4
d)	6 mod 6	e)	4 mod 8	f)	24 mod 13

3.

Was passiert beim Lesen und Schreiben von Variablen? Wo stecken die genau im Computer?

Arbeiten mit Zeichenketten

Kurz gesagt sind Zeichen-Ketten mehr oder weniger lange Texte. In Computersystemen sind alle Buchstaben in Groß- und Klein-Schreibung, aber auch Ziffern und Satzzeichen zugelassen. Häufig dürfen auch Zeichen benutzt werden, die nicht so einfach über die Tasttatur zugänglich. Einige dieser Zeichen schauen wir uns in einem Exkurs an (→).
Zur Kennzeichnung von Zeichenketten werden üblicherweise Anführungszeichen benutzt. In Snap! ist dies nicht notwendig. Die kürzeste mögliche Zeichenkette ist der leere String. So nennen Programmierer die Zeichenketten. Die Darstellung würde dann so aussehen "". Enthält die Zeichenkette irgendein Zeichen, dann ist sie automatisch größer als die leere Zeichenkette. Die Länge von Zeichenketten ist ein häufig gebrauchtes Maß. Praktisch jede Programmiersprache bietet zur Bestimmung der Länge einer Zeichenkette eine Funktion. In Snap! heißt sie: "Länge von (Welt)" und ist in der Kategorie "Operatoren" zu finden. "Welt" kann natürlich durch jede beliebige Zeichenkette oder eine Variable – die eine Zeichenkette enthält – ersetzt werden.

Das Arbeiten mit Zeichenketten, insbesondere das Zerlegen und / oder Zusammensetzen von Zeichenketten ist eines der Hauptgeschäfte von Programmierern.

ab hier eher Klasse 7

Arbeiten mit Listen

Listen
Erstellen einer Liste

leere Liste

Hinzufügen von Elementen zu Listen
einspeichern

add()

Index-Zugriff

interaktiver Modus

Beispiel-Aufgabe
Sortieren einer Liste von Zufalls-Zahlen nach dem Prinzip der Auswahl der kleinsten Elemente aus der Original-Liste und Anhängen in eine neu zusammenzustellende Liste der sortierten Elemente

Löschen einer Liste

Definition neuer / eigener Blöcke

Bestimmte Sequenzen von Blöcken brauchen wir immer wieder. Man kann Gruppen von Blöcken als eigene Befehls-Blöcke abspeichern. Das spart viel Platz auf der Programmier-Fläche, da ja aus vielleicht fünf Blöcken jetzt nur noch einer im Programm steht. Praktisch entspricht das dem Erstellen von Unterprogrammen.
Unterprogramme machen z.B. dann Sinn, wenn man zwar gleiche Block-Sequenzen braucht, diese aber nicht direkt wiederholen kann.

Erstellen einer eigenen Block-Sequenz

Definition eines neuen Blockes über ""-
Schaltfläche aus der Kategorie "Variablen"

neuer Block (Snap!-Bausteine; engl.:)

neue Funktion (Argumente, Funktionen (Blöcke mit runden Seiten); engl.: Reporter)

neues Prädikat (logische Funktion; Wahr-Falsch-Aussage; engl.:)

6.x.1.3. Programmieren mit NEPO (open Roberta)

NEPO ist die Programmiersprache

6.x.1.3.0. Wege zu "Roberta"

online verfügbar
derzeit nur für die online-Benutzung gedacht
universelle Programmier-Oberfläche für verschiedene Hardware:

- Roboter
  - Lego NXT
  - Lego EV3
  - Bot'n Roll
  -
- Experimentier-Boards / Einplatinen-Systeme / …
  - Calliope mini (Calliope)
  - micro:bit (microbit)
  - BOB3
  - NAO
  -

hier immer aktuell

Wiki zum ersten Ansehen (→https://jira.iais.fraunhofer.de/wiki/display/ORInfo)

Open Roberta Lab – online
online: https://lab.open-roberta.org/

Vorteile
praktisch immer verfügbar, wenn man eine Internet-Verbindung hat
Plattform-unabhängig (ev. begrenzt aber die Verfügbarkeit von Treibern für die Hardware (Roboter, ExperimentierBoards, …) die Nutzbarkeit)
immer aktuell
keine Installation notwendig

Nachteile
Abhängigkeit vom Internet
bei Volumen- oder Zeit-gebundenen Internet-Verbindungen nicht günstig

NEPO offline
derzeit nur über einen RaspberryPi möglich
ein passendes Image unter https://www.roberta-home.de/lab/ verfügbar (2,5 GB)

Vorteile
unabhängig vom Internet
moderne und relativ einfache Informatik in Reinkultur

mit wenig finanziellen Aufwand realisierbar (RaspPi's lassen sich auch als Desktop-Rechner für Standard-Aufgaben nutzen)

Nachteile
man braucht einen Raspberry Pi (ev. Extra-Kosten)
RaspberryPi's müssen eingerichtet werden (nicht jeder Administrator mag Linux)
relativ Leistungs-schwaches System (nur als Alternative / Zweit-System zu empfehlen)

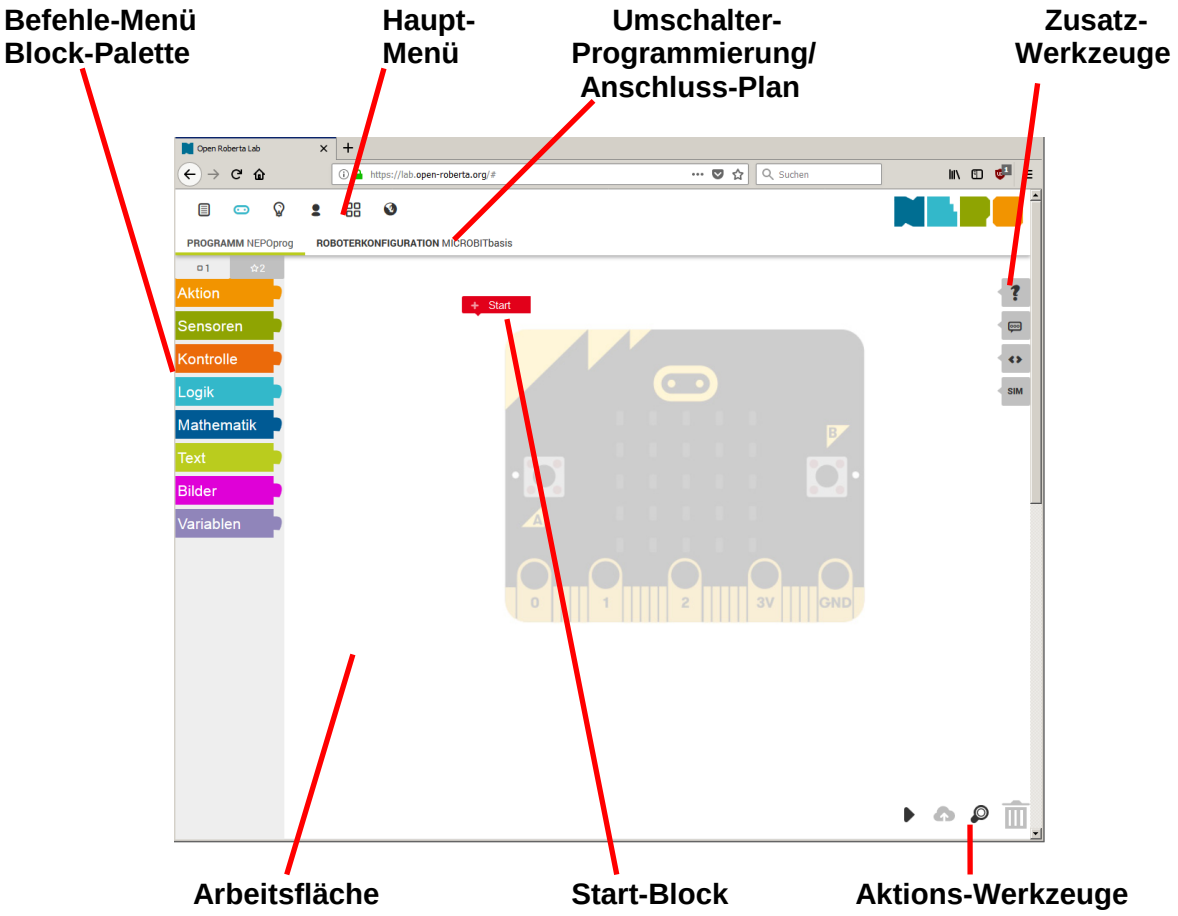
Tipp für Besessene / echt Interessierte:
Wenn Dir das Programmieren wirklich Spass macht und Du gar nicht aufhören möchtest, Du noch hunderte von Ideen hast und sowieso noch einen Wunsch zu Weihnachten oder zum Geburtstag frei hast, dann wünsch Dir doch mal so einen Raspberry Pi.
Zeig einfach mal Deine ersten Skripte und erzähle dem "Weihnachtsmann" oder Deinen Eltern, was Dich am Programmieren und am Rasp Pi so begeistert. Du wirst sehen, das wirkt Wunder.
Als Geschenk von einer anderen Partei (Großeltern, …) ist vielleicht ein kleiner Roboter drin. Vielleicht können sich mehrere Schenker zusammenschließen und sogar einen LEGO®-Robotor-Bausatz schenken? Die verschiedenen Roboter oder Hardware-Platinen braucht man keinen Rasp Pi. Es kann jeder Computer mit Internet-Browser, Internet-Zugang und USB-Anschluss genutzt werden.

Block-Gruppen	Farbe	englische Bezeichnung	enthält z.B. …	
Aktion		Action		
Sensoren		Sensors		
Kontrolle		Control		
Logik		Logic		
Mathematik		Math		
Text		Text		
Farben		Coulors		
Bilder		Images		
Variablen		Variables		
zusätzliche Blöcke / Erweiterung				
Listen		Lists		
Funktionen		Functions		
Nachrichten		Messages		

Scratch-Sprache ändern:

- auf Globus-Symbol klicken
- Sprache auswählen

online: https://lab.open-roberta.org/



Programm / Skript speichern (online-Version) / herunterladen:

- bei "Datei" "Herunterladen auf deinen Computer" anklicken
- Speicher-Ort auswählen
- Datei-Namen festlegen
- "Speichern"

***Programm / Skript speichern** (offline-Version):*

- auf Disketten-Symbol klicken
- Speicher-Ort auswählen
- Datei-Namen festlegen
  - "Projektautor" und "Über dieses Projekt:" (Projekt-Beschreibung) sind optional
- "OK"

Projekt / Skript öffnen (online-Version) / hochladen:

- bei "Datei" "Hochladen von deinem Computer" Speicher-Ort auswählen
- Datei aus Liste auswählen
- "Öffnen"

***Projekt / Skript öffnen** (offline-Version):*

- bei "Datei" "Öffnen …" Speicher-Ort auswählen
- Datei aus Liste auswählen
- "OK"

open-Roberta erkunden:

Welche Möglichkeiten verstecken sich hinter den Blöcken?
Wie kann man einen Block anhängen, einbauen, verschieben oder löschen?

- auf die Blöcke in der Block-Palette klicken → es öffnet sich eine Auswahl
- Blöcke mit der Maus bewegen

Programm Start-fähig machen / Start-Symbol einfügen:

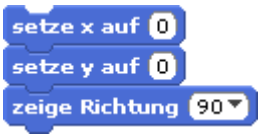
- "Start"-Block wird immer automatisch angezeigt (ein Löschen ist nicht möglich; es gibt nur einen Programm-Faden)

Objekt bewegen:

- oben in der Block-Palette "Bewegung" auswählen
- passenden Bewegungs-Block anklicken und in den Programmier-Bereich ziehen
  - ev. gleich mit anderen Blöcken verbinden / an die richtige Position ziehen
- Wert für Schritte, Drehwinkel oder Positionen eingeben (weiße Felder)

Ausgangs-Situation wieder herstellen / zurücksetzen:

- oben in der Block-Palette "Bewegung" auswählen
- Block "setze x auf 0" als ersten Block nach dem "Grüne Fähnchen"-Block
- danach Block "setze y auf 0" einfügen
- Richtung des Objektes mit Block "zeige Richtung 90" festlegen



eigenes Programm / Skript starten / ablaufen lassen:

- das grüne Fähnchen rechts oben über der Bühne anklicken

eigenes Programm / Skript in Einzelschritten laufen lassen (Block für Block):

- im Programm-Menü "Bearbeiten" "Ausführung in Einzelschritten"
  - ev. "Einzelschritte einstellen …" z.B. auf "Bausteine hervorheben (langsam)"
- dann zum Starten das grüne Fähnchen rechts oben über der Bühne anklicken

eigenes Programm / Skript beenden / unterbrechen:

- auf den roten Kreis rechts oben über der Bühne klicken

ODER

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- einen der Blöcke "stoppe dieses Skript" (gilt nur für ein Objekt) ODER "stoppe alles " (gilt für alle Objekte / das Gesamt-Programm) in das Skript einbauen

Aufgaben:

1. Erstellen Sie ein neues Programm (Skript) mit dem Datei-Namen "FeldAblaufen" in Ihrem eigenen Ordner!

2. Kombinieren Sie jetzt die verschiedenen Blöcke so, dass die Katze (/das Objekt) einmal am Rand um das gesamte Feld (/ die Bühne) läuft!

3. Speichern Sie das fertige Programm am Ende ab!

für Experten und zum Knobeln:

4. Lassen Sie das Objekt auch noch die beide Diagonalen der Bühne ablaufen!

einen Klang / Sound abspielen:

- oben in der Block-Palette "Klang" auswählen
- passenden Klang-Block auswählen und zur Abspiel-Position in den Block-Stapel ziehen
  - ev. Klang ändern
  - ev. Geschwindigkeit anpassen (extra Block vor den Abspiel-Block einfügen)
  - ev. Lautstärke anpassen (extra Block vor den Abspiel-Block einfügen)
  - ev. als einen de letzten Blöcke "stoppe alle Klänge" einbauen

(c,p) 2015 – 2020 lsp: dre

Aufgaben:

1. *Speichern Sie das fertige Programm (Skript) vom vorherigen Aufgaben-Block für verschiedene Erweiterungen unter einem neuen Datei-Namen "FeldAblaufenExtra" in Ihrem eigenen Ordner!*
2. *Das Objekt (/ die Katze) soll sich in den Bühnen-Ecken langsam einmal um mindestens 360° drehen (Richtung ist egal)! Dabei soll das Objekt mindestens einmal einen Klang abspielen (z.B. "Miau")!*
3. *Lassen Sie am Ende des Feldablaufes einen Schlagzeug-Wirbel spielen!*

für Experten und zum Knobeln:

4. *Das Objekt (die Katze) soll beim Drehen (Aufgabe 2) zwischen den kleinen Dreh-Bewegungen die Klänge von laut nach leise abspielen!*

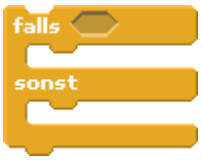
bedingte Ausführung von Programm-Teilen:

- oben in der Block-Palette "Steuerung" auswählen
- den "falls"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
- Bedingung (aus Bereich "Operatoren" oder "Fühlen") einfügen
  - ev. die bedingt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Bedingung festlegen:

- vorher einen geeigneten "Steuerung"s-Block (mit Waben-förmiger Bedingungs-Stelle) in das Skript einbauen
- oben in der Block-Palette "Operatoren" auswählen
- einen Waben-förmigen Bedingungs-Block auswählen
- variable Felder ausfüllen
  - ev. Bedingungen durch logische Verknüpfungen kombinieren

alternative Wege erstellen:



- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "falls .. sonst"-Doppel-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
- Bedingung (aus Bereich "Operatoren" oder "Fühlen") einfügen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die obere Klammer ziehen
  - die alternativ / sonst auszuführenden Blöcke / Programm-Teile in die untere Klammer ziehen

Programm-Abschnitt eine bestimmte Anzahl oft wiederholen:

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "wiederhole … mal"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
  - die abgezählt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

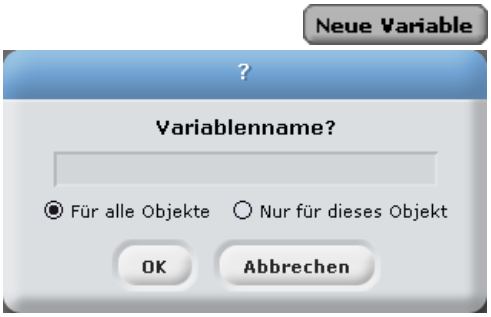
Programm-Abschnitte fortlaufend / immerzu wiederholen:

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "wiederhole fortlaufend"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Programm-Abschnitte bedingt fortlaufend wiederholen:

- oben in der Block-Palette "Steuerung" bzw. "Events" auswählen
- den "wiederhole fortlaufend, falls …"-Klammer-Block an die geeignete Ausführungs-Stelle ziehen
- Bedingung (aus Bereich "Operatoren" oder "Fühlen") einfügen
  - die bedingt auszuführenden Blöcke / Programm-Teile in die Klammer ziehen

Variable definieren (bekanntgeben):



- oben in der Block-Palette "Variablen" auswählen
- die "Neue Variable"-Schaltfläche klicken und im nachfolgenden Dialog den Namen der Variable eingeben

Variable verwenden:

- oben in der Block-Palette "Variablen" auswählen
- die Variable (hier z.B.: laenge) auswählen und in einen geeigneten Block hineinziehen

neues Objekt einfügen:

- in der Kopfzeile der Objekt-Liste "Neues Objekt aus Datei laden" auswählen
- in einen passenden Ordner wechseln
- Objekt auswählen

Achtung! *Jedes Objekt muss einzeln programmiert werden!*

Aufgaben:

1. *Erstellen Sie ein neues Programm, in dem eine Maus (als Objekt) durch Drücken der Pfeil-Tasten gesteuert wird!*
2. *Veränderen Sie das Programm soweit, dass beim Erreichen eines Randes ein Klang ausgegeben wird!*

für Experten und zum Knobeln:

3. *Steuern Sie die Hoch- und Runter-Bewegung des Objekt's durch die Umgebungslautstärke! (Mikrofon bzw. Head-Set notwendig!)*

(c,p) 2015 – 2020 lsp: dre

Objekt-Größe ändern:

- Objekt auf der Bühne anklickenund mit rechter Maus-Taste Kontext-Menü aufrufen
- "Die Größe dieses Objektes ändern" auswählen und am Objekt mittels Skalierer die Ausdehnung kleiner oder größer ziehen

Objekt-Spur zeichnen (beginnen):

- oben in der Block-Palette "Malstift" auswählen
- Block "senke Stift ab" an Start-Position (im Skript) für Zeichnen setzen
  - ev. Farbe setzen
  - ev. Stiftdicke setzen

Objekt-Spur zeichnen beenden:

- oben in der Block-Palette "Malstift" auswählen
- Block "hebe Stift an" an Ende-Position (im Skript) setzen

Berührung von Objekten testen:

- vorher einen geeigneten "Steuerung"s-Block in das Skript einbauen
- oben in der Block-Palette "Fühlen" auswählen
- passende Berührungs-Bedingung in den Steuerungs-Block hineinziehen

Zufallszahlen verwenden:

- vorher einen geeigneten Block mit einem variablen Bereich (rundlich, weiß) in das Skript einbauen
- oben in der Block-Palette "Operatoren" auswählen
- Block "Zufallszahl von … bis …" an in den Block mit variablen Bereich hineinziehen

Aufgaben:

1. *Erstellen Sie ein Programm, bei dem sich die Maus immerzu zwischen den Rändern hin und her und auch hoch und runter bewegt!*
2. *Speichern Sie das Programm von Aufgabe 1 nochmals unter einem neuen Namen ab! Erweitern Sie das Programm um eine Katze, die durch Tasten-Steuerung die Maus fangen soll! Immer, wenn die Maus einmal erwischt wurde (Berührung), soll ein Zähler hochzählen und die Maus zufällig an eine neue Position springen!*
3. *Öffnen Sie sich das Programm von Aufgabe 1 und speichern Sie es wiederum unter einem neuen Namen ab! Verändern Sie das Programm nun so, dass die Maus eine Spur hinterlässt!*

für Experten und zum Knobeln:

4. *Legen Sie auf der Bühne einige Hindernisse an! Diese müssen nun von den Figuren auch umgangen werden.*

6.x.1.3.1.fortgeschrittenes Programmieren mit NEPO

Das Austauschen von Nachrichten dient der Kommunikation von Skripten / Objekten untereinander. Damit kann einem anderen Objekt etwas mitgeteilt werden oder ein Skript kann auf eine spezielle Nachricht eines anderen Skriptes / Objektes reagieren.

senden einer Nachricht:

- oben in der Block-Palette "Ereignisse" bzw. "Events" auswählen
-
  -
  -

empfangen von Nachrichten:

-
-
  -
  -

senden einer Nachricht und warten (*z.B. auf eine Reaktion / Antwort****):***

-
-
  -
  -

6.x.1.4. Programmieren des Calliope mini mit MakeCode

Programmieren des Calliope mini mit microsoft MakeCode (ehemals PXT)

http://calliopemini.info/
https://makecode.calliope.cc/ (Programmier-Tool MakeCode zum Calliope)

(http://pxt.calliope.cc/index.html)

Block-Gruppen	Farbe	englische Bezeichnung	enthält z.B. …	
Aktion		Action		
Sensoren		Sensors		
Kontrolle		Control		
Logik		Logic		
Mathematik		Math		
Text		Text		
Farben		Coulors		
Bilder		Images		
Variablen		Variables		
zusätzliche Blöcke / Erweiterung				
Listen		Lists		
Funktionen		Functions		
Nachrichten		Messages		

Aufgaben:

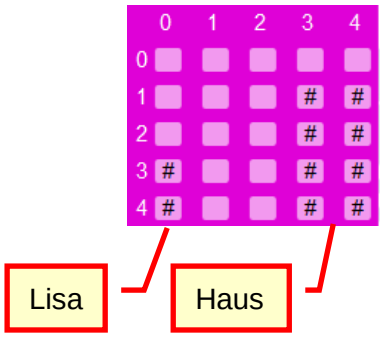
1.
2. *Simuliere eine Fußgänger-Ampel-Schaltung! Überlege Dir zuerst, was simuliert werden kann und muss! Einigt Euch in der Klasse auf einen Ablauf! Erstelle und prüfe dann Dein Simulations-Programm!*
3. *Erstelle die passende Auto-Ampel-Schaltung zur Fußgänger-Ampel aus Aufgabe 2!*
4. *Lasst je einen Calliope mit einem Programm von Aufgabe 2 und 3 parallel laufen! Drückt dann gemeinsam die Anforderung für die Fußgänger-Ampel! Welche Probleme könnten auftreten? Verbessert Eure Programme entsprechend!*

Aufgaben:

1. *Lisa geht zur Schule. Plötzlich geht an einem Haus in der Straße der Feueralarm. Der Dauerton (Tonhöhe) ist sehr laut. Nach 3 Sekunden geht der Alarm aus. Scheinbar war es nur ein Fehlalarm. Da die Feuerwache aber trotzdem informiert wurde, ist das Feuerwehr auto schon losgefahren. Lisa hört, wie sich ein Feuerwehr-Auto mit seiner Sirene (3x Ton h und Ton e im Wechsel) nähert. Erst leise, dann immer lauter. Nun sieht sie auch das blaue Blinklicht. Das Feuerwehrauto hält an dem Haus mit dem Feueralarm. Der Fahrer schaltet die Sirene aus. Nachdem die Feuerwehrleute sich überzeugt haben, dass es doch nicht brennt, wird auch das Blaulicht abgeschaltet.*
   a) *Erstelle einen Ablaufplan für diese Szene! (z.B. aus kleinen Papier-Kärtchen mit ganz kurzen Szenen-Namen!)*
   b) *Vergleiche Deinen Ablaufplan mit dem eines Mitschülers! Findet Fehler und berichtigt diese!*
   c) *Simuliere die ganze Geschichte auf dem Calliope! Fange zuerst mit Feueralarm an dem Haus an! Teste das Teil-Programm und korrigiere eventuelle Fehler! Füge dann die nächste Teil-Szene hinzu!*
   d) *Vervollständige dann die Szenen, bis die gsammte Geschichte auf dem Calliope simuliert wird!*

für Experten und zum Knobeln:

2. *Stelle das Haus und Lisa ganz vereinfacht auf der LED-Matrix dar und lass Lisa bis zum Haus gehen! Der Feuer-Alarm könnte durch eine extra leuchtende LED auf dem Haus angezeigt werden.*



Links:
http://calliope-saarland.de/

(c,p) 2015 – 2020 lsp: dre

6.x.1.5. Programmieren von Arduino's und kompatiblen / abgeleiteten IoT-Bausteine

6.x.1.5.0. Wege zur Programmier-Umgebungen

nur online verfügbar

Open Roberta Lab – online
online: https://lab.open-roberta.org/

Vorteile
praktisch immer verfügbar, wenn man eine Internet-Verbindung hat
Plattform-unabhängig (ev. begrenzt aber die Verfügbarkeit von Treibern für die Hardware (Roboter, ExperimentierBoards, …) die Nutzbarkeit)
immer aktuell
keine Installation notwendig

Nachteile
Abhängigkeit vom Internet
bei Volumen- oder Zeit-gebundenen Internet-Verbindungen nicht günstig

NEPO offline
derzeit nur über einen RaspberryPi möglich
ein passendes Image unter https://www.roberta-home.de/lab/ verfügbar (2,5 GB)

Vorteile
unabhängig vom Internet
moderne und relativ einfache Informatik in Reinkultur
mit wenig finanziellen Aufwand realisierbar (RaspPi's lassen sich auch als Desktop-Rechner für Standard-Aufgaben nutzen)

Nachteile

zwar noch frühe Versionen, aber für viele Umgebungen verfügbar (offline für diverse Betriebssysteme)
abgewandeltes Scratch
auch online- und android-Version

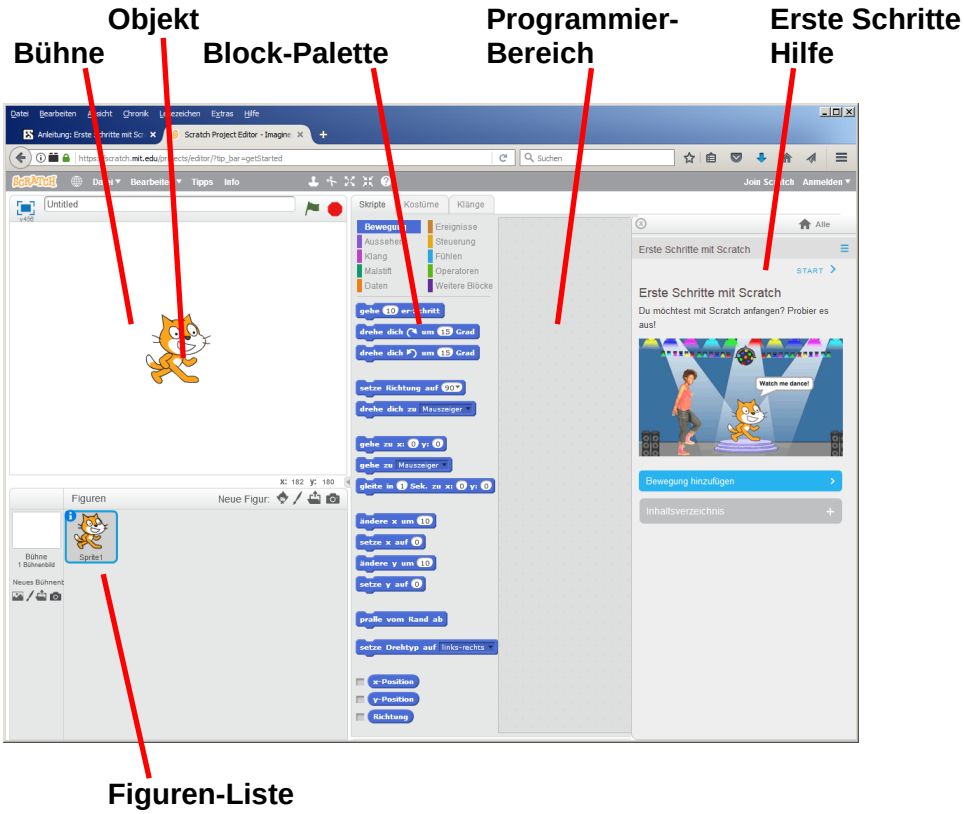
Block-Gruppen	Farbe	englische Bezeichnung	enthält z.B. …	
Bewegung		Motion		
Aussehen		Looks		
Klang		Sound		

Malstift		Pen		
Daten		Data		
Ereignisse		Events		
Steuerung		Control		
Fühlen		Sensing		
Operatoren		Operators		
Arduino		Arduino		

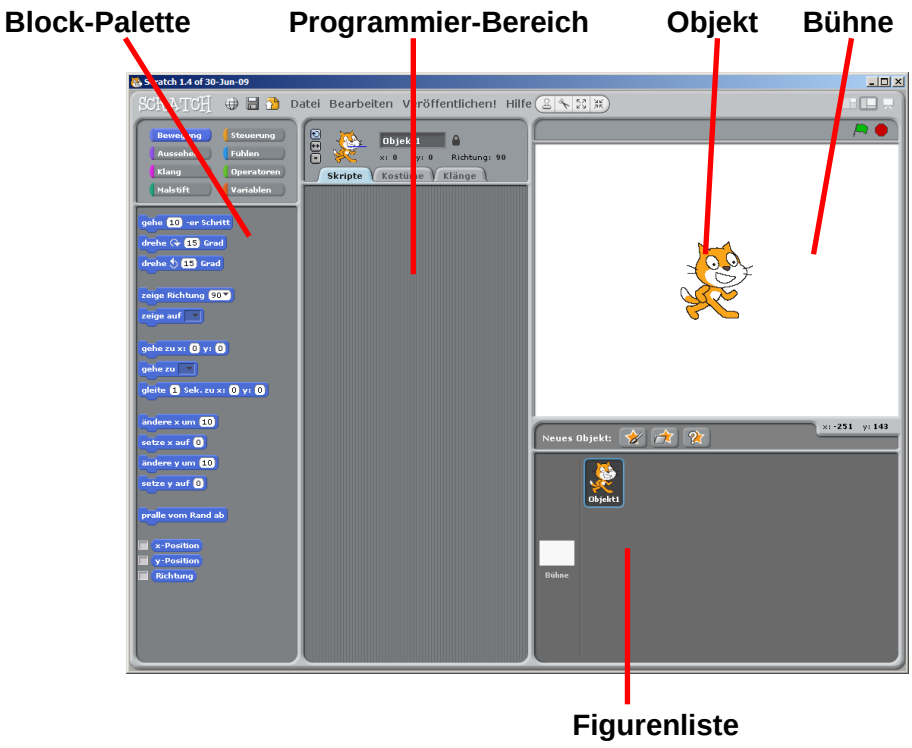
Scratch-Sprache ändern:

- auf Zahnrad-Symbol klicken
- Sprache (Language …) auswählen

online: https://scratch.mit.edu



Für die offline-Version von Scratch – z.B. vom IoStick – ist nur die Verteilung der Bereiche anders. Es fehlt aber auch der Hilfe- / Erste-Schritte-Bereich.



Programm / Skript speichern** (offline-Version)**:

- auf Disketten-Symbol klicken
- Speicher-Ort auswählen
- Datei-Namen festlegen
  - "Projektautor" und "Über dieses Projekt:" (Projekt-Beschreibung) sind optional
- "OK"

Programm / Skript speichern (online-Version) / herunterladen:

- bei "Datei" "Herunterladen auf deinen Computer" anklicken
- Speicher-Ort auswählen
- Datei-Namen festlegen
- "Speichern"

Projekt / Skript öffnen** (offline-Version)**:

- bei "Datei" "Öffnen …" Speicher-Ort auswählen
- Datei aus Liste auswählen
- "OK"

Projekt / Skript öffnen (online-Version) / hochladen:

- bei "Datei" "Hochladen von deinem Computer" Speicher-Ort auswählen
- Datei aus Liste auswählen
- "Öffnen"

ansonsten wie bei der Scratch-Programmierung (s.a. weiter vorn (→))

!!!ergänzen Ansteuerung der Pin's

vorher muss eine spezielle Verbindungs-Software installiert werden
in Arduino-Programmier-Umgebung → Datei Beispiele Firmata StandardFirmata
damit reagiert das Board quasi online auf in der Programmier-Umgebung gestartete Programme
nach dem Beenden der Programmier-Umgebung ist auch das Programm nicht mehr auf dem Board aktiv
?? wie geht kompilieren

Links:
http://snap4arduino.rocks/
http://s4a.cat/snap

5. In der vernetzten Welt kommunizieren

mithilfe eines Internetdienstes kommunizieren

Ein geeigneter Dienst ist die E-Mail.

- *Nachrichten unter Angabe eines geeigneten Betreffs an einen oder mehrere Empfänger versenden*

Neuerstellung, Beantwortung und Weiterleitung sind Möglichkeiten des versendens von Nachrichten.

- *Anhänge hinzufügen und speichern*

Die SuS können Dateigröße in der Maßeinheit Byte mit Präfixen und Dateityp angeben.

- *Authentizität und Gefahrenpotential von eingehenden Nachrichten abschätzen*

Die Beurteilung erfolgt anhand der Absenderadresse, des Betreffs oder des Inhalts. Die SuS reagieren in geeigneter Weise.

- *mit personenbezogenen Daten verantwortungsbewusst umgehen*

Die SuS beachten sowohl den Schutz der eigenen Daten als auch den Schutz von Daten anderer Personen.

Aufbau und Wirkungsweise des Internets modellhaft beschreiben

- *das Internet mit Hilfe eines Zwei-Schichten-Modells beschreiben*

Die SuS unterscheiden die Schicht der technischen Infrastruktur und die Schicht der Internetdienste, von denen sie das World Wide Web (WWW) und den E-Mail-Dienst prototypisch angeben können.

- *die Client-Server-Kommunikationsstruktur erläutern*

Die Erläuterung kann anhand des WWW-Dienstes erfolgen.

- *die Aufgabe eines Protokolls im Kommunikationsprozess beschreiben*

Ein Protokoll ist ein Regelwerk für die Kommunikation. Die Herstellung von Alltagsbezügen, z.B. zu Verhaltensregeln im Unterricht oder zum diplomatischen Protokoll, bieten sich an.

Problem-Fragen für Selbstorganisiertes Lernen

Wie funktioniert das Internet? Wie kommen die Daten von einem Ort zum anderen?

Wie finden die Daten ihren Weg durchs Internet?

Was ist das Zwei-Schichten-Modell? Gibt es andere Schicht-Modell in der Informatik?

Wie funktioniert der eMail-Dienst im Internet?

Was ist ein Protokoll? Ist die Einhaltung bzw. Beachtung von Protokollen wirklich wichtig? Was machen Computer, wenn Protokolle verletzt werden?

Wie hat man es hin bekommen, das die Internet-Seiten auf allen möglichen Geräten und Bildschirmgrößen ungefähr ähnlich angezeigt werden?

Kommen immer alle Nachrichten an?

Problem-Fragen für Selbstorganisiertes Lernen
Muss man über alle Medien Bescheid wissen? Ich nutze nur mein Smartphone als Medium? Reicht das? Wie mache ich das richtig? Ist das Smartphone das einzige Medium der modernen Jugendlichen? Gibt es einen Jugendschutz in der Medienwelt?

5.x. ???

Einstieg für höhere Klassenstufe / anderes Thema!!! →

"Medienkompetenz ist das beste Mittel gegen Online-Sucht" titelte die Ostsee-Zeitung in ihrer Ausgabe vom 26./27. November und zeigte dazu ein Bild mit einer an sein Smartphon gefesselte Person (exakt: deren Hände).

Aber der Weg zur Medien-Kompetenz ist schwierig
er setzt Kenntnisse über Medien voraus
Welche Medien gibt es? Wie werden sie erstellt? Welche Garantie für realistische Darstellungen eines Thema's gibt es? Wie kann man Medien sinnvoll und passgerecht benutzen?

noch (2016) ist die Online-Spielsucht keine anerkannte Krankheit
soll aber bald durch die WHO in die Internationale Klassifikation ausgenommen werden

Aufbau und Arbeitsweise des Internets modellhaft beschreiben

E-Mails versenden

E-Mails mit Anhängen versenden

E-Mails abrufen

mit personenbezogenen Daten verantwortungsbewußt umgehen

die Client-Server-Kommunikationsstruktur erläutern

die Notwendigkeit von Protokollen für den Kommunikationsprozess begründen

Idee (y):
Programmierung des Calliope oder eines anderen WLAN- bzw. Bluetooth-fähigen Baustein-Rechner als Mikro-Internet.

Rechner vereinbaren einen Kanal
die einzelnen Rechner werden unterteilt in End-Stationen und Router
Rechner sind so verteilt, dass sich immer nur wenige Rechner empfangen können (Abstand muss ausprobiert werden
die Router empfangen eine Nachricht und senden sie wieder (ev. Anzeige der Adresse)
End-Geräte haben eine Adresse
Var1: ist die Nachricht die eigene Adresse dann werden ein paar Töne abgespielt und die Adresse angezeigt
Var2: enthält die Nachricht zu Anfang deren Adresse, zeigen sie den Rest der Nachricht an

für Calliope z.B. die folgendenden Grund-Funktionen möglich:

beim Start – setze Gruppe X

wenn Knopf A+B gedrückt sind – sende *Nachricht*

wenn Datenpaket empfangen Nachricht – zeige *Nachricht* – spiele Note für ein Takt

soziale Netzwerke

unzugängliches Internet

Nicht alles im versteckten Internet ist kriminell. Viele Nutzer möchten sich einfach frei im Internet bewegen. Dazu wurde das Internet ja auch ursprünglich konzipiert.
Leider nutzen kriminelle Personen und Organisationen die Anonymität dieses Teils des Internet's für ihre Geschäfte aus.

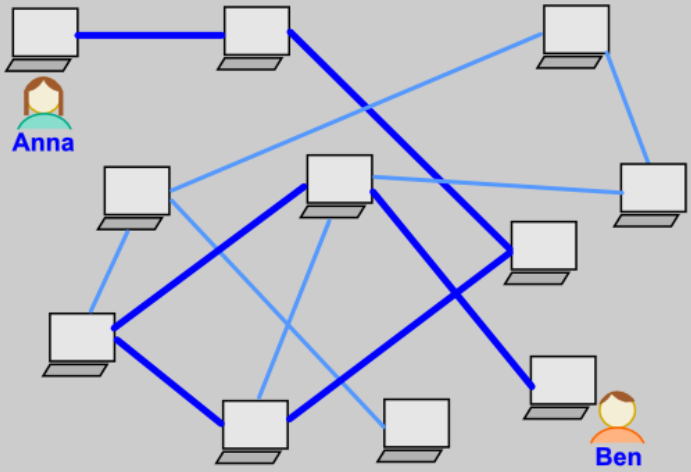
TOR-Browser

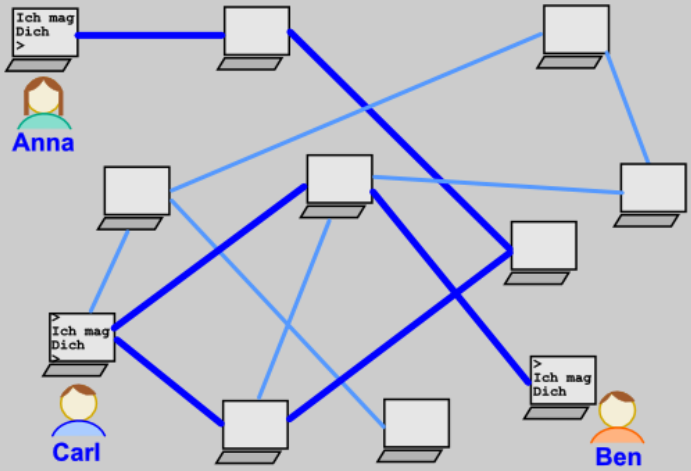
offizielles Logo
Q: de.wikipedia.org
(The Tor Project Inc.)

Der TOR-Browser ist ein Internet-Programm, das für anonymes Surfen konzipert wurde und sich auch für das reguläre Internet eignet. Es geht nur um die Verschleierung der Dtenspuren, nicht um eine kriminelle Nutzung.
Der Browser selbst speichert praktisch fast keine Daten ab. Die wenigen Daten die aus irgendwelchen Gründen unbedingt gespeichert werden müssen, werden nach der Nutzung des TOR-Browsers wieder gelöscht.
Der TOR-Browser braucht nicht installiert werden und kann auch z.B. von einem USB-Stick als "portable App" laufen. Damit lässt es sich fast auf allen Windows-PC's einsetzen.
Der Tor-Browser nutzt für die Aufrufe ins Internet ein spezielles Netzwerk von Rechnern in verschiedenen Ländern. Einfach gesagt, werden die Anfragen von einem Gerät über viele Zwischenstationen in verschiedenen Ländern verteilt und immer jeweils die Herkunft verschleiert. Die Zwischenstationen wissen nur, von welcher Vorgängerstation die Anfrage kam, nicht von wem ursprünglich. Damit können die Ergebnisse auch wieder zum anfragenden Gerät zurückgeschickt werden. Die Aufklärung, wer welche Information abgerufen hat wird aber extrem aufwändig – aber nicht unmöglich! Kriminelles Handeln im Darknet ist also auch aufklärbar. Und es gibt auch Tricks das TOR-Netzwerk auszutricksen! In bestimmten Fällen soll das durch Ermittlungs-Organe (Polizei, Geheimdienst auch schon geschehen sein.
Die Zwiebel im Projekt-Logo von TOR ist sinnbildlich für das Netzwerk- und Arbeits-Prinzip gewählt worden. Jede einzelne Schicht steht für eine Zwischenstation, die jeweils nur mit der nächst-inneren und –äußeren Schicht kommuniziert. TOR war ursprünglich die Abkürzung für "The Onion Routing" (Weiterleitung nach dem Zwiebel-Prinzip) – heute ist es mehr ein Eigenname und wird auch meist Tor geschrieben.
Der Tor-Browser lässt sich unter https://www.torproject.org/ herunterladen. Der Download muss dann entpackt werden und kann dann auch sofort genutzt werden. Eine spezielle Installation ist nicht notwendig. Wer will, kann den Ordner, in dem der Tor-Browser entpackt wurde, am Ende einfach gelöscht werden. Nach einem endgültigen Löschen aus dem Papierkorb sind dann auch wirklich alle Spuren von der Tor-Nutzung verschwunden.
Mit dem Tor-Netzwerk is das Aushebeln der Länder-Beschränkungen einiger Webseiten möglich, weil das Tor-Netzwerk diesen Webseiten einen ganz anderen – ausländischen – Rechner (Nutzer) vorgaukelt.

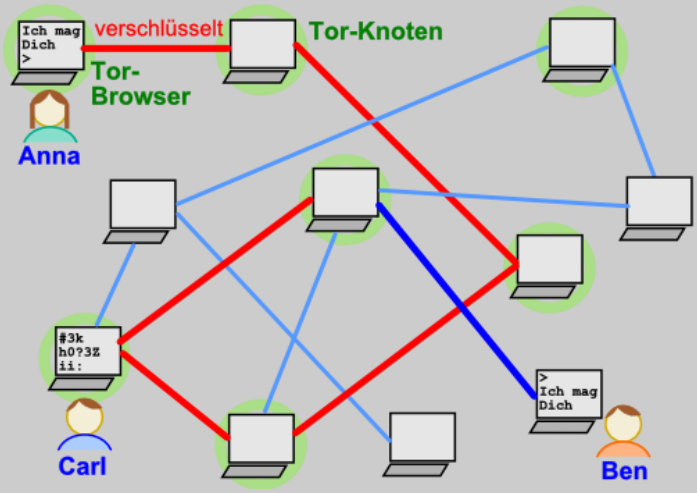
Exkurs: Wie arbeitet das TOR-Netzwerk?

Im Internet sind die Computer über mehrere Zwischenstationen (Server) mehr oder weniger zufällig miteinander verbunden.
Viele Interrnet-Services – wie z.B. eMail oder http (Webseiten) – arbeiten aus historischen Gründen unverschlüsselt.
Wenn Anna dem Ben eine Nachricht schickt, dann werden diese Daten in kleinen Paketen (von Nullen und Einsen) über die Leitungen transportiert.
Neugierige Personen und Institutionen können diese Daten meist ohne größeren Aufwand mitlesen.
In unserem Beispiel ist Carl die neugierige Person.
Er ist schon lange neidisch auf Ben.
Da die Nachrichten unverschlüsselt über das Netz gehen, kann er mit ein bisschen technischem Wissen seinen Computer zum Mitlesen bringen. Dazu muss aber zufällig die Internet-Verbindung über seinen Rechner laufen. (Das ist übrigens fast nie so. Die normalen PC's sind immer nur Client-Rechner. Die echten Server sind meist Großrechner, an denen nur ausgewähltes Personal arbeiten darf, aber das könnte natürlich auch Carl sein oder ein neugieriger Geheimdienst usw. usf. Wir wollen hier auch nur das Prinzip deutlich machen.)
Rein theoretisch könnte Carl jetzt auch die Daten manipulieren und vielleich eine andere Nachricht einschleusen.



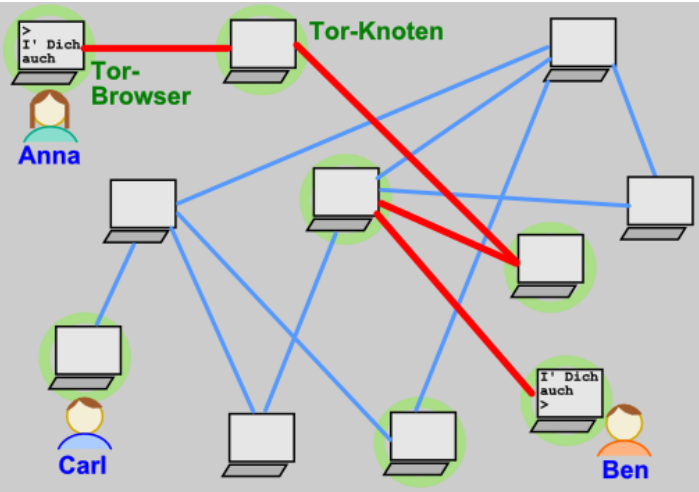


So könnte er eine Nachricht "Ich liebe Carl" an Ben schicken und würde so aussehen, als käme sie von Anna.
Anna kennt die Probleme der unsicheren Kommunikation im Internet. Also hat sie sich informiert und sich dann für das TOR-Netzwerk entschieden. Sie hat einfach den TOR-Browser heruntergeladen und benutzt ihn nun für die Kommunikation mit Ben.
Der TOR-Browser nutzt nun nur bestimmte Rechner im Internet für seine Kommunikation. Sie werden TOR-Knoten (Tor node) genannt. Vom TOR-Browser bis zum Ausstiegs-Rechner sind alle Verbindungen verschlüssel. Selbst, wenn Carl in diesem Netzwerk drin hängt, kann er nicht sinnvoll mitlesen.



Das Mitlesen wird auch dadurch erschwert, dass die Kommunikation mehrfach verschlüsselt wurde und die Zwischenstationen immer nur eine Verschlüsselung kennen.
Wenn Ben über die bestehende TOR-Verbindung (Circuit) antwortet, dann sind auch wieder alle Verbindungen – außer der ersten von Ben aus – verschlüsselt. Auch die Antwort könnte Carl nicht mitlesen.

Da Anna Ben über die Vorteile des TOR-Netzwerkes informiert hat, hat sich auch Ben damit versorgt.
Seine neue Verbindung ins Internet ist jetzt auch verschlüsselt und die Daten nehmen auch einen völlig anderen – wieder zufälligen Weg – durch das TOR-Netzwerk. Da ist der neugierige Mitleser schnell außen vor.
Ab sofort können Anna und Ben nun sehr sicher und ungestört kommunizieren.
Der TOR-Browser bedient nicht alle Internet-Services. Wenn man einen anderen nutzt, dann ist die Verbindung – trotz bestehender TOR-Verbindung – nebenläufig unverschlüsselt.



Nachteilig am TOR-Netzwerk ist die geringere Geschwindigkeit. Das mehrfache Ver- und Entschlüsseln und das zusätzliche Weiterleiten fordern ihren Tribut.

6. Texte strukturieren und gestalten

Texte überarbeiten und strukturieren

- *Fließtexte prüfen und korrigieren*

Die SuS achten sowohl auf sprachliche Korrektheit als auch auf die Einhaltung grundlegender, auf der DIN 5008 aufbauender Standards und Regeln für das maschinelle Schreiben. Die inhaltliche Prüfung der Texte kann unter Verwendung ausgewählter Quellen und Online-Medien erfolgen. Die Orthographie- und Grammatikprüfung wird reflektiert genutzt.

- Fließtexte strukturieren

Mittel zur Strukturierung sind u.a. Absätze, Zeilenumbrüche, Überschriften und Seitenumbrüche.

Texte formatieren

- *Attributwerte von Zeichen und Absätzen bestimmen und begründet ändern*
  - *Zeichenattribute: Schriftklassen, Schriftgrößen, Schriftschnitte*
  - *Absatzattribute: Ausrichtung, Zeilenabstand, Abstände vor und nach dem Absatz*
- *Formatvorlagen verwenden*
  - *Standard*
  - *Titel und Überschriften*
  - *Nummerierungen und Aufzählungen mit einer Gliederungsebene*

Die SuS unterscheiden Schriftarten mit und ohne Serifen und setzen sie dem Zweck entsprechend ein.
Schriftschnitte werden zur Akzentuierung von Textelementen verwendet.
Zur verbesserten Darstellung von Texten sind Werte von Absatzattributen gezielt zu ändern.

Bilder einfügen und anpassen

Eine Skalierung der Bilder erfolgt unter Beibehaltung des Seitenverhältnisses. Die Möglichkeit umlaufenden Textes kann exemplarisch thematisiert werden.

rechtliche Aspeckte beachten

Die SuS beachten sowohl den Schutz der eigenen Daten als auch den Schutz von Daten anderer Personen.
Das Urheberrecht sollte auch auf Textquellen bezogen werden.

Vorschlag zur inhaltlichen Vertiefung:
Die SuS modifizieren ausgewählte Formatvorlagen.

Problem-Fragen für Selbstorganisiertes Lernen

Erfassen und Bearbeiten von Texten

Texte erfassen und bearbeiten

Aspekte der Mikrolithografie beachten

Beim Start vieler Office-Programme bekommt man zuerst eine sogenannte Backstage-Ansicht, in der man über das nachfolgende Vorgehen entscheiden kann und muss. Da werden Vorlagen angeboten oder die alten gespeicherten Dateien oder ebnen die Möglichkeit, eine neue Date anzulegen.

Exkurs: Welches Textverarbeitungs-Programm soll ich benutzen?

Was auf den ersten Blick vielleicht ganz einfach ist, wird beim genaueren Betrachten ein echtes Problem. Es ghört zu den grpßen Streit-Themen in der angewandten Informatik, was wohl das beste Textverarbeits-Programm ist? Heute gibt es viele Programme im Angebot. Einige – relativ weit verbreitete und bekannte – Textverarbeitungen wollen wir hier mit ihren Merkmalen sowie Vor- und Nachteilen vorstellen.
Dabei gehen wir auf die Besonderheiten und die Funktionen ein, mit den wir bis zum Schulabschluß zu tun haben können.
Es empfiehlt sich, eine Textverarbeitung auszuwählen und diese dann konsequent zu benutzen. Das übt und man bekommt Erfahrungen, auch mal mit schwierigen Situationen fertig zu werden. Denn auch Textverarbeitungen sind Programme, die Fehler haben – so wie jedes Programm. Mit genügend Erfahrungen kann man die Stolpersteine umfahren und effektiv arbeiten.
Ein Wort noch zu den portablen Versionen. Die meisten Programme müssen mit Administrator-Rechten installiert werden. Nicht so die portablen Versionen. Sie kommen ohne Installation aus. Meist muss nur ein entzippter Ordner irgendwo gespeichert werden und das Programm kann aus diesem Ordner heraus gestartet werden. Neben der eigenen Festplatte kann man auch eine USB-Stick als Speicher-Ort wählen. Dann kann man diesen Stick fast an jedem Rechner nutzen und damit auch sein dort gespeichertes Programm. Es gibt für den USB-Stick auch noch ein komfortables Menü-System, miot dem man viele solcher portable Apps nutzen und verwalten kann. Es heißt portableApps (→ portableapps.com) und hat sogar ein eigenes Update-System. Da hat man dann immer die aktuelleste Version seines Programm's auf dem Stick. (Kleiner "Geheim"-Tipp: Das portable-Apps-Menü lässt sich mit all seinen Vorteilen auch auf einem PC einrichten (auch ohne Administartor-Rechte!).)

Programm	**Merkmale**	**Vorteile / PRO-Argumente**	**Nachteile / CONTRA-Argumente**
WORDPAD (microsoft)	• für Windows • RTF (auch: DOCX, TXT, …)	• bei Windows vorinstalliert • sehr übersichtlich	• keine Extra-Kosten • sehr eingeschränkter Funktions-Umfang •
WORD **(microsoft)**	• für Win und Mac • klassische Grund-Funktionen • Format-Vorlagen • Tabellen, Bilder, Grafik • automatische Verzeichnisse und Indices • Fußnoten • Serienbriefe • … • DOCX (auch: RTF, DOC, PDF, •	• **Quasi-Standard** • sehr viele Funktionen (läßt keine Wünsche übrig • erzeugt auch PDF • recht aktuelles Update-Management	• Kosten-pflichtig • keine portable Version • nur Ribbon-Menü's • für Android nur eingeschränkte Version oder Betrachter-Programme verfügbar

(c,p) 2015-2020 lern-soft-projekt: drews

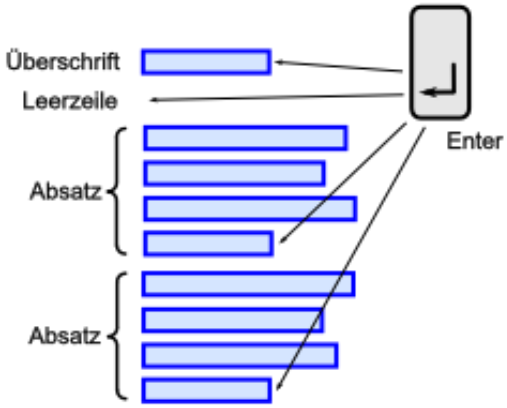
Programm	Merkmale	Vorteile / PRO-Argumente	Nachteile / CONTRA-Argumente
WRITER **(Libre- bzw. Open-Office)**	• für Win und Linux • klassische Grund-Funktionen • Format-Vorlagen • Tabellen, Bilder, Grafik • automatische Verzeichnisse und Indices • Fußnoten • Serienbriefe • … • ODT oder DOCX (auch: TXT, RTF, …) •	• **Kosten-frei** im Intenet • **portable Version** verfügbar • kann praktische alle üblichen Text-Dateien öffnen und speichern • erzeugt auch PDF • wird intensiv weiterentwickelt	• nur klassische Menü's • gewöhnungsbedürftiges Markieren und Bedienen •
TEXTMAKER (FREE) **(SoftMaker)**	• für Win, Mac und Linux verfügbar • klassische Grund-Funktionen • Format-Vorlagen • Tabellen, Bilder, Grafik • automatische Verzeichnisse und Indices • Fußnoten • (auch: DOCX, RTF, …) •	• Kosten-frei im Internet • traditionelle und Ribbon-Menü's	• Voll-Version Kosten-pflichtig • keine portable Version • etwas eingeschränkter Funktions-Umfang im Vergleich zur Voll-Version • kann kein ODT (speichern)
ABIWORD ()	•	• portableVersion •	•
Editoren • Editor (Notepad) • Notepad++ • …	• sehr eingeschränkte TV-Funktionen	• Dateien sind mit jedem anderen Programm lesbar • super für Programmierung	• es fehlen die Schul-notwendigen TV-Funktionen!!!

Mit diesen vielen Argumenten wird es wohl noch schwerer, die richtige Wahl zu treffen. Es gibt aber auch einige andere Aspekte zu beachten. Wenn z.B. die Schule ein bestimmtes Programm bevorzugt, dann ist man gut beraten, dieses auch zu benutzen. Zu oft könnte es sonst Probleme geben, wenn man Hausaufgaben in der Schule nutzen will, oder eben Schul-Dateien zuhause.
Bietet die Schule mehrere Programme an, dann kann man sich daran orientieren, was zuhause zur Verfügung steht. Genau ist es, wenn die von der Schule bevorzugten Programme nicht zuhause genutzt werden können. Kosten-pflichtige Programme sind nicht jedermans Sache. Dann bleibt nur die alternative

Nutzung von freien, Kosten-losen Programmen.
Letztendlich muss man sich aus seiner eigenen Situation heraus entscheiden. Die für eine schulische Textverarbeitung geeignten Programme sind in der Übersicht **fett** hervorgehoben.

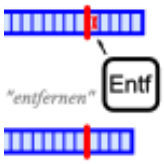
(c,p) 2015 – 2020 lsp: dre

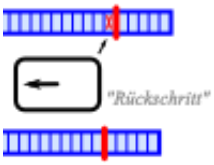
6.1. Text-Erfassung



Zuerst sollte man den Text immer ohne irgendeine Veränderung von Schriftarten usw. eingeben. Man benutzt die vorgegebene leere Seite des Textverarbeitungs-Programm's, um den Text einfach nur einzutippen. Lediglich Zeilenumbrüche mit der Taste [Enter] werden gemacht.
Solche Zeilenumbrüche werden nach Überschriften, für Leerzeilen und bei Absatzenden gebraucht.
Auf den ersten Blick ist es etwas ungewöhnlich, dass man innerhalb eines Absatzes am Ende der Zeile nicht auch einen Zeilenumbruch macht.
Früher – bei der Texteingabe an einer Schreibmaschine – war das auch notwendig. Da musste man kurz vor Erreichen des Endes einer Zeile immer überlegen, ob das Wort in die Zeile passt, ob man nur die ersten Silben eingibt oder einen Zeilenumruch macht.
Heute übernimmt der Computer – genaugenommen das Textverarbeitungs-Programm - genau diese lästige Arbeit. Das Programm testet bei der Eingabe, ob das Wort noch passt und sonst wird es einfach in die nächste Zeile gesetzt. Auch beim späteren Ändern korrigiert das Textverarbeits-Programm alle Wort-Positionen.
Am Schluß der Text-Erfassung erhält man so einen Rohtext, den man dann in geeigneter Form gestalten kann. Viele – gewissenhafte – Textarbeiter speichern sich diesen Rohtext auch extra ab, um ihn im Falle der Zerstörung des gestalteten Textes wieder als Ausgangs-Material zu haben.
In modernen Systemen mit der Möglichkeit die letzten Arbeitsschritte rückgängig zu machen und automatisch Sicherungskopien anzulegen, ist dieses Vorgehen nicht mehr unbedingt notwendig. Für größere Texte (Hausarbeiten, …) ist es aber immer noch ein gutes Mittel, wenigstens einen Text zu haben, auch wenn die gesamte Formatierung schief läuft.
Ganz moderne Systeme erzeugen aus bestimmten Rohtexten dann automatisch schön formatierte Versionen des gleichen Textes für sehr unterschiedliche Anwendungen, wie z.B. ein Buch oder eine Webseite.

6.1.1. Zwei Lösch-Tasten – Wozu braucht man die denn?





Als kleine Eselsbrücke kann man sich die Tasten-Lücke auf der Tastatur zwischen dem alfanumerischen Block und dem Steuerungs-Block als Cursor vorstellen. Die "Rückschritt"-Taste steht vor dem gedachten "Cursor", also wird das Zeichen davor gelöscht. Mit der Taste "Entf"ernen – hinter dem gedachten "Cursor" – löscht auch das Zeichen hinter dem Cursor.

6.2. Text-Bearbeitung

Mikro- und Makro-Typographie
(Mikro- und Makro-Lithographie)

6.2.1. Auswählen und Markieren von Texten

Markieren mit der Tastatur

Markieren mit der Maus

Markieren mit Touch

6.2.2. Zeichen-Formatierung

Mikrolithographie / Mikrotypographie
Gestaltung von Zeichen, Wörtern, Absätzen und ganzen Texten

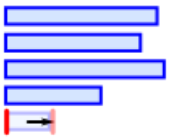
Schrift-Stile

Schriftgröße

Schriftarten

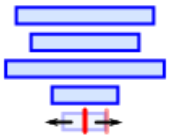
Ligaturen / Buchstaben-Verbund

6.2.3. Absatz-Formatierung

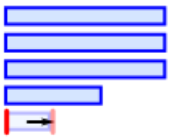


linker Text-Rand glatt
rechter Text-Rand von der Länge der in die Zeile pasenden Wörter abhängig
neue Zeile beginnt immer links und wächst nach rechts
Cursor nimmt rechtsseitige Position ein
Abstände zwischen den Wörtern immer gleich
typische Text-Ausrichtung in Briefen, modernen Zeitungen
besonders zur Text-Eingabe geeignet (andere Absatz-Formate sollten immer erst nachträglich gesetzt werden)

rechter Text-Rand glatt
linker Text-Rand abhängig von der Länge der in die Zeile passenden Wörter
neue Zeile beginnt immer rechts und wächst nach links
Cursor nimmt rechtsseitige Position ein
Abstände zwischen den Wörtern immer gleich



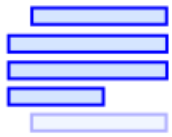
rechter und linker Text-Rand gleichweit vom Seiten-Rand entfernt
Text steht also exakt mittig
neue Zeile beginnt mittig und wächst zu beiden Seiten gleichzeitig
Cursor nimmt rechtsseitige Position ein
Abstände zwischen den Wörtern immer gleich



Linker und rechter Text-Rand sind mit dem Seiten-Rand deckungsgleich
nur die letzte Zeile ist linksbündig angelegt
neue Zeile beginnt immer links und wächst nach rechts
Cursor nimmt rechtsseitige Position ein
Abstände zwischen den Wörtern sind in innerhalb einer Zeile gleichweit, in anderen Zeilen sind die Abstände – je nach Zeilenspreizung – unterschiedlich
bei sehr weiten Abständen können mittels Silben-Trennung Teile von Wörtern aus einer folgenden Zeile hoch geholt werden und die – zu großen – Lücken füllen.
Der Blocksatz ist ein typischer Druck-Satz. Erst kurz vor dem Drucken stellt man auf Blocksatz.
Bei der Eingabe kann ein voreingestellter Blocksatz Zeilensprünge verursachen. Eine spätere Korrektur vorlaufender Zeilen bewirkt dann schnell viele Veränderungen in den nachlaufenden Zeilen und Texten.

Einrückungen der ersten Zeile eines Absatzes nutzt man um in großen Tex-

ten Absätze sicher abzugrenzen. Bei sehr langen End-Zeilen des vorherigen Absatzes ist das u.U. sonst schwer.
Texte mit einem Erstzeilen-Einzug machen – genau so wie im Blocksatz gesetzte Texte – einen professionelleren Eindruck.



Beginnt die erste Zeile am linken Seiten-Rand und die nachfolgenden Zeilen des Absatzes sind eingerückt, dann spricht man vom hängenden Einzug.
U.U. braucht man dieses Absatz-Format, wenn man Aufzählungen oder Gliederungen benutzt. Häufig zählt das Aufzählungs-Zeichen dann noch zur ersten Zeile. Damit der gesamte – eigentliche – Text dann einen glatten linken Rand bekommt, beginnt man die erste Zeile mit dem Aufzählungs-Zeichen weiter links.

Kapitel-Initiale
Gestaltungs-Mittel als dem historischen Buchdruck
meist als Kennzeichen für den Beginn eines größeren Abschnittes (Kapitels) benutzt.

Kapitälchen sind verkleinerte Groß-Buchstaben statt Klein-Buchstaben
z.B. für Text-Betonungen (Sperrung: N I C H T nachmachen!) oder für Namen von Persönlichkeiten / Wissenschaftler
Albert EINSTEIN und Marie CURIE

6.2.4. Seiten-Formatierung

Stege, Ränder Satzspiegel

Aufgaben:

1. *Erstelle mit Deinem Textverarbeitungs-Programm ein neues Dokument mit dem Namen "Rezept gefüllte Quark-Orange"! Für das weitere Speichern bist Du selbst verantwortlich!*
2. *Schreibe den nachfolgenden Text ab! Verwende dabei die voreingestellten Schriften usw. usf. Deines Textverarbeitungs-Programm's!*

Gefüllte Quark-Orange

Zutaten:
1 Orange (unbehandelt), 70 g Magerquark, 1 Messerspitze Ingwer-Wurzel (frisch), flüssiger Süßstoff, 1 Eiweiß (Eiklar)
Zubereitung:
Von der Orange oberes Drittel als Deckel abschneiden, davon eine Scheibe zum Garnieren beiseite legen. Die Orange aushöhlen und das Fruchtfleisch auspressen. Die ausgehöhlte Orange eine halbe Stunde ins Gefrierfach stellen. Den ausgepressten Saft mit dem Quark und zwei Spritzer Süßstoff zu einer cremigen Masse verrühren. Von dem Orangen-Deckel mit einer Reibe die orange Schicht abreiben und zur Masse geben. Eventuell mit weiterem Süßstoff abschmecken. Den Ingwer sehr fein hacken und in die Masse geben. Das Eiklar mit einer Prise Salz zu Ei-Schnee schlagen und vorsichtig unter die Quarkmasse heben. Die Masse in die gefrorene Orangen-Schale füllen und mit der zurückgelegten Orangen-Scheibe garnieren.

3. *Kontrolliere den Text auf exakte Schreibung und Zeichensetzung!*
4. *Sorge nun dafür, dass der gleiche Text noch zweimal unter dem eingegebenen Text auf der Seite vorhanden ist! Lasse zwei Leer-Zeilen zwischen den Texten!*
5. *Formatiere nun nach den folgenden Vorschriften!*
   a) *die Rezept-Überschrift soll immer fett und unterstrichen sein*
   b) *die Teilüberschriften "Zutaten" und "Zubereitung" sollen immer kursiv und fett gestaltet werden*
   c) *im ersten Rezept: alle Früchte und Wurzeln mit Effekte Kapitälchen*
   d) *im zweiten Rezept: alle Wörter 'Orange' in die Schriftart "Curlz MT" mit der Größe 14 pt und fett*
   e) *im dritten Rezept: alle Vorkommen des Wortes oder Wortteils 'Deckel' hochgestellt und die Vorkommen von 'Schale' tiefgestellt*
6. *Schreibe Deinen Namen unter das letzte Rezept und drucke das Dokument einmal aus! (Mehrfaches Drucken gibt Abzüge bei der Bewertung!)*

6.2.5. Texte professionell formatieren

Fließtexte erfassen und bearbeiten

Texte inhaltlich prüfen und korrigieren

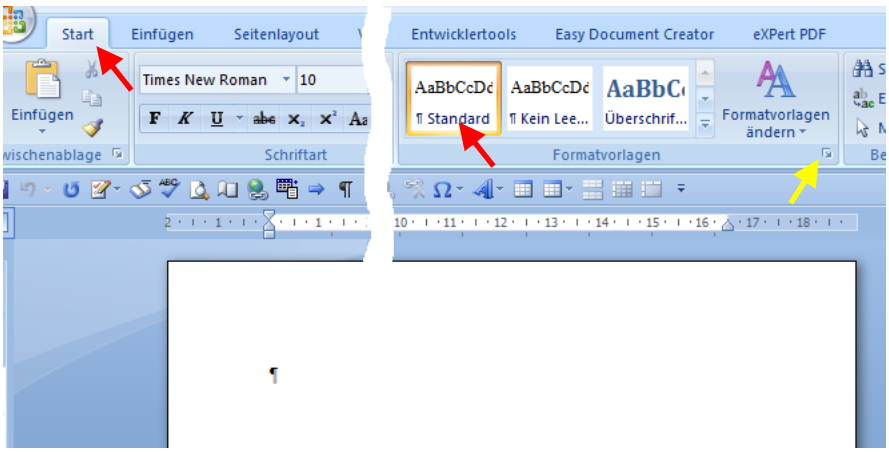
Möglichkeiten der maschinellen Textkorrektur reflektiert nutzen

Attributwerte von Zeichen- und Absatzobjekten bestimmen, darstellen und begründet ändern

Formatvorlagen

Das Gestalten großer Dokumente kann schnell zu Alptraum werden, weil man einfach den Überblick verliert, wie man die eine oder andere Überschrift formatiert haben möchte.
Was es denn nun Schriftgröße 15 oder 16, war es fett und kursiv zusammen oder nur das Eine davon? Welcher Blau-Farbton war es nun genau?
Spätestens ab drei, vier solcher Überschriften merkt man sich das nicht mehr alles so genau. Bearbeitet man dann noch verschiedene Dokumente, dann ist das Chaos vorprogrammiert.
Die meisten Textverarbeitungen bieten die Möglichkeiten Gestaltungs-Muster festzulegen. Diese werden i.A. Formatvorlagen genannt. In den modernen WORD-Versionen tauchen diese schon in den Symbol-Leisten oder Menü-Bändern auf.
Eine Formatvorlage enthält alle Festlegungen zur Formatierung. Dabei sind damit auch die Formate, die uns als halbprofessionelle Textverarbeiter interessieren.

Die Formatvorlage, mit der wir üblicherweise arbeiten, heißt "Standard". Sie beinhaltet die Festlegungen für die normalen Text-Absätze.
Eine Formatvorlage weist man einem Textabschnitt zu, indem man die Textstelle markiert und dann die entsprechende Formatvorlage aus der Vorlagen-Liste auswählt.
In der Menüband-Ansicht ist das Aussehen und die Benennung der Formatvorlagen nur teilweise sichtbar.
Praktischer ist die Listen-Ansicht. In ihr lassen sich später auch leicht Veränderungen an Formatvorlagen vornehmen.



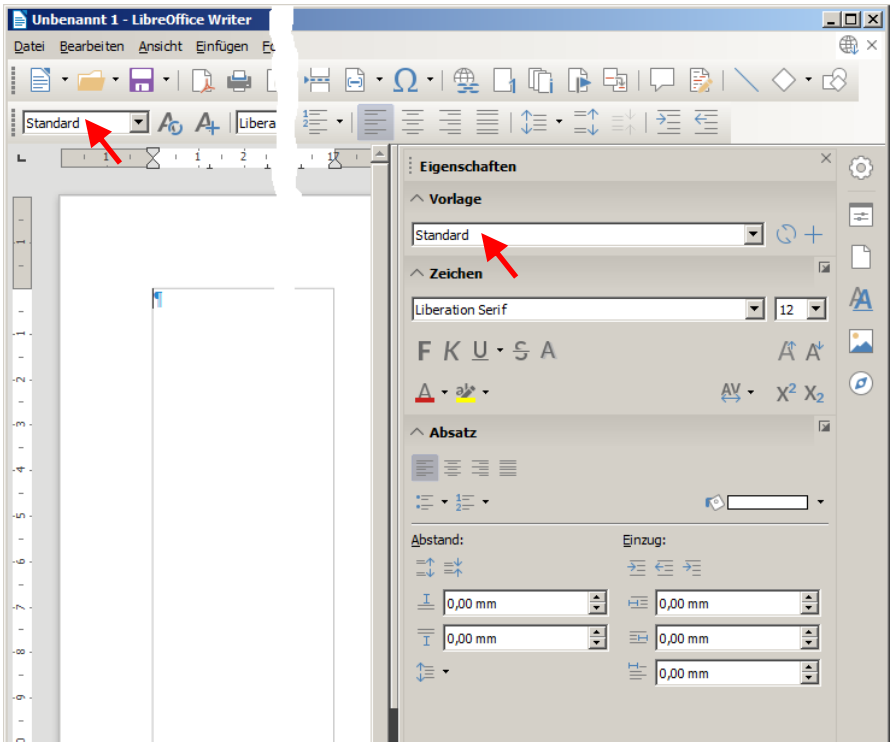
im Menüband "Start" findet man den Bereich "Formatvorlagen"
über die Erweiterung in der rechten unteren Ecke (gelber Pfeil)
gelangt man zur Listen-Ansicht

angedockte Listen-Ansicht der Formatvorlagen

In Libre-Office sind die Formatvorlagen

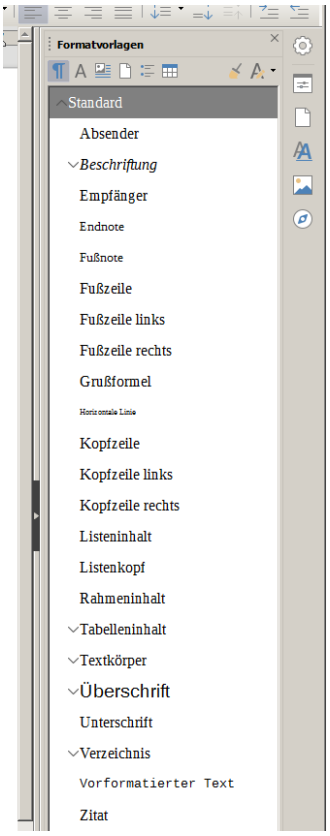
(c,p) 2015 – 2020 lsp: dre

– ganz klassisch – über eine Auswahl-Box erreichbar. Ev. muss man sich die "Symbolleiste" "Formatierungen" unter "Ansicht" einstellen.
Die Listen-Ansicht erhält man über die Taste [ ] oder die Menü-Punkte: "Ansicht" "Benutzeroberfläche" "Seitenleiste".
Die Seitenleiste ist eine sich ständig aktualisierende **Eigenschaften**-Anzeige. Hier können die jeweils passenden Attributwerte verändert werden.
Über die Taste [F11] kommt man zur Listen-Ansicht der Formatvorlagen.



Auswahl-Möglichkeiten für Formatvorlagen in LibriOffice Writer

Andere Textverarbeitungen nutzen mindestens eine der oben gezeigten Auswahl-Möglichkeiten. Das Prinzip ist immer gleich.



Listen-Ansicht der Formatvorlagen in Writer

Allerdings erfordert das Arbeiten mit Formatvorlagen ein abgestuftes und diszipliniertes Arbeiten mit Texten:

Arbeits-Regeln:

für kurze Texte, die in einem Ritt erstellt werden (z.B. Hausaufgaben, Briefe, Spickzettel, …)

1. zuerst nur reinen Text eingeben!
2. gesamten Text (z.B. mit [Strg] + [a]) markieren
3. dann auf den gesamten Text die Formatvorlage "Standard" setzen
4. die Überschriften markieren (als Zeile) und entsprechend ihrer Rangfolge mit "Überschrift 1 bis 3 usw." formatieren
5. restliche Formatierungen (z.B. Hervorhebungen, Hoch- und Tiefstellung, …) vornehmen

für größere Texte, die über lange Zeiträume erstellt werden (Facharbeit, Handout, …):

1. zuerst Zeile oder Absatz als normalen Text ("Standard") eingeben
2. Text markieren
3. Formatvorlage für Zeile oder Absatz auswählen und zuweisen
4. ev. Korrekturen an den Formatvorlagen vornehmen (s.a. →)
5. kleine Extra-Formatierungen (z.B. Hervorhebungen, Hoch- und Tiefstellung, …) vornehmen
6. mit nächster Zeile oder nächstem Absatz fortsetzen

Die letzte Arbeits-Variante ist dann für das normale Arbeiten zu empfehlen, wenn man sich an das Hantieren mit Formatvorlagen gewöhnt hat.

Als Mehrwert bietet es sich an, aus den – mit Formatvorlagen gestalteten – Überschriften ein Inhaltsverzeichniss erstellen zu lassen – und das fast ganz automatisch. Dazu mehr nach der Besprechung der einfachen Nutzung von Formatvorlagen.

grundlegende Formatvorlagen nutzen

Formatvorlagen nutzen

Objekte einfügen und anpassen

den Textfluss um Grafiken festlegen

Objekte und Strukturen in der Textverarbeitung

Wie auch in der Bild-Bearbeitung kennen wir in der Textverarbeitung verschiedene **Objekte**, denen wieder **Attribute** (Eigenschaften) und **Methoden** (Funktionen, Arbeitsschritte) zugeordnet werden.
Dieses Mal betrachten wir die Objekt-Struktur einmal von oben noch unten, also vom größten zum kleinsten Objekt. Diese Herangehensweise nennt man in der Informatik ""Top down" ("von oben nach unten").

Objekt:			**Bemerkungen / Hinweise**
Text-Dokument			
→ Unterobjekt(e):			
• Seite			
•			
→ Attribut(e):		***Wert***	
• Dateiname	=	*Bewerbung*	dieses Attribut kann der Nutzer direkt ändern
• Dateityp	=	*DOCX*	
• geändert	=	*wahr*	dieses Attribut wird vom Programm verwaltet
• Schreibschutz	=	*wahr*	
• Passwort	=	*p6$Rq=%*	Passwort wird verschlüsselt abgespeichert
•	=		
→ Methode(n):			
• Neu erstellen			
• Speichern unter …			
• Speichern			
• Öffnen			
• Schließen			
•			

Was sagt jetzt die Objekt-Darstellung aus? Das Objekt hat den Namen "Text-Dokument". Zu diesem Objekt gehört das Unterobjekt "Seite". Darauf gehen wir gleich noch weiter ein.
DasText-Dokument-Objekt hat diverse Eigenschaften – die wir eben **Attribute** nennen. Z.B. besitzt ein Text-Dokument einen Namen. Dieser Attribut-Wert – also die aktuelle Belegung des Attributes kann zu Anfang auch noch "Unbenannt x" oder "Dokumentx" lauten. Irgendwann wird dem Dokument ein passender Name zugeordnet – also ein neuer Attribut-Wert zugeordnet. Im obigen Beispiel ist es die Bezeichnung: ***Bewerbung.DOCX***.
Eines der wichtigsten Attribute ist "geändert". Es ist beim Starten einer Text-Verarbeitung auf "Falsch" (auch: "False" oder "Nein") gesetzt. Sobald man auch nur die kleinste Veränderung am Dokument vornimmt, z.B. das Eingeben eines Leerzeichens oder die Änderung des Seitenformates, sofort wird das "geändert"-Attribut auf den Attribut-Wert "Wahr" (auch: "True" oder "Ja") gewechselt. Ab dieseem Augenblick muss der Text-Benutzer entscheiden, ob er diese – auch noch so kleine (oder eben größere) Veränderung speichern möchte. Man kommt nicht um diese Entscheidung herum!
Damit sind wir auch schon bei den Methoden. Sie beschreiben ja die möglichen Arteiten / Tätigkeiten. die mit dem Objekt möglich sind. Das "neu erstellen" wird heute immer schon automatisch beim Start der Textverarbeitung erledigt. Aber man kann sich fast immer auch weitere Text-Objekte "neu erstellen".
Das "Speichern" und "Speichern unter …" sowie das spätere "Öffnen" haben wir schon bei den Grafik-Programmen besprochen (→).
Das "Schließen" beendet eine Text-Ansicht oder –Bearbeitung. Allerdings nicht bevor das "geändert"-Attribut abgetestet wird. Ist es "Wahr" muss der Nutzer entscheiden, ob die Veränderungen gespeichert werden sollen oder nicht. Übrigens – entgegen vieler Nutzer-Behauptungen – diese Funktion funktioniert in allen Programmen perfekt. Es ist quasi eines

der ersten Arbeiten, die ein Programmierer erledigt. Ein Programm – egal welches – schliesst nicht ohne diese Prüfung! Es ist immer der Nutzer, der die Entscheidung getroffen hat. (Ausnahme sind lediglich Programm-Abstürze, dann ist aber meist auch die Datei beschädigt!)
Kommen wir nun zu einem Unterobjekt – der "Seite". Auf einer Seite können verschiedene (Unter-)Objekte angeordnet werden. Das sind in der klassischen Textverarbeitung Absätze, Tabellen und Grafiken.

Objekt:			**Bemerkungen / Hinweise**
Seite			
→ Unterobjekt(e):			
• Absatz			meint Text
• Tabelle			
• Grafik			
•			
→ Attribut(e):		***Wert***	
• Seitenformat			
○ Seitengröße	=	*A4*	
○ Orientierung	=	*hoch*	
○ Randabstände			
▪ links	=	*2 cm*	
▪ rechts	=	*3 cm*	
▪ oben	=	*2 cm*	
▪ unten	=	*2 cm*	
• gefüllt / nächste Seite	=	*falsch*	
•	=		
→ Methode(n):			
• Neu erstellen			erzeugt eine neue Seite mit dem akt. Format
• Seitenformat festlegen / ändern			
• Ansicht aktualisieren			
• Druckvorschau erzeugen			
•			

Informatiker nutzen gerne auch noch eine zweite Schreibweise, um sich die Zusammenhänge von Daten-Elementen (Objekten) genauer zu verdeutlichen. Diese sieht z.B. so aus:

```
Dokument  ::=  Seite
```

Dabei liest man die doppelten Doppelpunkte mit dem Gleichheitszeichen (**::=**) als "setzt sich zusammen aus". Ein Dokument besteht also aus einer Seite. Aber es können ja auch mehrere Seiten werden. Das kann man nun durch Aufzählungen notieren. Für zwei Seiten sähe das dann so aus:

```
Text  ::=  Seite  Seite
```

Entsprechend für drei oder vier Seiten:

```
Text  ::=  Seite  Seite  Seite
Text  ::=  Seite  Seite  Seite  Seite
```

Solche unbestimmten Wiederholungen verkürzt man durch die verwendung von geschweiften Klammern (**{ }**). Das Objekt, welches in der geschweiften Klammer steht, muss aber mindestens einmal vorhanden sein. Somit ergibt sich die verkürzte Notierung:

```
Text  ::=  { Seite }
```

Damit kann unser Text-Objekt aus beliebig vielen Seiten bestehen. Wer schon etwas mehr Erfahrung mit Textverarbeitungs-Programmen hat, weiss auch, dass man für jede Seite ein anderes Format (z.B. Hochformat oder Querformat oder eine andere Größen als A4) festlegen kann. Diese Eigenschaften sind also der Seite zugeordnet und nicht dem Dokument.
Der Aufbau einer Seite in der Text-Schreibweise sieht so aus:

```
Seite  ::=  { Absatz  |  Tabelle  |  Grafik }
```

Die senkrechten Striche kennzeichnen dabei die Alternative. Es kann also ein Absatz oder eine Tabelle oder eine Grafik auf einer Seite angeordnet sein. Durch die geschweiften Klammern haben wir wieder die beliebige Wiederholung der Objekte. Somit kann eine Seite auch aus z.B. fünf Absätzen bestehen.
Schauen wir uns nun noch das Objekt **Absatz** an. Es stellt einen typischen Text-Abschnitt dar. Im einfachsten Fall besteht der Absatz nur aus einem Zeilen-Umbruch. Das entspricht einer Leerzeile. In der Zeile können aber auch mehrere Leerzeichen-getrennte Worte stehen. Dann hätten wir es z.B. mit einer Zeile oder Überschrift zu tun. Enthält der Absatz viele Worte in einem mehrzeiligen Fliesstext, dann haben wir den typischen (Text-)Absatz vorliegen.

```
Absatz  ::=  { Zeichen } Zeilenumbruch
```

Die etwas umfassendere Objekt-Darstellung zeigt die vielfältigen Möglichkeiten auf, die bezüglich eines Absatzes in einer Textverarbeitung nutzbar sind.

Objekt:		**Bemerkungen / Hinweise**
Absatz		
→ Unterobjekt(e):		
• Zeichen		alle druckbaren, wie auch d. Steuer- Zeichen
•		
→ Attribut(e):	***Wert***	
• Absatzformat		
○ Ausrichtung	= *links*	
○ Einzug	= *hoch*	
▪ 1. Zeile	= *0,5 cm*	
▪ ab 2. Zeile	= *0 cm*	
○ Rahmen	= *falsch*	
○ Zeilenabstand	= *1,5*	
○ Abstand vor Absatz	= *3 pt*	
○ Abstand nach Absatz	= *6 pt*	
○ Tabulatoren	= *2*	
▪ Tabulator 1	= *0,5 cm*	
▪ Tab 1 Ausrichtung	= *links*	
▪ Tabulator 2	= *12,5 cm*	
▪ Tab 2 Ausrichtung	= *rechts*	
•	=	
→ Methode(n):		
• Neu erstellen		erzeugt eine neuen Absatz mit dem akt. Format
• Absatzformat setzen		
• Absatz darstellen / anzeigen		
•		

In unserer kleinen Objekt-Übersicht für eine Textverarbeitung fehlt jetzt nur noch das Objekt **Zeichen**. Wir haben schon gesehen, dass es zwei Arten von Zeichen gibt. Da sind einmal die ganznormalen – druckbaren – Zeichen. Zu ihnen gehören alle Buchs,taben Ziffern, Satzzeichen und Spezialzeichen. Dabei ist es für die Textverarbeitung völlig egal, ob es ein latei-

nischer Buchstabe oder ein chinesisches Silbensymbol ist. Das Textverarbeitungs-Programm hat keine "Ahnung" von der Bedeutung des Zeichen. Für das Programm ist es einfach ein Symbol.
Die zweite Art Zeichen sind die Steuerzeichen. Sie sind nicht-druckbar. Dazu gehört z.B. der Zeilenumbruch. Exakt gesehen sind es eigentlich zwei Zeichen. Das erste Zeichen ist "Wagenrücklauf" gefolgt von "Zeilenvorschub". Diese Kombination stammt noch aus den Urzeiten der Fernschreiber (Fernmelder) – das waren praktisch Schreibmaschinen, die über Telefonleitungen (oder Funk) miteinander verbunden waren. Da mussten die Schreibköpfe noch so gesteuert werden.
Die Text-Schreibung würde dann z.B. so aussehen:

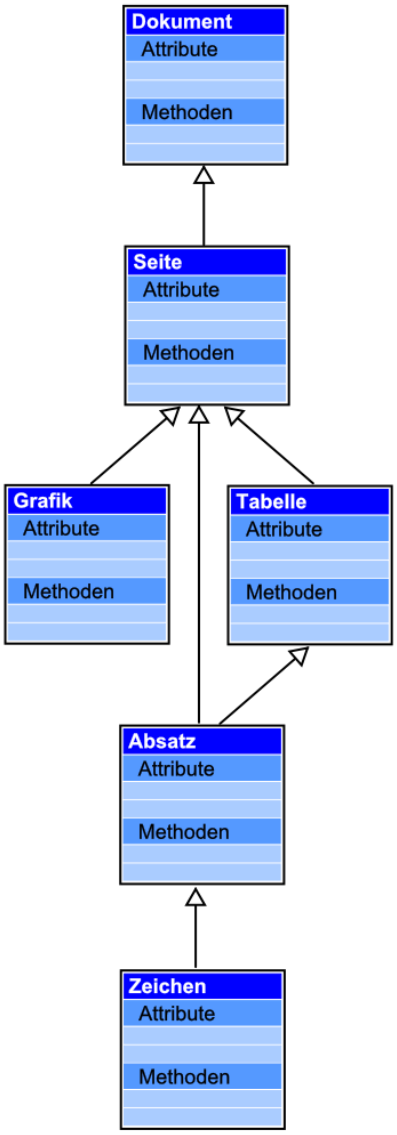
```
Zeichen  ::= druckbaresZeichen  |  nichtdruckbaresZeichen

druckbaresZeichen       ::= 0 | 1 | 2 | … | 9 | A | … | Z | a | … | z | …
nichtdruckbaresZeichen  ::= CR | LF | BEL | EOT | …
```

Das Objekt Zeichen ist in den modernen Textverarbeitungen mit vielen Attributen ausgestattet. Sie machen ein tolles Layout möglich.

Objekt:			**Bemerkungen / Hinweise**
Zeichen			meint hier nur druckbare Zeichen
→ Attribut(e):		***Wert***	
• Zeichen-Code	=	*0041h*	Windows-interner Code für "**A**"
• Zeichenformat			
○ Schriftart	=	*Arial*	
○ Schriftstil			
▪ fett	=	*wahr*	
▪ kursiv	=	*falsch*	
▪ unterstrichen	=	*falsch*	
○ Schriftfarbe	=	*rot*	intern auch Nummer: RGB = 255,0,0
○ Hintergrundfarbe	=	*keine*	RGB = 255,255,255 (weißer Hintergrund)
•			
→ Methode(n):			
• Neu erstellen			Übernahme von der Tastatur
• Schriftart setzen			
• Schriftstil ändern			
• Schriftfarbe setzen			
• Hintergrundfarbe setzen			
•			

Zum Schluss wollen wir uns die Zusammenhänge zwischen den Objekten noch mal in einer graphischen Darstellung ansehen. Auf die speziellen Attribute und Methoden verzichten wir hier, damit das Ganze übersichtlich bleibt.
Solche Objekthierrarchien durchziehen die moderne Informatik. Sie machen die Programmierung von so komplizierten Programmen, wie es Textverarbeitungen heute sind, erst möglich.
Wer sich die Anfänge von Textverarbeitung ansehen möchte kann ja mal einen Editor benutzen und dort einen Schreibmaschinen-ähnlichen Text erstellen.



Aufgaben:

1. *Ermittle für ein (vom Kursleiter) vorgegebenes Dokument für 15 Attribute welche Attributwerte gesetzt sind!*
2. *Ordne die folgenden Sachverhalte den Kategorien Objekt, Attribut, Methode und Attributwert zu!*

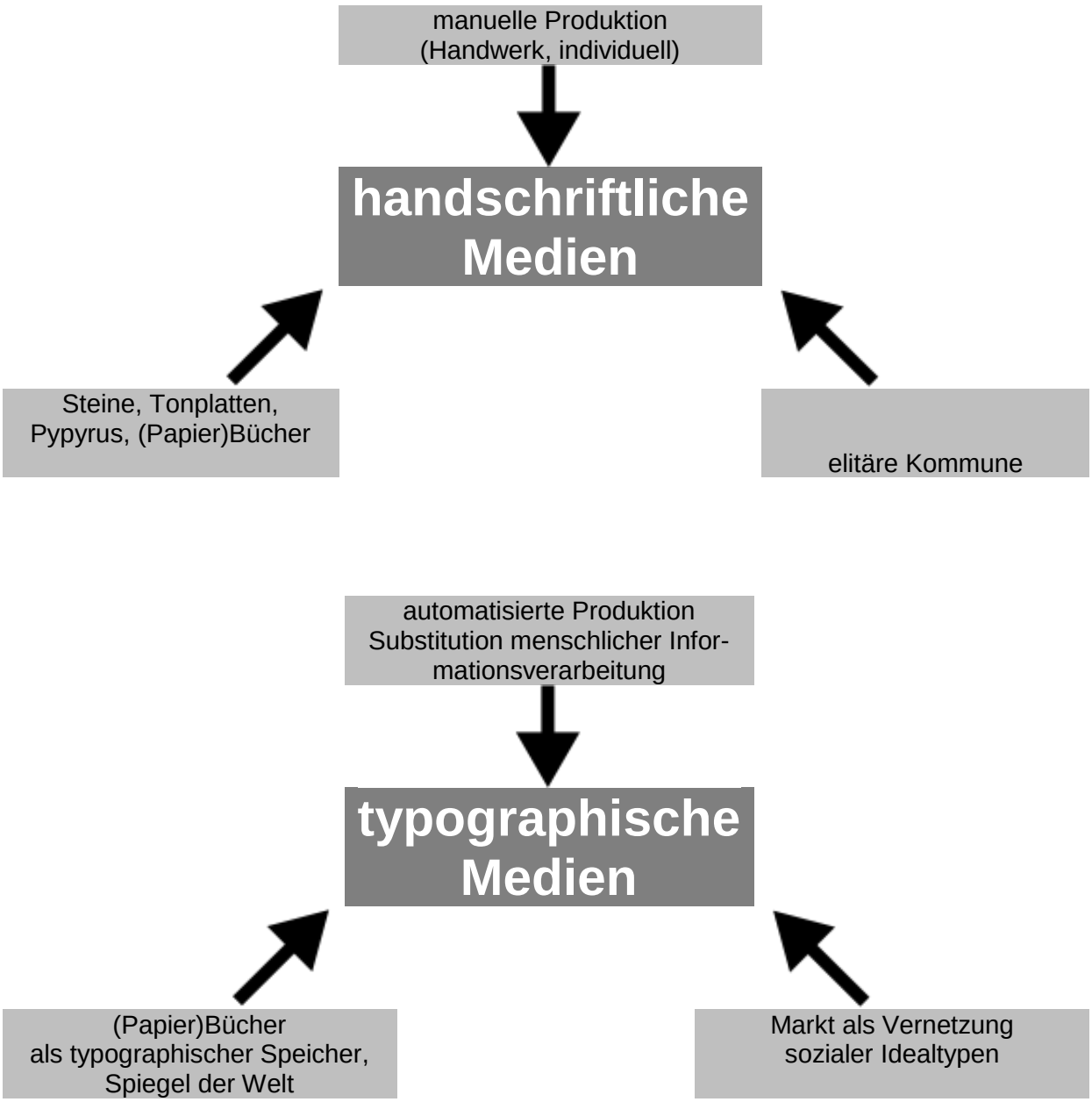
a)	Absatz	b)	wahr	c)	2 cm
d)	Hintergrundfarbe	e)	rechts	f)	darstellen
g)	24 pt	h)	Seitengröße	i)	Schriftstil

3. *In einer besseren Textverarbeitung sollen jetzt die Objekte Kopf- bzw. Fuß-Zeile integriert werden. Beide sollen als ein Objekt "Randzeile" integriert werden, allerdings mal mit Position "oben" und mal "unten". Erweitern Sie die Objekt-Hierrarchie an einer geeigneten Stelle! Definieren zugehörige / notwendige Details zum Objekt!*

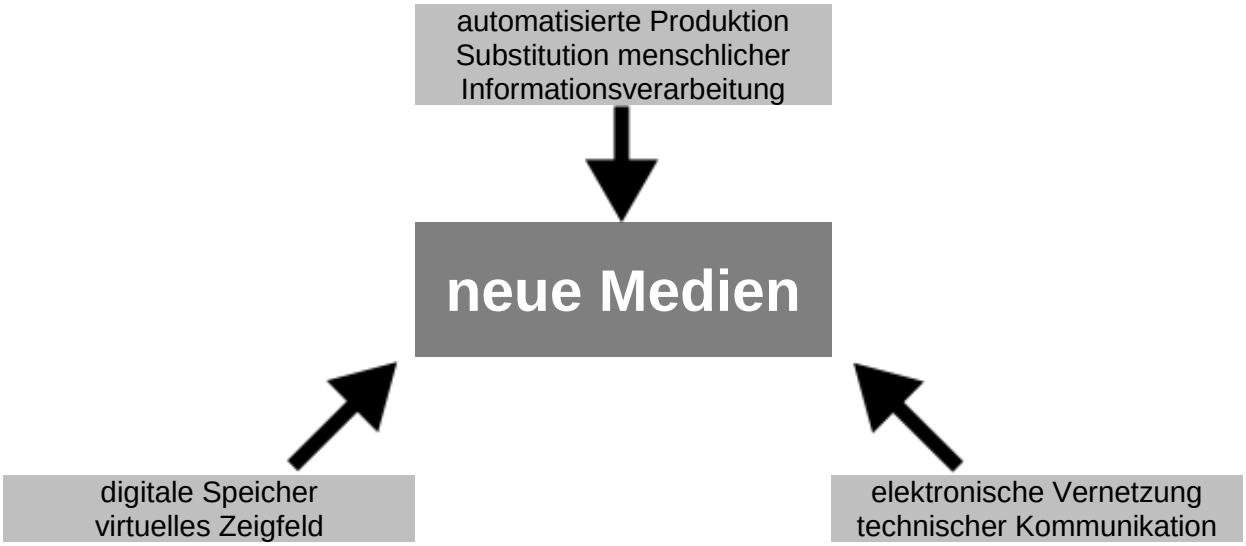
für Experten und zum Knobeln:

1. *Bestehen eigentlich Zeichen auch aus irgendwelchen (Unter-)Objekten? Wenn JA, welche, wenn NEIN, warum geht das nicht!*

4.3. (gesellschaftliche / historische Aspekte)



Q: IN: Log IN Heft 187/188 (2017) S.29 (Layout geänd.)



Q: IN: Log IN Heft 187/188 (2017) S.29

Bildrecherche in Mediensammlungen unter Nutzung von Filtern zielgerichtet ausführen und die Ergebnissee nach inhaltlichen, gestalterischen und technischen Kriterien bewerten

Quellen unter Nennung des Urhebers und des genauen Fundorts angeben

Grafiken und Fotos einfügen und anpassen

Jahrgangsstufe 7

7. Spiele entwickeln und multimedial dokumentieren

Algorithmen und Daten

- *Regeln und Abläufe beschreiben*

Dazu gehören das Formulieren der Spielregeln, das Dokumentieren des Programms oder das Anfertigen eines Storyboards.

- *Algorithmen beschreiben, interpretieren und modellieren*

Die Beschreibung des Algorithmus erfolgt durch die Blockdarstellung und verbal. Die Algorithmen enthalten geschachtelte Strukturen und zusammengesetzte bedingungen.

- *Operationen mit Daten verschiedener Datetypen ausführen*

Die SuS entscheiden, ob die zu verarbeitenden Daten vom Typ Zahl, Zeichenkette oder Wahrheitswert sind und wählen geeignete Operatoren aus.

Medien gestalten

- *grafische Elemente gestalten*

Die SuS gestalten Raster- und Vektorgrafiken für die Verwendung im Spiel.

- *Arbeitsprozess oder –ergebnis multimedial dokumentieren*

Möglichkeit des multimedialen Dokumentierens sind z.B. eine illustrierte Bedienungsanleitung, ein Bildschirmmitschnitt, ein Werbefilm oder ein Audiobeitrag. Die SuS beachten das Urheberrecht und Persönlichkeitsrechte.

Medien nutzen

- *den eigenen Umgang mit Computerspielen reflektieren*

das Prinzip eines Komprimierungsverfahrens anschaulich erklären

Die Dateigröße des Medienprodukts ist Anlass für eine Diskussion über Möglichkeiten der Datenkomprimierung durch Lauflängenkodierung, eine Reduktion der Farbtiefe oder der Wahl des Dateiformats.

Problem-Fragen für Selbstorganisiertes Lernen

Eigenständigkeit im Umgang mit Anwendungen

Gebrauch der Fachsprache

Informationen strukturieren und adressatengerecht präsentieren

Techniken und Funktionalitäten des Betriebssystems und der genutzten Software zum effektiven Präsentieren nutzen

Recht auf informelle Selbstbestimmung beachten und wahrnehmen

Foto-, Audio- und Video-Produkte erstellen

7.x. Multimedia-Spiele erstellen

algorithmische Strukturen interpretieren und implementieren

geschachtelte algorithmische Strukturen interpretieren und implementieren

Operationen mit Daten ausführen

Regeln und Abläufe beschreiben

6.x.1.6. Programmieren des eigenen Smartphone's

Nachdem wir uns (eventuell) mit den verschiedenen Experimentier-Board's und Robotern beschäftigt haben, stellt sich irgendwie die Frage: Können wir auch größere System programmieren? Können wir vielleicht unser eigenes Smartphone's das machen lassen, was wir wollen?
Was auf den ersten Blick etwas schwierig aussieht, ist in der Praxis recht einfach realisierbar.
Das nötige Werkzeug hat das MIT (Massachusetts Institute of Technology) entwickelt. Das System nennt sich App Inventor" – also "App-Erfinder" bzw. "Programm-Erfinder". Neben der schon bekannten Block-orientierten Programmierung können wir nun auch Objekte zur Anzeige bringen. Diese Bedien-Elemente sind uns aus dem täglichen Bedienen unserer

Smartphons und Computer allzugut bekannt. Dazu gehören z.B. Schaltflächen (Button's), Schieberegler (Slider) oder Auswahlboxen (Spinner).
Letztendlich können wir uns eine eigene App schreiben, die optisch kaum von einer professionellen App zu unterscheiden ist.
Interessant sind auch die Möglichkeiten, die Sensoren der Smartphon's abzufragen oder LEGO ® Mindstorms ® Roboter mit der eigenen App zu steuern.

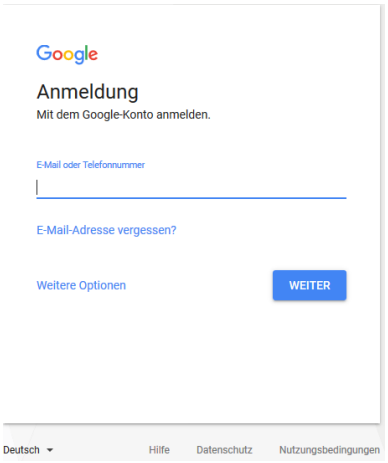
basiert auf Java
deshalb auch stark Objekt-orientiert

6.x.1.6.0. Wege zum MIT App Inventor

online: http://appinventor.mit.edu/

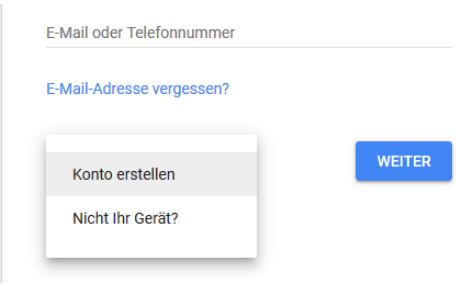
Links:
http://www.appinventor.org/book2 (ein engl. Buch zum AI2 als PDF's (BY-NC-SA-Lizenz))
http://appinventor.mit.edu/explore/ai2/support/blocks.html (Hilfe zu den eingebauten Block's)
http://ai2.appinventor.mit.edu/reference/components/ (Dokumentation zu den Komponenten)

Vorarbeit: Anlegen eines google-Konto's

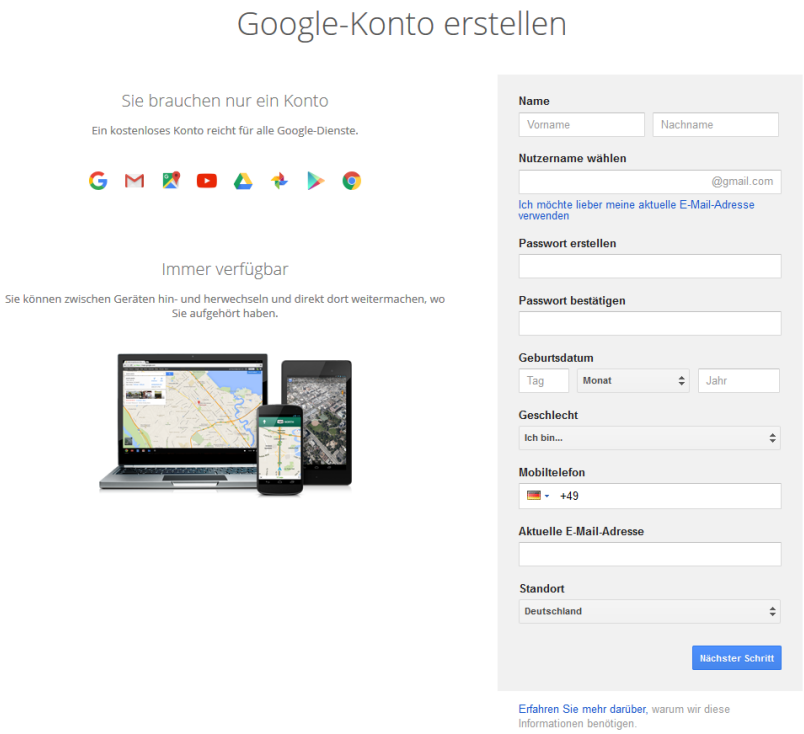


Leider funktioniert der MIT App Inventor nur vollständig mit einer Anmeldung über ein google-Konto.
Um dabei möglichst wenig in die Fänge von google zu geraten, gibt es zwei Möglichkeiten.
Entweder man legt sich ein quasi anonymes google-Konto an und nutzt dieses nur für die Programmierung und vielleicht anderen anonymen Aktivitäten. Oder die zweite Möglichkeit ist die Erstellung eines google-Konto's mit der eigenen eMail-Adresse.
Wir empfehlen die erste Variante. Dort muss man nicht wirklich persönliche Daten hinterlegen und hat ein zusätzlich zur normalen eMail-Adresse eine zweite anonyme.
Unter accounts.google.com kann man ein neues google-Konto anlegen (direkt: → https://accounts.google.com/SignUp).
Hat man schon ein Konto eingerichtet, dann wird unter Umständen jetzt schon ein Anmeldename (Kontoname) eingetragen sein (→ Anmeldung mit google-Konto).
Wer noch keinen Account hat geht auf den Schriftzug "Weitere Optionen". Man kann dann zwischen "Konto erstellen" und "Nicht Ihr Gerät?" wählen.
Zum Späteren Anmelden an mehrfach genutzten PC's z.B. in der Schule macht dann die zweite Opti-

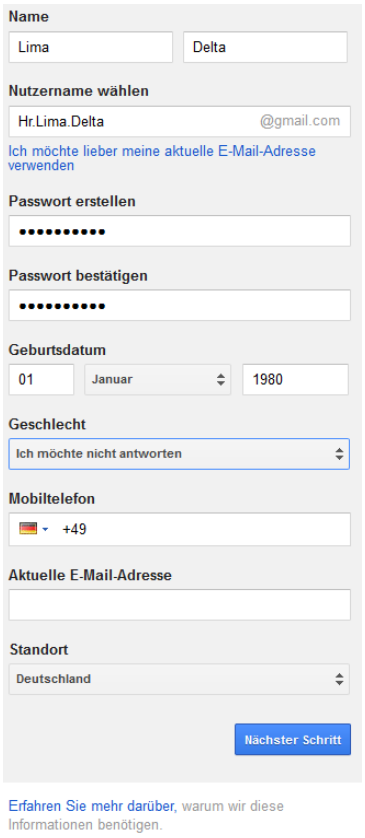
on einen Sinn. Wir nutzen zu Erstellen unseres google-Konto jetzt "Konto erstellen".



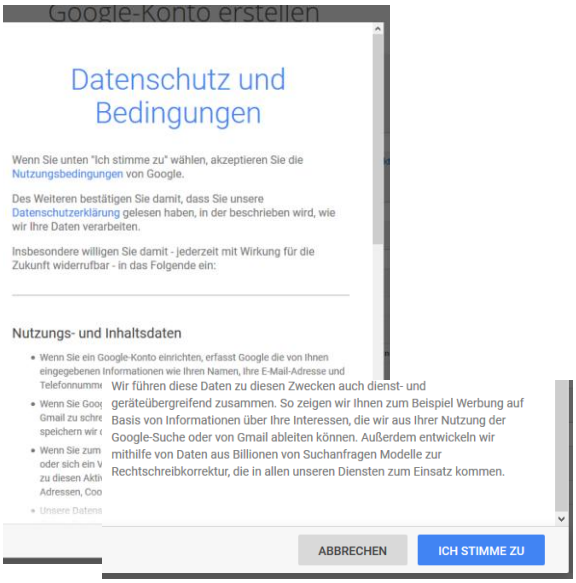
Die angeforderten Daten sollte man jetzt nach Gutdünken ausfüllen. Wer das google-Konto als sein Standard-eMail-Konto nutzen möchte, macht exakte Angaben, für die anderen Nutzungs-Zwecke können die Daten schon unbestimmter ausfallen.
In dem Fall sollte man auch Wert auf eine aussagekräftige eMail-Adresse legen. Hier hilft nur einwenig rumexperimentieren oder sich mit einem Vorschlag von google zufrieden zu geben.
Aber Achtung, nicht nun doch noch extra Daten über die Adresse freigeben.



Das Geburtsjahr usw. sollte da nicht drin vorkommen!
Eine Handy-Nummer oder eine andere eMail-Adresse muss man nicht angeben. In dem Fall muss man sich aber auch sein gewähltes Passwort gut merken, denn es gibt dann keinen Rettungsschirm (bei "Passwort vergessen").
Ein Beispiel für ziemlich anonyme Daten zeigt das nebenstehende Anmelde-Formular. Das kann man natürlich noch weiter treiben. Aber ab und zu braucht man doch die Daten wieder und dann sind so einigermaßen verständliche Angaben schon recht hilfreich.

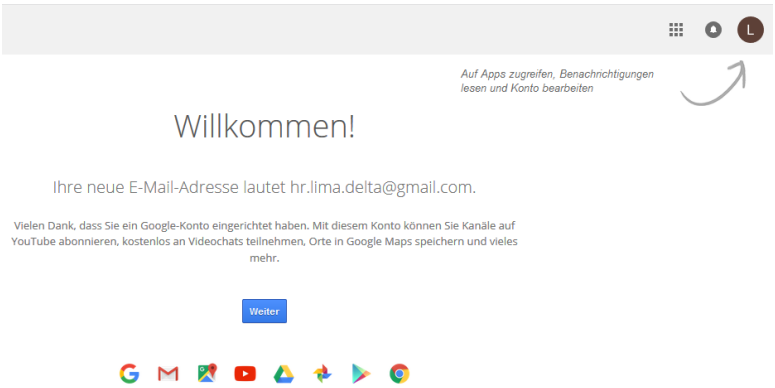


Die nachfolgenden Nutzungs-Bedingungen muss man nun durchlesen / durchscrollen und am Schluss bestätigen.



Wenn alles geklappt hat, dann erhalten wir eine passende Information und können nun direkt "weiter"-machen.
Interessant ist in der folgenden Konto-Anzeige die Möglichkeit sich einen "goggle Drive"-Speicher anzulegen. Ob allerdings so eine Cloud bei google die optimale Lösung ist, hängt

von den persönlichen Nutzungs-Optionen ab. Hier sollte man nicht unbedingt seine privaten Dokumente, Bilder usw.usf. speichern.



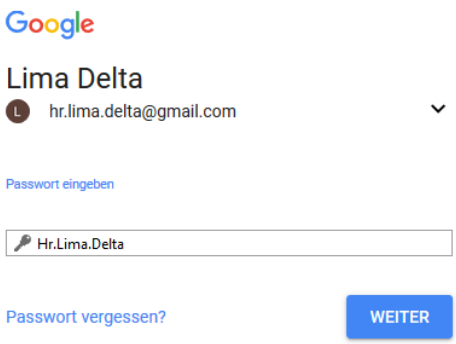
Der direkte Zugriff auf die eMail-Funktionen ist über www.google.com/gmail möglich. Die eigene Cloud (Google Drive) erreicht man mit drive.google.com/drive.

Ab nun kann ich mich mit meinem google-Account direkt bei den verschiedensten Internet-Diensten anmelden. Wir wollen das ja gleich beim MIT App Inventor tun.

Anmeldung mit google-Konto

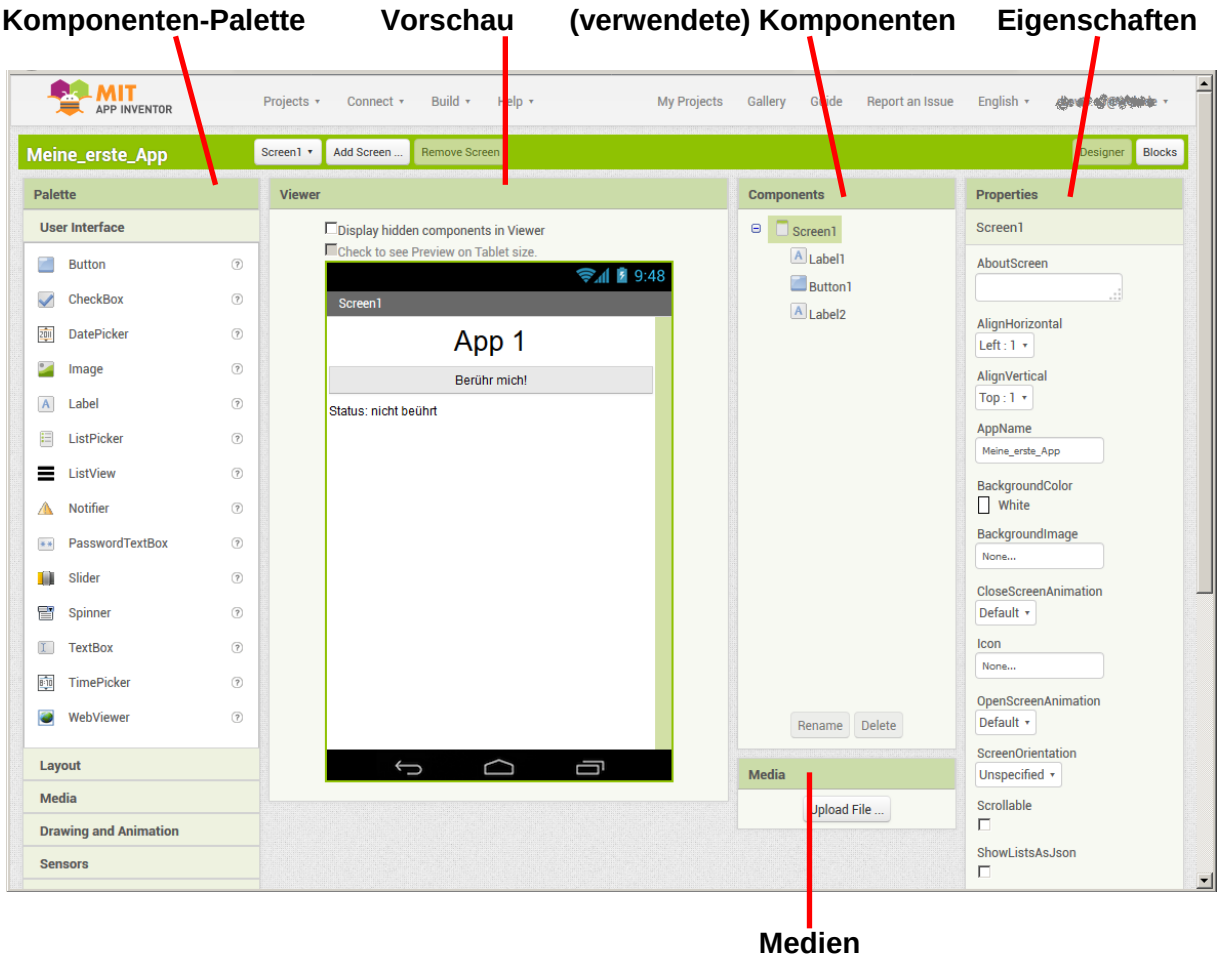
MIT App Inventor – online
online: http://ai2.appinventor.mit.edu/

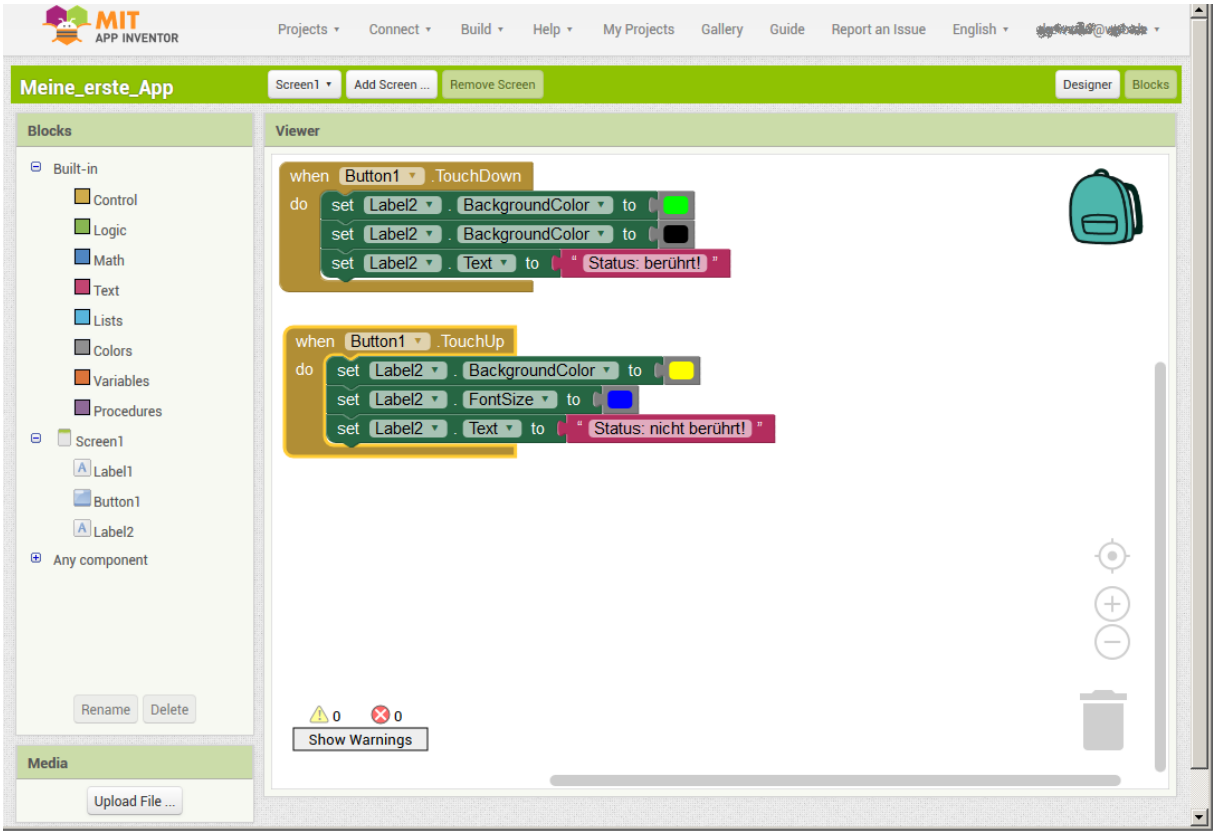
http://appinventor.mit.edu/



Vorteile
praktisch immer verfügbar, wenn man eine Internet-Verbindung hat
Update's sind nicht notwendig (Webseiten-Betreiber kümmert sich um alles)
keine Installation notwendig

Nachteile
nur online verfügbar
Abhängigkeit vom Internet
bei Volumen- oder Zeit-gebundenen Internet-Verbindungen nicht günstig
eigene Dateien (Programme, App's) sind nur dann immer verfügbar, wenn man sich Kopien auf den eigenen Datenträger angelegt hat
wenn die Webseite offline geht / schließt, dann verliert man ev. seine Daten und auch weitere Möglichkeiten zum Weiter-Programmieren (mit diesem System)

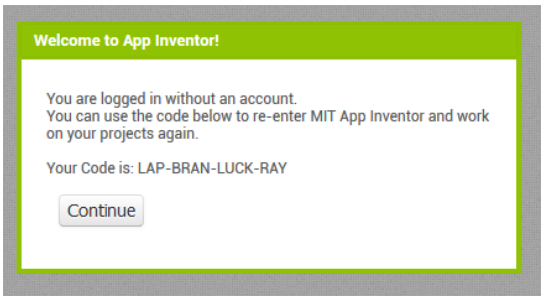

Komponenten-Palette
Vorschau
(verwendete) Komponenten
Eigenschaften
Medien
MIT APP INVENTOR
Projects
Connect
Build
Help
My Projects
Gallery
Guide
Report an Issue
English
Meine_erste_App
Screen1
Add Screen ...
Remove Screen
Designer
Blocks
Palette
User Interface
Button
CheckBox
DatePicker
Image
Label
ListPicker
ListView
Notifier
PasswordTextBox
Slider
Spinner
TextBox
TimePicker
WebViewer
Layout
Media
Drawing and Animation
Sensors
Viewer
Display hidden components in Viewer
Check to see Preview on Tablet size.
9:48
Screen1
App 1
Berühr mich!
Status: nicht beührt
Components
Screen1
Label1
Button1
Label2
Rename
Delete
Media
Upload File ...
Properties
Screen1
AboutScreen
AlignHorizontal
Left : 1
AlignVertical
Top : 1
AppName
Meine_erste_App
BackgroundColor
White
BackgroundImage
None...
CloseScreenAnimation
Default
Icon
None...
OpenScreenAnimation
Default
ScreenOrientation
Unspecified
Scrollable
ShowListsAsJson


MIT APP INVENTOR
Projects
Connect
Build
Help
My Projects
Gallery
Guide
Report an Issue
English
Meine_erste_App
Screen1
Add Screen ...
Remove Screen
Designer
Blocks
Blocks
Built-in
Control
Logic
Math
Text
Lists
Colors
Variables
Procedures
Screen1
Label1
Button1
Label2
Any component
Rename
Delete
Media
Upload File ...
Viewer
when Button1 .TouchDown
do set Label2 . BackgroundColor to
set Label2 . BackgroundColor to
set Label2 . Text to " Status: berührt! "
when Button1 .TouchUp
do set Label2 . BackgroundColor to
set Label2 . FontSize to
set Label2 . Text to " Status: nicht berührt! "
0
0
Show Warnings

Anmeldung ohne google-Mail-Konto

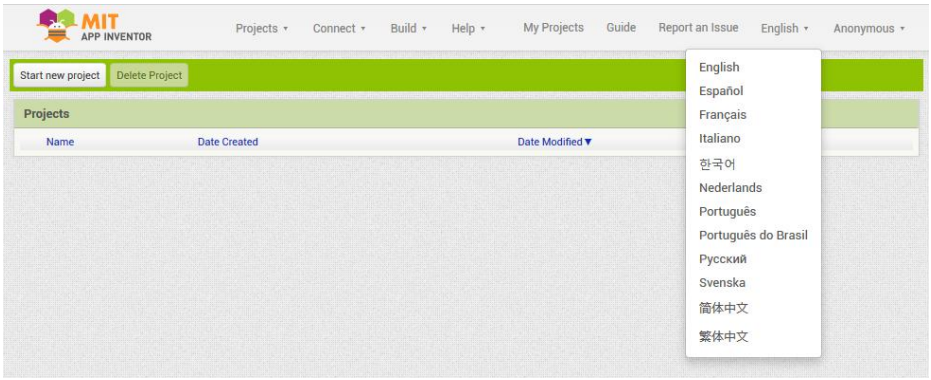
http://code.appinventor.mit.edu/

Code muss man sich notieren, um gegegebenenfalls Projekte fortzusetzen.

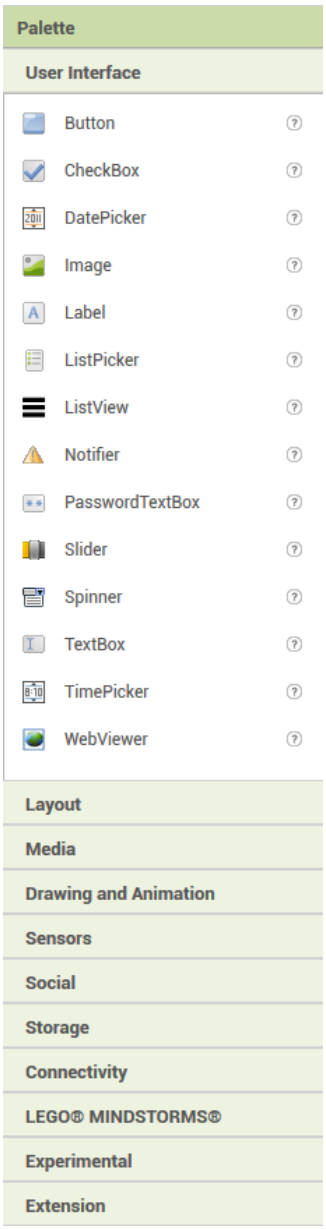


Sprache ändern

Leider ist Deutsch nicht in der Sprach-Auswahl, so dass wir bei Englich bleiben. Vielleicht tut sich ja in der nächsten zeit hier was, dann ist die Umstellung hier machbar.



App-Komponenten und Bedienelemente



Die Palette bietet alle Komponenten. die man in seine App einbauen kann.
Zuerst werden uns die Elemente des "User Interface" – also der Benutzer-Kommunikation bzw. Nutzer-Schnittstelle – interesieren. Unsere App soll ja schließlich was auf dem Display zeigen.
Die anderen Komponenten-Gruppen werden wir bei Bedarf genauer besprechen. Interessant sind sicher die Sensoren, die unser Smartphone beherbergt und die Verbindung zu LEGO ® Mindstorms ®. Da lassen sich dann auch Roboter mittels App steuern.

Für die Bedien-Oberfläche brauchen wir vorrangig Elemente aus dem Bereich "User Interface". Viele der dort angezeigten Elemente sind uns aus anderen Programmen und App's bekannt.
Mit einem Button (dt.: Schaltfläche, Bedienknopf) bekommen wir ein Element auf den Bildschirm, den der Nutzer anklicken / antippen kann. Dabei kann auf der Fläche ein Text stehen oder ein Bild angezeigt werden.
Label's werden für die Beschriftung von Elementen benutzt. Das können Überschriften sein, aber auch Kennzeichnungen für Eingabe-Felder. Schließlich soll ein Nutzer ja auch wissen, wo er etwas eingeben muss.
Die eigentlichen Eingabe-Felder sind Text-Boxen. Auch Zahlen lassen sich hier eingeben.
Zur Vereinfachung von Eingaben gibt es diverse andere Bedien-Elemente. Bei Check-Boxen können wir Optionen einrichten. Die Slider (dt.: Schieberegler) vereinfachen die Zahlen-Eingabe. Jenachdem wo der Schieber innerhalb eines Bereichs steht, wird eine Zahl generiert.
Einfach nur anzuzeigende Bilder oder Graphiken werden mittels "Image" eingebaut.
Die anderen Bedien-Elemente erklären sich z.T. selbst. Einige werden wir in den folgenden Projekten besprechen.
Alle weiteren Elemente, wie abzuspielende Sound's oder Video's werden über die anderen Paletten-Bereich zugänglich.
Mit dem AppInventor können praktisch echte App's entwickelt werden. Bis zu einer App im App-Store ist es dann nicht mehr weit. Hier sind gute Ideen gefragt.

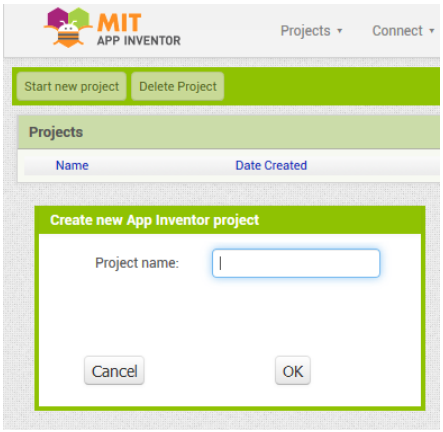
Beispiel-Projekt: Bild-Ton-App

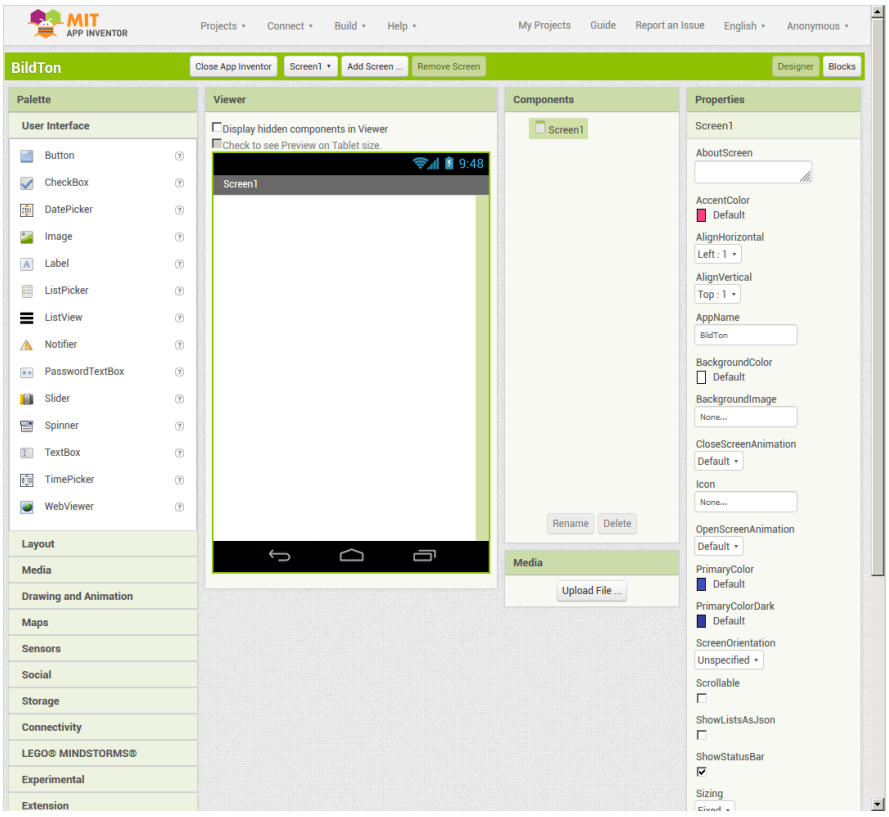
Q: AppCamps (geändert: dre)

Ziel / App-Idee:

In der App soll ein Bild von einem Objekt (z.B. eine Katze) angezeigt werden. Nach dem Antippen soll ein passender Ton (z.B. ein "Miau") abgespielt werden. Es folgt ein anderes Objekt (z.B. ein Hund). Nach dem Tippen auf dieses Bild erfolgt wieder eine passende Tonausgabe und die Rückkehr zum 1. Bild.

Projekt anlegen





Zusammenstellen der Anzeige- und Bedien-Elemente

Dieser Teil ist für Anfänger immer ersteinmal schwierig. Man muss sich in die Bedien-Struktur von App's und Programmen hineindenken.
Das Aussehen unserer App stellen wir uns im "Viewer" zusammen. Man erkennt schon im leeren Screen (dt.: Bildschrim") das typische Aussehen eines Smartphone's oder Tablet's.

Soll etwas als Fläche angeklickt / berührt werden, dann ist das ein sogenannter "Button". Im deutschen sprechen wir auch von Schaltfläche. Button's werden üblicherweise mit einem Text beschriftet. Man kann aber auch Symbole oder Bilder benutzen. Da wir ein Bild antippen wollen, wählen wir also das "Button"-Symbol in der Palette ("User Interface") aus und ziehen es an die gewünschte Position.

Was genau auf dem Button angezeigt werden soll und was dieser später machen soll, stellt man im rechten Bereich "Properties" (dt.: Eigenschaften) ein.
Bei uns soll der Button ein Bild anzeigen, also gehen wir zum Merkmal "Image" und klicken entweder ein vorhandenes Bild an oder laden ein neues hoch ("Upload File…").

Quelle des eingefügten Bildes: bixabay.com

Sollte noch kein eigenes Bild vorhanden sein, dann kann man sich aus freien Quellen ein solches besorgen. Es reichen sehr kleine Bilder (geringe Auflösung / Pixelzahl).
Die von mir verwendeten Bilder stammen z.B. von bixabay.com.
Mit den anderen Eigenschaften kann man ruhig etwas experimentieren. Wenn eine Eigenschaften-Änderung nicht gefällt, dann einfach wieder auf die vorherige Einstellung gehen.
Bei uns ist z.B. sinnvoll den Text zu entfernen. Dazu löscht man einfach den Inhalt in der Button-Eigenschaft "Text".

Nun brauchen wir für den Nutzer noch einen Hinweis auf die Bedienung. Er soll ja das Bild antippen / anklicken.
Das passende Bedien-Element ist ein "Label". Die Beschriftung soll nur angezeigt werden.
Wir klicken also das Label-Symbol im User Interface-Bereich an und ziehen es an die gewünschte Position. Das kann über oder unter dem Button-Bild sein.
Später können wir Beschriftungen auch im Programm-Verlauf ändern.
In der Aufgabe war auch angegeben, einen Sound abzuspielen.

Dazu müssen wir uns auch ein Sound-Objekt mit in das Projekt einfügen.
Einsolches Objekt findet man im Bereich "Media" der Palette.
Auch hier ziehen wir das Sound-Symbol auf die Screen-Fläche.

Wenn keine Sound's oder Musik-Dateien zur Verfügung stehen, dann gibt es auch wieder diverse freie!!! Sounds im Internet. Für meine App habe ich audiyou.de benutzt. Hier ist aber eine Registrierung notwendig.

Programmierung der Funktionalität

Die Programmierung erfolgt Block-orientiert. Um in die Quell-Text-Ansicht zu kommen, nutzt man den Button "Blocks" rechts oben im App Inventor. Mit dem danebenliegenden "Designer" gelangen wir wieder in die Design-Ansicht. Ein Wechsel ist jederzeit möglich. Die Programmierung liegt für den Nutzer unsichtbar hinter dem Design. Deshalb nennt man diesen Bereich auch Backend. Das Fontend ist der sichtbare Teil – also all das, was wir mit dem Designer bearbeiten.
Die Funktionalität eines Bedien-Elementes stellen wir uns mit Hilfe von Blöcken zusammen. Die "Blocks"-Palette links zeigt die verfügbaren Block-Bereiche. Hier kann man ruhig mal alle Gruppen durchklicken, um sich eine Übersicht zu verschaffen.

Block-Gruppen	Farbe	englische Bezeichnung	enthält z.B. …	
Steuerung		Control		
Logik		Logic		
Mathematik		Math		
Text		Text		
Listen		Lists		
Farben		Colors		
Variablen		Variables		
Unterprogramme		Procedures		

Eine gute Startstelle ist immer ein Block aus dem Bereich "Control". Die meisten Funktionen sollen ja erst beginnen, wenn ein Bedien-Element verändert wurde.

Abspeichern des Projekt's

Die eigentliche Arbeit(-Zeit) steckt in unserem Projekt – also praktisch in den Daten des App Inventors. Auch, wenn es sich nur um Test's handelt, man sollte sich gleich von Anfang an daran gewöhnen, seine Daten auf dem eigenen Rechner (bzw. im eigenen Ordner) zu speicher / sichern.

"Projects" "Exported selected project (.aia) to
my computer"

(c,p) 2015 – 2020 lsp: dre

So abgespeicherte Projekte lassen sich dann später auch wieder laden / importieren. Sie können dann erweitert, verbessert oder völlig erneuert werden.

Veröffentlichen und Testen der App

Wir haben zwei grundsätzliche Möglichkeiten unser Produkt zu testen. Einmal können wir uns einen Link generieren lassen. Überdiesen kann jeder die erstellte App testen. Der Link wird vom App Inventor aus QR-Code bereitgestellt.
Zum Testen auf einem android-Smartphone oder –Tablet benötigt man also einen QR-Code-Reader. Eventuell muss dieser zuerst aus einem AppStore geladen werden.
Wer keine Veränderungen an seinem android-Gerät vornehmen möchte, der kann auch einen Test der erstellten App in einer simulierten Umgebung – einem sogenannten Emulator durchführen.

direktes Testen über einen QR-Code

"Build" "App (provide QR code for .apk)"

Testen in einem Emulator

Wir bleiben zum Testen auf dem eigenen Rechner. Das benötigte "Testgerät" wird emuliert – also nachgebildet. Es erscheint als Programm "" auf dem Desktop.
Je nach Betriebssystem muss man dazu eine spezielle Software installieren.

Emulatoren

- **für Windows** Anleitung: http://appinventor.mit.edu/explore/ai2/windows.html
  Installer: http://appinv.us/aisetup_windows
  Achtung: Zur Installation werden Admnistrator-Rechte benötigt!
- **für Mac / iOS** Anleitung: http://appinventor.mit.edu/explore/ai2/mac.html
- **für android** Installation der "MIT AI2 Companion"-App aus dem PlayStore
- **für Linux** Anleitung: http://appinventor.mit.edu/explore/ai2/linux.html

"Build" "App (save .apk to my computer)"

Erweitern und Verbessern der App

Wechseln von zwei Bildern und Wiedergabe des richtigen Sounds zum Bild

Fehler in einer App finden

Das Finden von Fehlern ist natürlich schon für die eigene Programmieung wichtig. Spätestens, wenn die App's etwas komplexer werden, wird man auch Fehler machen. Das ist völlig normal. Die Schwierigkeit liegt darin diese Fehler zu finden. Oft sind es Denk-Fehler von uns. Computer denken eben nicht mit - sie tun exakt nur das, was man ihnen gesagt hat. Ob das Sinn macht oder nicht, können Computer nicht kontrollieren.

Um ein fremdes oder abgespeichertes Projekt wieder zu laden kann man dieses als AIA-Datei in den App Inventor importieren.

Wo stecken typische Fehler?

Einhaltung des Algorithmus

Benennung von Variablen, Funktionen, …

Nicht-Beachtung von Arbeitweisen des Computers

klassisch Abfrage einer Taste

passiert nur einmal, genau hinter der letzten
Anweisung und dem nächsten Befehl

Wenn an dieser Stelle die Taste nicht gedrückt wird, dann übergeht das Programm diese Stelle und macht in jedem Fall mit der nächsten Anweisung weiter

gemeint ist aber oft ein Warten auf einen Tasten-Druck
das muss man dem Computer aber auch genauso sagen

obige Variante ist aber nicht ohne Sinn. Bei langwierigen Berechnungen / Simulationen soll es vielleicht eine Möglichkeit geben diese zu unterbrechen. Jetzt macht eine kurz- bzw. zwischen-zeitige Abfrage Sinn. Würde man hier jedes Mal auf einen echten Tasten-Druck warten, dann könnten sich Simulationen ev. endlos in die Länge ziehen.

Aufgaben:

1. ***Baue die oben beschriebene App nach! Es können andere Bilder und Sounds benutzt werden!***
2. ***Erweitere das programm nun um ein drittes Objekt! Die Objekte sollen immer in der gleichen Reihenfolge hintereinander angezeigt werden!***
3. ***Erstelle eine kleine Quiz-App! Es sollen mindestens drei Fragen gestellt werden (also 3 Screen's). Oben soll die Frage stehen, z.B. "Was ist ein Auto?" Darunter sind zwei Bilder angezeigt, eines mit einem passenden Objekt zur Frage und eine nicht passendes. Beim Antippen des falschen Bildes soll ein passendes Geräusch oder eine kurze (selbst-augenommene) Ansage (z.B: "das ist falsch") kommen. Beim Antippen des richtigen Bildes erfolgt ein postives Feedback und es wird die nächste Frage gestellt!***

Beispiel-Projekt: Glas-Kugel-App

Q: AppCamps (geändert: dre)

Ziel / App-Idee:

Die App soll eine mystische Kugel simulieren, die auf JA-NEIN-Fragen mit einer zufälligen (unbestimmten) Antwort reagiert. Die Ausgabe soll als (künstlicher) gesprochener Text erfolgen.

Projekt anlegen

Design einrichten / Arrangement der Bedien-Elemente

Meist sind es mehrere Bedien-Elemente, die sich den knappen Platz auf dem Bildschirm teilen müssen. Hier gilt es, sich eine Nutzer-freundliche Anordnung auszudenken, die auch alle notwendigen Funktionen abdeckt.
Im Bereich "Layout" gibt es verschiedene Arrangement's.
Die Arrangement's sind wiederum eigen-ständige Programm-Objekte. Sie haben ver-schiedenste eigene Eigenschaften.

Bedien-Elemente umbenennen

Programmierung der Funktionalität

(Antwort-)Listen erstellen und Zufalls-gesteuert verwenden

Listen beginnen im App Inventor bei 1
Das ist ein wichtiger Unterschied zu vielen Programmiersprachen. Es ist aber für An-fänger deutlich einsichtiger bei einer Liste eben wirklich beim ersten – und nicht beim nullten – Element zu beginnen.

bei den Listen-Blöcken gibt es schon einen Block, der die zufällige Auswahl eines Lis-ten-Elementes ermöglicht
will man bestimmte Werte, Texte usw. mehr-fach im Programm benutzen, dann bietet sich immer die Benutzung von Bezeichnern (Variablen, Platzhaltern) an. Das macht auch vor allem dann Sinn, wenn sich dieser Wert an einer bestimmten Stelle auch mal ändern soll. Das Variablen-Konzept kennen wir aus der Mathematik. In der Informatik gibt es da

nur wenige Besonderheiten, auf die wir später noch kurz eingehen.
Hier zeigen ich zuerst die unschöne Lösung ohne Variable. Sie zeigt uns aber sehr schön, wie flexibel und effektiv der Umgang mit Objekten in der Programmierung funktioniert.

die Version mit einer Variablen ist etwas leichter verständlich und später wesentlich einfacher zu erweitern.

mit der Sprach-Ausgabe arbeiten (Text-to-Speech)

"Text-to-Speech"-Element als nicht-sichtbares Objekt

Sprache einstellen

Aufrufen über call-Block

Ausgabe von Ton- / Sprach-Ausgaben hintereinander

Umwandeln einer einfachen Sound-Ausgabe durch ein steuerbares Player-Objekt

Abspielen des zweiten / nachfolgenden Sound's erst nachdem der Player seine Ausgabe beendet hat

Veröffentlichen und Testen

Siehe dazu unter dem vorherigen Projekt (→ Veröffentlichen und Testen der App). Das Verfahren ist immer das Gleiche.

Aufgaben:

1. Entwickle die gerade beschriebene App nach!
2. Erweitere die App um drei weitere Antworten!
3.

Beispiel-Projekt: Mal-App

Q: AppCamps (geändert: dre)

Ziel / App-Idee:

Die App soll es ermöglichen mit der Maus oder dem Finger (Touch) auf dem Bildschirm zu malen. Dabei sollen die Farbe und die Strichstärke veränderbar sein. Ein fertiges Bild soll gespeichert und auch wieder geladen werden können.

Projekt anlegen

Zeichen-Fläche anlegen

Zeichen-Fläche heißt in der Programmierung
Canvas
unter "Drawing and Animation"

volle Breite
automatische Höhe

Button-Leiste / ein Menü erstellen

horizontales Arrangment

mit 4 Button
umbenennen entsprechend der Funktionalität, damit bei der Programmierung ein verständlicher Zugriff möglich ist
Farbe und Form lässt sich entsprechend anpassen

Programmierung der Funktionalität

"when Canvas1.Dragged do"

die in der zweiten Zeile im Block stehenden Begriffe sind Eigenschaften (Attribute) des Canvas-Objektes
Klickt man darauf, dann bekommt man zwei kleine Hilfs-Funktionen angeboten. Dabei handelt sich um einen Getter (Gib-Funktion) und einen Setter (Setz-Funktion) für das Attribut. Attribute von solchen – gekapselten - Objekten lassen sich nicht direkt Lesen oder Schreiben. Dafür muss man die Getter und Setter benutzen.

Button zur Farbauswahl

when Button.Click do

set Canvas.PenColor do

das genauso auch für die anderen beiden Schaltflächen

Zufallsfarbe

wie eben
als Fanktion nimmt man nun make color und dann müssen die Rot-Grün-Blau-Anteile (RGB-Farb-Codierung) als Liste mitgeteilt werden.
Die Werte für die Farb-Anteile dürfen von 0 bis 255 gehen und werden bei uns zufällig bestimmt.
Alle drei Werte auf 0 ergibt schwarz, entsprechend alle 255 würde weiss aussehen. Setz man nur einen Wert auf 255 (und die anderen beiden auf 0), dann erhält man eben die Grund-Farbe.

Link:
http://www.spectrumcolors.de/index.php (Color Lab; Webseite mit vielen Tools zum RGB-Farb-System)

Aufgaben:
1. Realisiere die App bis hier hin!

2. *Füge zwischen der Zeichenfläche und den Button's drei passende Bedien-Elemente ein, die den jeweiligen Farb-Anteil nach einer Zufalls-Farb-Wahl oder des aktuellen Stiftes anzeigen! Überlege Dir, wie Du die Bedien-Elemente arrangieren musst! Die richtige RGB-Farbe sollte für die Beschriftung des jeweiligen Bedien-Elementes auch benutzt werden.*
3.

Radierer programmieren

erste Lösungs-Idee ist sicher wir malen mit Weiss drüber

passt aber nicht, wenn eine andere Hintergrundfarbe gesetzt ist.
Die Hintergrund-Farbe ist abfragbar und wird dann aeinfach als Stift-Farbe für den Pen benutzt

Aufgaben für Freaks:

1. *Die Mal-App soll nun einen Button bekommen, der für das Setzen (vollständiges Neumalen) des Hintergrund's in einer Zufalls-Farbe verantwortlich ist! (Die Zufalls-Zahlen können hier vielleicht nur im Bereich 0 bis 127 benutzt werden?)*
2. *Ergänze einen Button, der eine weiße Grundfläche bewirkt!*
3.

Erweitern und Verbessern der App

Stift-Stärke einstellen

passendes Bedien-Element ist ein Schieberegler (Slider)

der eingestellte Wert muss dann – bei einer Änderung – auf den Zeichen-Stift übertragen werden
when Slider.PositionChanged do

die Stift-Breite ist das Attribut LineWidth und

dieses wird durch Slider.thumpPosition ersetzt

Abspeichern des Bildes

Anlegen einer Datenbank
Bereich "Storage"
hier TinyDB
ist wieder ein nicht-sichtbares Objekt in unserem Programm

Erstellen einer Menü-Leiste für zwei Button zum Speichern und später auch zum Öffnen

Einspeichern in die Datenbank mit call TinyDB.StoreValue
praktisch sind das Kombinationen aus einem Schlüssel(-Wort) (tag) und einem Wert (hier das Bild – also unser Canvas) (value)
call Canvas.SaveAs fileName

praktischerweise müsste natürlich noch eine Abfrage erfolgen, wie das Kunstwerk heißen soll
daruaf verzeichten wir hier und benutzen unsere Datenbank nur mit einem einzigen Eintrag – der dann der letzten abgespeicherten Zeichnung entspricht

Löschen mit Nachfrage

einfaches Löschen (ohne Nachfrage) geht mit call Canvas.Clear

für die Nachfrage brauchen wir einen Dialog (in AI nennt sich das Notifier)

call Notifier.ShowChooseDialog (Auswahl-Dialog)

Button werden mit passenden Antworten belegt
canceable sollte auf false gesetzt werden, wenn der Nutzer sich zwischen den Button's entscheiden soll und eben ein Abbruch nicht möglich ist

die anderen Eigenschaften erklären sich von selbst

Funktionalität über when Notifier.AfterChoosing
einfache Verzweigung (bedingte Abarbeitung) nur dann Löschen, wenn der Nutzer "ja" angeklickt hat

compare texts (Text-Vergleich) zwischen der Auswahl aus dem Notifier (get choise) und der Ziel-Antwort "ja" für Löschen
(nur) dann wird die Ziel-Funktion wirklich ausgeführt

Öffnen eines gespeicherten Bildes

das zu öffnende Bild soll entweder nur angezeigt werden oder vielleicht auch als Vorlege für weitere kreative Aktionen dienen
deshalb ist das "alte" (geöffnete) Bild Hintergrund
nach einem Speichern wird nicht mehr wirklich zwischen Vorder- und Hintergrund unterschieden – es gelten nur noch die Farben der einzelnen Pixel

set Canvas.BackgroundImage to

call TinyDB.GetValue
mit Schlüssel und Wert

bei valueIfTagNotValue dient zur definition eines Ersatz-Wertes, wenn der tag nicht existiert dann bleibt das Programm hier nicht stehen bzw. stürzt ev. ab

zum Zugriff auf mehrere Datenbank-Einträge müssten wir noch eine Abfrage dazwischenschalten, die den Namen (Schlüssel) abfragt
da wir beim Speichern auch schon auf die Möglichkeit mehrerer abzuspeichender Bilder verzichtet haben, kommen wir auch hier ohne aus

Veröffentlichen und Testen

Siehe dazu unter dem ersten Projekt (→ Veröffentlichen und Testen der App). Das Verfahren ist immer das Gleiche.

Beispiel-Projekt: Hund-jagt-Katze-Spiel

Q: AppCamps (geändert: dre)

Ziel / App-Idee:

In der App soll ein Hund mit Hilfe von Schaltflächen über den Bildschirm zu einer Katze dirigiert werden. Wen diese erreicht ist wechselt die Katze ihre Position und muss erneut angesteuert werden. Beim Erreichen der Katze erhält der Spieler einen Punkt. Ingesamt steht nur ein begrenzter Zeitraum (30 s) zur Verfügung.

Projekt anlegen

Zeichen-Fläche und Sprite's anlegen

Sprite's sind kleine Bildchen, die über den Bildschrim bewegt werden können.

aus dem Bereich "Drawing and Animation" nehmen wir dazu die "ImageSprite"s

Steuerungs-Schaltflächen anlegen

TableArrangement mit 3 Spalten und 3 Zeilen

an die Kanten jeweils einen Button für die Richtungssteuerung

Programmierung der Funktionalität

Objekt bewegen auf dem Screen setzt Kenntnisse über die Positions-Angaben bzw. das verwendete Koordinaten-System voraus
der Punkt 0,0 (x-, y-Wert) liegt in der linken, oberen Ecke der Zeichenfläche (Canvas)

Hund bewegen

when Buton.Click do

für den Hoch-Button
set ImageSprite.Y to alter Wert - 10
Subtraktion, weil wir uns ja dem Nullwert annähern
runter ist entsprechend mit einer Erhöhung der y-Koordinate verbunden

Zusammentreffen von Hund und Katze auswerten

when ImageSprite.CollidedWith do

call ImageSprite.MoveTo
zufallsgesteuertes Bewegen der Katze über random-Funktion aus dem "Math"-Bereich

die Ausdehnung (Größe) des Canvas kann auch abgefragt werden
Canvas.Width
Canvas.Height

Punkte zählen

Anzeige des Spielstands über zwei Label; einer für die Beschriftung z.B. "erreichete Punkte:" und das zweite Label für den Punkte-Wert
man könnte zwar auch beides über ein Label realisieren, das macht am aber nicht, angestrebt ist eine Trennung zwischen Layout und Daten
Z.B. könnte es ja sein, dass wir die App in einer spanischen Version ausliefern wollen. Dann ist es einfacher nur an einer Stelle den Label-Text zu ändern als jedes Mal z.B. beim Neu-Belegen nach einer Spielstands-Änderung

dazu wird eine Variable benötigt

initialize global … to

beim Spielstart auf 0 setzen

jedes Mal beim Erreichen der Katze um 1 hochzählen
im Block, wo die Kollision von Hund und Katze schon bearbeitet wird

Ausgabe des aktuellen Wert auf ein Label

Hinweis!: globale Variablen gelten nur innerhalb eines Screen's!

Timer verwenden

unter "Sensors" gibt es Clock-Objekte

die Uhr bekommt einen Start-Wert

Reaktion auf Timer-Ereignis
when Clock.Timer do

z.B. durch Änderung des Label-Textes

Spiel / Runde beenden

eine einfache Lösung besteht darin, die Steuerungs-Bedienelemente auf unsichtbar zu stellen

set TableArrangement.Visible to false

einige Programmiersprachen ermöglichen auch ein Abschalten der Bedien-Elemente, sie sind dann zwar noch sichtbar, aber nicht mehr benutzbar

das besprochene Vorgehen erfordert es die App neu zu starten, um weiter spielen zu können

nächste Möglichkeit ist der Einbau eines Button, der zuerst nicht sichtbar ist, nach Erreichen des Runden-Endes angezeigt wird und alle Attribute usw. auf die Start-Situation zurückstellt

Erweitern und Verbessern der App

Aufgaben:

1. *Lasse den Hund einmal bellen, wenn er die Katze erreicht hat! Auch beim Verschieben der Katze soll ein passender Sound abgespielt werden.*

2. Überlege Dir ein paar Veränderungen am Spiel! Besprich mit anderen Kursteilnehmern, ob sich das umsetzen lässt! Versuche eine mögliche Spiel-Veränderung zu realisieren! Dokumentiere, wie Du vorgegangen bist! Wenn die Spiel-Veränderung nicht möglich ist oder noch nicht funktioniert, dann dokumentiere auch die gemachten Fehler!
3. Stelle Deine Umsetzung der Spiel-Veränderung dem Kurs vor! Sammle Verbesserungs-Vorschläge!
4.

Anzeige des Spielstands

Aufgaben:

1. Lasse Dir neben dem aktuellen Spielstand auch noch die verbleibende Zeit anzeigen!
2. Nun ist die Anzeige des besten Spielstands gewünscht! Immer, wenn ein besseres Ergebnis erreicht wird, soll der wert aktualisiert werden!
3.

Anzeige des Spielstands

Anzeige des Spielstands

Veröffentlichen und Testen

Siehe dazu unter dem ersten Projekt (→ Veröffentlichen und Testen der App). Das Verfahren ist immer das Gleiche.

7.x. multimediale Präsentation

Ergebnisse multimedial präsentieren

kooperativ arbeiten

das Prinzip eines Komprimierungsverfahrens erklären

moderne Datei-Formate als gezippte Daten-Pakete (Container-Datei-Formate)

Umbenennen einer WORD-Datei (*.DOCX) in eine ZIP-Datei
entpacken der Datei bzw. betrachten des Datei-Inhalts mit einem Dekompressions-Programm (z.B. 7zip, WinZIP, WinRAR, …)

gleiches Prinzip bei XLSX und PTTX
bei EXCEL-Dateien tradionell weniger eingebettete Objekte

E. Probleme untersuchen und durch Programmierung lösen

E.x. vom Problem zum Programm

aus einer Problemfrage eine Spezifikation entwickeln

in Abhängigkeit von der Problemfrage
- Sensoren und Aktoren
- algorithmische Strukturen und Datentypen
begründet auswählen und verwenden

Softwareprojekte dokumentieren

(c,p) 2015 – 2020 lsp: dre

E.x.y. App-Entwicklung mit dem App Inventor

Quelle der Screen-Shot's direkt von: http://ai2.appinventor.mit.edu

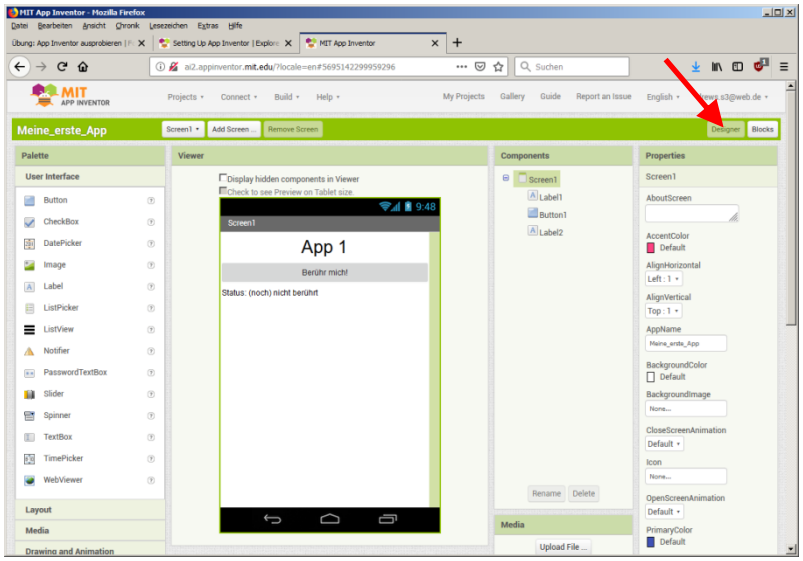
kurz AI oder AI2
→ http://appinventor.mit.edu/

engl. mit einem Touch von deutsch

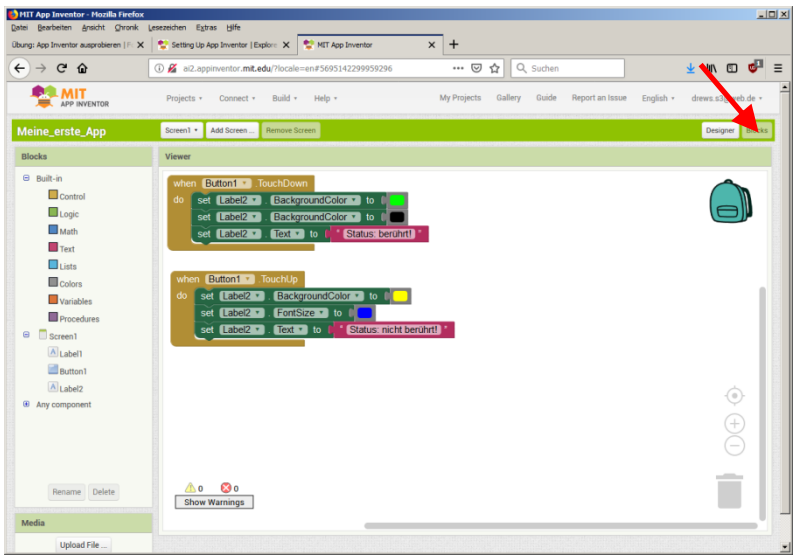
Internet-Explorer funktioniert nicht! Chrome oder Firefox empfohlen

Block-orientierte Programmierung (Snap!-ähnlich)

Design-Bereich



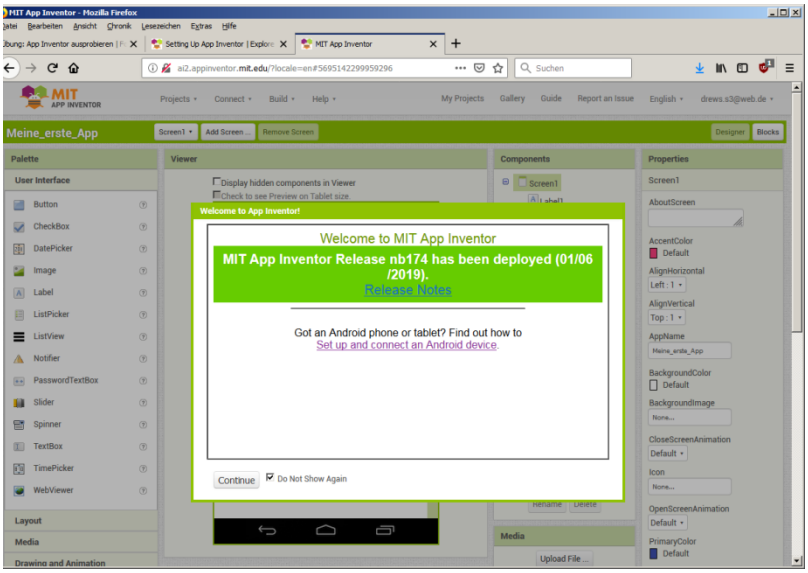
Block- oder Programmier-Bereich



App kann auf Android-Gerät getestet werden
alternativ gibt es einen Emulator, dieser muss aber vorher extra installiert werden

problematisch ist ev. die zwangsweise Anmeldung

es gibt eine installierbare offline-Version
etwas versteckt → http://appinventor.mit.edu/explore/ai2/update-setup-software.html



Start der App-Entwicklung über → http://appinventor.mit.edu/explore/ai2/setup.html
hier muss man sich für das weitere Vorgehen entscheiden

Arbeits-Möglichkeiten

- **Programmierung auf dem PC und Übertragung des Programm's per WLAN auf ein android-Gerät**

  auf dem android muss eine App (AI2 Companion) installiert werden, die die Kommunikation und das Abspielen der selbst programmierten App übernimmt

- **Programmierung und Emulation eines Programm's auf dem PC**

  mit der Setup-Datei wird ein installierbarer Emulator bereitgestellt

- **Programmierung auf dem PC und Übertragung des Programm's per USB-Kabel auf ein android-Gerät**

  USB-Verbindungs-Software in der Setup-Datei enthalten
  android-Gerät muss über Daten-fähiges USB-Kabel (*??? OTG*) am PC angeschlossen werden

Links:
http://schuelerlabor.informatik.rwth-aachen.de/modulmaterialien/erste-app (Modulhandbuch zur Installation und Programmierung mit AI)

https://appcamps.de/unterrichtsmaterial/app-entwicklung/ (App-Entwicklung mit AI auf AppCamps)

8. Sicher kommunizieren

Verschlüsselung verstehen

Die SuS beschreiben klassische Verfahren unter Verwendung der Begriffe Klartext- und Geheimtextalphabet, Klartext und Geheimtext, Schlüssel, Verschlüsseln und Entschlüsseln. Sie erkennen, dass durch Verschlüsselung eine vertraulische Kommunikation möglich wird.

- *klassische Verfahren der symmetrischen Verschlüsselung anschaulich erläutern*
- *kurze Nachrichten verschlüsselt austauschen*

Die SuS argumentieren zur Sicherheit der Verfahren.

Vertraulichkeit herstellen

- *Möglichkeiten der Verschlüsselung in Standardprogrammen nutzen*
- *verschlüsselte Daten als E-Mail-Anhang senden*
- *Beispiele für verschlüsselte und unverschlüsselte Kommunikation nennen und eigene Kommunikation bewerten*

Die SuS kennen Indikatoren für die Verwendung verschlüsselter Kommunikationsprotokolle (z.B. gesicherter WLAN, https) bzw. informieren sich in geeigneten Quellen über die Sicherheitsmerkmale einer Kommunikation.

- *Merkmale sicherer Kennwörter begründen und sichere Kennwörter verwenden*

Bei der Wahl sicherer Kennwörter beachten die SuS sowohl technische als auch psychologische Aspekte.

Vorschlag zur inhaltlichen Vertiefung:
Nutzung eigenständiger Programme für die Verschlüsselung und Komprimierung.

Problem-Fragen für Selbstorganisiertes Lernen

Verschlüsselung ist was für Millitärs! So was brauch' ich nicht! Ich habe keine Geheimnisse? Wahr oder falsch?

Was ist der Unterschied zwischen Codierung und Chiffrierung?
Können wir ohne Codierung und Chiffrierung auskommen?
Gehen Codierung und Chiffrierung heute nur noch mit Computern?
Mit welchen prinzipiellen Methoden lassen sich Nachrichten verschlüsseln?
Funktionieren diese Methoden auch für Codierungen?

Was sagt uns der Schlüssel oder das Schloß in der Adress-Zeile des Browsers?

Was hat CÄSAR mit Verschlüsselung zu tun?

Wo wird heute Verschlüsselung eingesetzt?

Was machen Kryptologen?
Verschlüsseln mit einem Schlüssel und entschlüsseln mit einem ganz anderen – geht das?
Was ist ein "Brute force"-Angriff?
Kann man jede verschlüsselte Nachricht entschlüsseln?

Ist jede Verschlüsselung sicher?
Welches Verschlüsselungs-Verfahren ist unknackbar?
Gibt es Verschlüsselungs-Verfahren, die öffentlich bekannt sind und die trotzdem sicher sind? Wie geht das?

Kann man jede verschlüsselte Nachricht, doch irgendwie entschlüsseln?

Kann man heute mit einer Enigma-ähnlichen Maschine sichere Nachrichten verschicken?

klassische Verfahren der symmetrischen Verschlüsselung erläutern and anwenden

Softwarelösungen zur Verschlüsselung nutzen

Merkmale sicherer Passwörter begründen

8.0. Grundlagen

8.0.1. Kommunikation

Man kann nicht nicht kommunizieren.
Paul WATZLAWICK

bedeutet eigentlich Verbindung, Zusammenhang (lat.: communicatio = Mitteilung, Benachrichtigung)
im Zusammenhang mit der Verbreitung und Aufnahme von Signalen

Austausch und Übertragung von Informationen

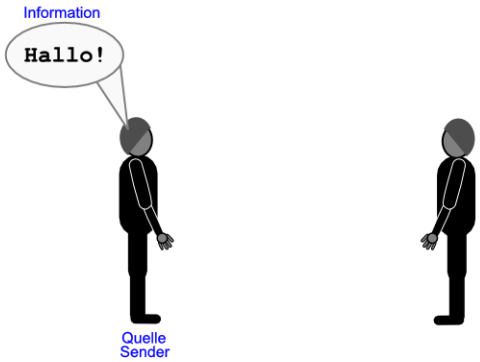
im modenen Sinn und unter Beachtung der Entwicklungen in der Kommunikations-Technologie auch als wechselseitige Steuerung von zwei Kommunikanten

Definition(en): Kommunikation
Kommunikation ist der Austausch von Nachrichten (Informationen) (zwischen zwei Kommunikatoren).
Kommunikation ist der Prozess der Übertragung von Nachrichten zwischen einem Sender und einem oder mehreren Empfängern.

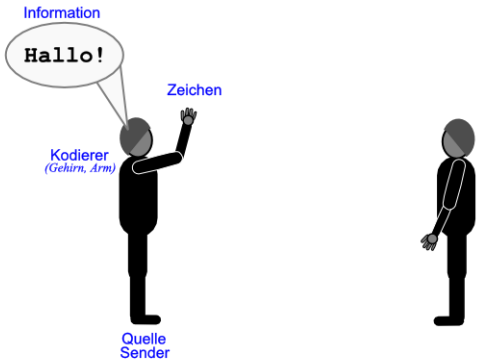
8.0.1.1. ein einfaches Kommunikations-Modell

Gehen wir von einem einfachen Beispiel aus:

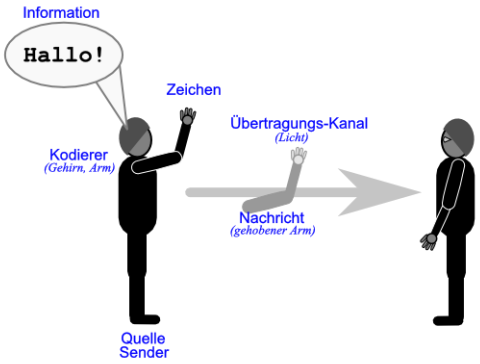
Eine Person möchte der anderen "Hallo!" sagen.



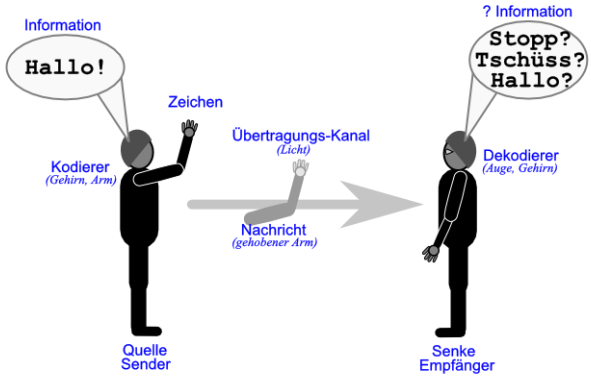
Das "Hallo!" ist die Information, die der (Nachrichten-)Sender an den (Nachrichten-)Empänger weitergeben will. Die Informations-verbreitende Person wird auch als Informations-Quelle bezeichnet. Die Begriffe Quelle und Sender werden im Bereich der Kommunikation synonym benutzt.
Die Information befindet sich im Augenblick nur im Gehirn (Speicher, Register) des Senders.
Damit die Information nun weitergeben werden kann, muss sie in für Zeichen umgesetzt werden. Z.B. kann dazu ein erhobener Arm benutzt werden. Die Übersetzung der Information "Hallo!" erfolgt über Gehirn und Arm-Muskelatur in das Zeichen "gehobener Arm".
Ein oder mehrere Zeichen werden auch als Signal bezeichnet.
Wichtig ist auch, dass der Sender ein Zeichen benutzt, was geeignet ist die gewünscht Information zu übertragen. Ein nach vorn gestrecktes Bein wird sicher nicht als "Hallo!" gedeutet werden.

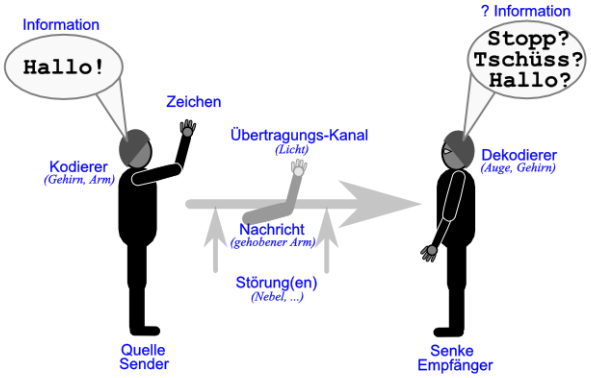


Trotzdem wäre auch das ein mögliches Zeichen in einer Kommunikation.
Der gehobene Arm ist die Nachricht, die übertragen wird. In unserem Fall ist Licht der genutzte Übertragungs-Kanal. Oft wird auch nur kurz vom Kanal gesprochen.
Wichtig ist in diesem Zusammenhang, dass hier nicht die Übertragung des Armes erfolgt, sondern eben nur veränderte Licht-Situationen.
Der Empfänger nimmt die Nachricht auf. Dazu muss er über ein geeignetes Empfangs- und Dekodier-System verfügen. Bei optischen Signalen wird das sicher das Auge sein.

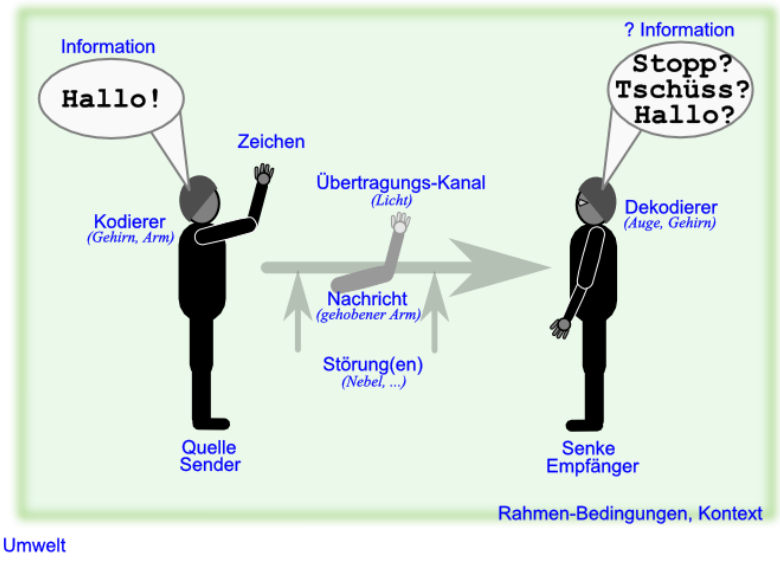


Die optischen Informationen werden von Auge und Gehirn verarbeitet und in eine Information umgesetzt.
Dazu muss der Dekodierer die gesendete Nachricht in seinem Verständnis-Bereich haben. Wenn er das Signal nicht versteht, dann wird eine Übersetzung / Dekodierung nicht zum Ziel führen oder ein falsches Ergebnis liefern.
Aber schon ein bekanntes Zeichen muss nicht zwangsläufig eine bestimmte Information beim Empfänger erzeugen. So könnte der Empfänger die Nachricht "gehobener Arm" auch als "Stopp!" oder "Tschüss!" übersetzen.
Die Übertragung des Signals "gehobener Arm" über das Licht unterliegt vielen störenden Einflüssen. Vielleicht sind beide Kommunikatoren sehr weit voneinander entfernt, so dass der gehobene Arm gar nicht so deutlich zu sehen ist. Oder es herscht Nebel. Je nach Übertagungs-Kanal können viele Faktoren störend wirken und die Nachricht verrauschen – wie der Kommunikations-Wissenschaftler sagt.
Für die Interpretation der Nachricht "gehobener Arm" nutzt unser Empfänger Informationen aus seiner Umgebung.





Hat er die andere Person heute zum ersten Mal gesehen? Kennt er die Person? Besteht eine Situation zur Begrüßung usw. usf.?
Diese Rahmen-Bedingungen beeinflussen das Interpretations-Ergebnis sehr stark. Man nennt sie kurz auch Kontext.
Somit haben wir ein Modell für die einfache Kommunikation zwischen zwei Partnern.
In einer verallgemeinerten Form ohne konkrete Kommunikanten und Signale usw. stammt das Modell von Claude E. SHANNON und Warren WEAVER. Es entstand in den 40iger Jahren des letzten Jahrhunderts.
Nach SHANNON und WEAVER sind bei einer Kommunkation immer 6 Elemente enthalten / notwendig:

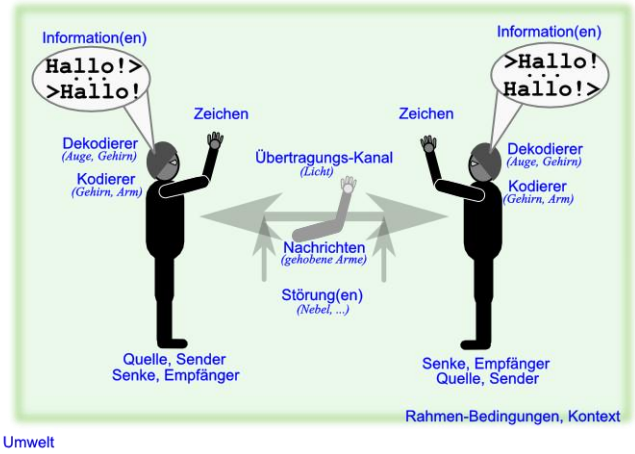


Informations-Quelle / Sender
Verschlüsselung / Kodierer
Nachricht (Signal)
Kanal
Entschlüsselung / Dekodierer
Empfänger / Informations-Senke

nicht zu verhindern sind:
Störungen

Die Störungen selbst gehören aber nicht zur Kommunikation dazu, deshalb fehlen sie in den meisten Abbildungen. Wir können

Der Empfänger wird in irgendeiner Form mit dem Sender kommunizieren. Wie wir oben gehört haben, kann man nicht nicht-kommunizieren. Auch wenn der Empfänger nicht reagiert ist das schon eine Nachricht, die bestimmte Informationen beim ursprünglichen Sender hervorruft.

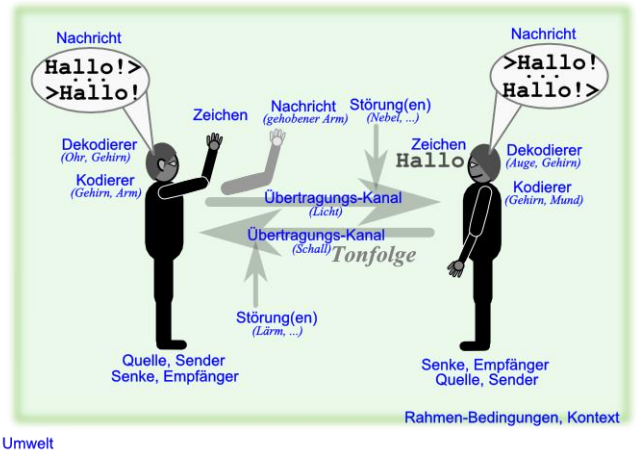


Vielleicht wir der Empfänger als Antwort aber mit dem gleichen Signal antworten, weil er sich bei seiner Interpretation für ein "Hallo!" entschieden hat.

Eine solche wechselseitige oder beidseitige Kommunikation ist dadurch gekennzeichnet, dass beide Partner nun Sender und Empfänger sind.

Natürlich kodieren und dekodieren sie die Informationen und Nachrichten auch wechselseitig. Kodierer (Gehirn, Arm) und Dekodierer (Auge, Gehirn) sind bei beiden Kommunikanten gleich.

Das muss aber nicht sein. Die Kommunikanten können auch unterschiedliche Übertragungs-Kanäle oder unterschiedliche Zeichen(-Sätze) für den Nachrichten-Austausch benutzen.



Z.B. könnte der ursprüngliche Empfänger als Antwort einen Handschlag anbieten.

Nutzt er als Antwort-Kanal den Schall, dann erfolgt die Übertragung der Information als Tonfolge.

Auch ene parallele Kommunikation wäre denkbar. Der Empfänger antwortet zugleich mit einem erhobenem Arm und einem gesprochenem "Hallo!".

Dann ist die Antwort sehr sicher. Man sagt in der Kommunikations-Technik, die Nachricht ist redundant.

Aufgaben:

1. ***Erstellen Sie ein Schema, in dem eine kurze Kommunikation zwischen zwei Personen (nach einem Hallo) dargestellt wird! (Die Auswahl des Übertragungs-Kanals und des Signals ist Ihnen freigestellt!)***
2. ***Wählen Sie eine Kommunikation, bei der maximal eine Person vorkommt! Mindestens ein Kommunikant ist also ein technisches Gerät. Stellen Sie für die Kommunikation zwischen diesen beiden Partner ein beschriftetes Schema auf!***
3. ***Entwickeln Sie ein allgemeines Schema einer einseitigen oder wechselseitigen Kommunikation!***

8.0.1.2. Arten / Einteilung der Kommunikation:

nach der Anzahl der Empfänger:
- Individual-Kommunikation
- Massen-Kommunikation

nach der Richtung des Kommunikations-Flusses:
- einseitige Kommunikation
- beidseitige / wechselseitige Kommunikation

nach der zeitlichen Abfolge der Kommunikation:
- synchrone Kommunikation
- asynchrone Kommunikation

- Simplex
- Halb-Duplex
- Voll-Duplex

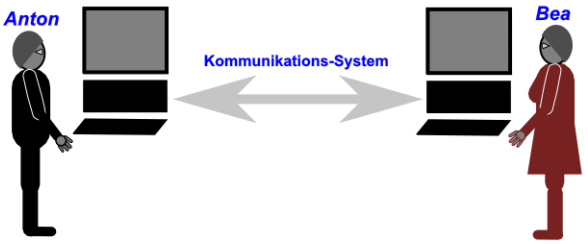
nach der organisatorischen / hierrachischen Strukturen:
- horizontale Kommunikation
- vertikale Kommunikation

nach Art des Kontextes
- gegebener Kontext
- offene Fragestellung
- eindeutige Fragestellung
- ohne menschliche Kommunikanten

Wer redet über was? Unterschiedliche Themen und Befindlichkeiten.
Informations-Gehalt unklar

Rolle des Kommunikations-Systems unklar

Kommunikanten sprechen wahrscheinlich verschiedene "Sprachen"
unterschiedliche Interpretation der Nachrichten

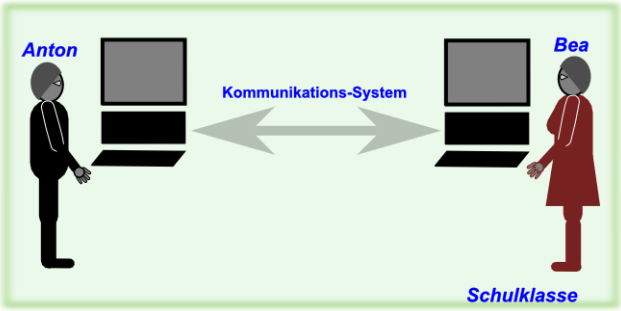


Kommunikation ohne Kontext /
Kommunikation mit offener Fragestellung

klare Bedeutung der Nachrichten
Informationsgehalt hoch, aber nicht eindeutig

Informations-System ist nebensächlich, praktische eine Möglichkeit; Werkzeug-Charakter
Daten-Inhalt spielt keine Rolle für das Kommunikations-System

Kommunikanten sprechen eine "Sprache"
relativ ähnliche Interpretation der Nachrichten

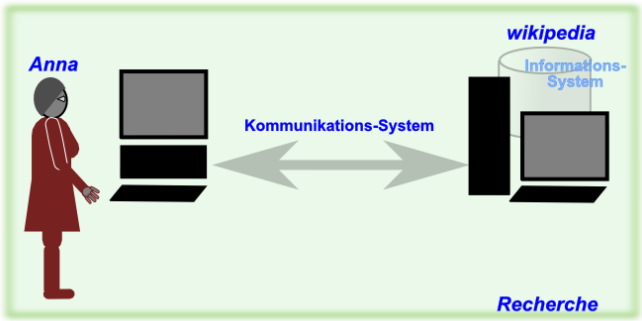


Kommunikation mit Kontext /
Kommunikation mit klarerr Fragestellung

definierte Bedeutung der Nachricht

Teile des Kommunikations-System sind direkt für Kommunikation notwendig
ein Kommunikant

Teile der Daten (üblicherweise die Antworten) werden von Maschine generiert
technischer Kommunikant hat auch Beziehung zum Daten-Inhalt



Mensch-Maschine-Kommunikation /
Kommunikation mit eindeutiger Fragestellung

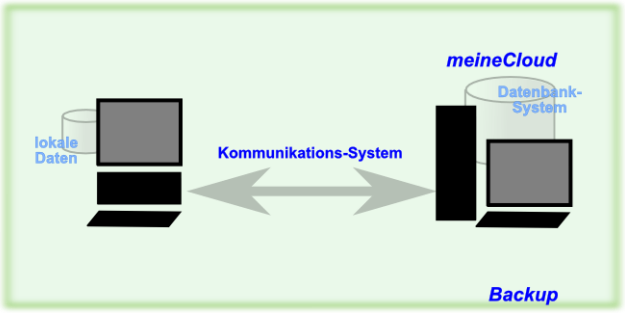
definierte Sprache
zur Bedienung des Kommunikations-System ist auf menschlicher Seite eine Schrittfolge wichtig, aber nicht immer zwingend
Maschine arbeitet Algorithmen ab
Maschine interpretiert eindeutig, beim Menschen bestehen nur wenige Interpreations-Möglichkeiten
Maschine erweckt den Eindruck "intelligent" zusein und menschlich kommunizieren zu können. Aber sachlich bewegt sich die Kommunikation nur in dem Rahmen, den die Programmierer bzw. Datenbank-Designer vorgesehen haben

automatisierte Kommunikation

beide Kommunikanten sind Maschinen
absolut notwendig für Kommunikation

Quelle erzeugt technische Daten, die das Ziel in gleicher Form empfängt

eineindeutige Sprache
beide Kommunikanten arbeiten Algorithmen ab



technische Kommunikation /
Maschine-Maschine-Kommunikation
Kommunikation ohne menschliche Interaktion

Daten-Austausch und "-Interpretation" im Rahmen der vorprogrammieren Rahmen-Bedingungen

8.0.2. Kooperation

(c,p) 2015 – 2020 lsp: dre

8.1. Codierung und Chiffrierung

8.1.1. Codierung im Alltag

Ampel

Memory mit Ziffern und Mengen

Anlaut-Tabelle
Buchstaben-Tabellen (engl. Aussprache)

elektrisches Test-Spiel mit rückseitiger Verkabelung der Zuordnungen

GTIN 8 / 13 und EAN
Global Trade Item Number mit 8, 13 oder 14 Stellen
oft als Ziffern-Folge und Bar-Code auf den Artikeln im Handel

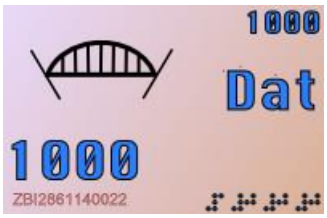
Steifen codieren Ziffern
Rand-Streifen dienen auch zur Eichung der Streifenbreite für einen Scan
eine Ziffer wird über 3 Streifen codiert
insgesamt 4 Streifen notwendig, um auch nächstes Zeichen abzugrenzen
lange Randstreifen haben Code: schwarz_1, weiß_1, schwarz_1, weiß_1

um die richtige Ziffern-Folge (von links nach rechts) zu erhalten muss linker und rechter Teil erkannt werden
würde prinzipiell über unterschiedliche Anzahl Ziffern links und rechts von der Mittel-Kennung gehen
oder es werden zwei Codes für eine Ziffer benutzt, einer für die linke (also beginnende) Seite und eine für rechte (also endende) Seite

QR-Code

8.1.2. Codierung von Mengen und Zahlen

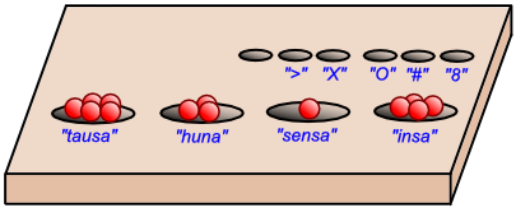
In der Republik Informatika – dem digitalen Staat mitten auf dem analogen Ozean – gibt es eine eigene Währung – das Dat. Weil die Informatikaner schon immer etwas anders waren, als der Rest der Welt, haben sie sich ein eigenes Geldschein-System (Stückelung) ausgedacht. Die gültigen Geldscheine sind in der Abbildung rechts zu sehen. Vorder- und Rückseiten sehen exakt gleich aus.



Aufgaben:

1. *Wieviele Geldscheine brauchst man, um einen Betrag von 175 Dat zu bezahlen? Wieviele braucht man für einen gleichen Euro-Betrag (gesucht ist die geringstmögliche Zahl?*
2. *Die Serien-Nummern der Banknoten enthalten ein Sicherheits-Merkmal. Findest Du es? Wie wird es ermittelt / berechnet?*
3. *Beurteile die Qualität dieses Sicherheits-Merkmals!*
4. *Der 100-Dat-Schein wird besonders gern gefälscht. Erkenne sichere Zeichen für eine Fälschung!*

Bei der letzten galaktischen Schlacht gelang es aus einem Bevölkerungs-Transporter das nebenstehend abgebildete Mulden-Brett aus der dortigen Schule zu erobern. Nun will man raus bekommen, wie die Pasopaner zählen. Von den Pasopanern ist z.B. bekannt, dass sie zwei Köpfe und Hände haben. Jede Hand ist mit drei Fingern ausgestattet, die ihnen ein sicheres Greifen ermöglichen.



Wie könnte das Zahlen-System der Pasopaner funktionieren?

Definition(en): Kodierung
Unter Kodierung versteht man die eindeutige Umsetzung einer Sprache / eines Alphabet's / einer Gruppe von Symbolen in eine andere Sprache / ein anderes Alphabet / eine andere Gruppe von Symbolen. (Die Umsetzung muss nicht Zeichen (/ Symbol) zu Zeichen (/ Symbol) sein!)
Kodierung ist die (eindeutige) Umsetzung-Vorschrift für eine Sprache in eine andere.
Kodierung ist die Darstellung von Daten nach bestimmten, allgemeingültigen Regeln.

Definition(en): Dekodierung
Unter Dekodierung versteht man die Rekonstruktion der ursprünglichen Sprache / des Al-

phabetes / der Symbol-Gruppe aus der kodierten Sprache / dem Alphabet / der Symbol-Gruppe.
Dekodierung ist die (eindeutige) Rück-Umsetzungs-Vorschrift der kodierten Sprache in die ursprüngliche.
Die Dekodierung ist die Umkehrung der Kodierung mittels des gleichen oder eines anderen Verfahrens.

8.1.3. Codierung von Buchstaben und anderen alphanummerischen Zeichen

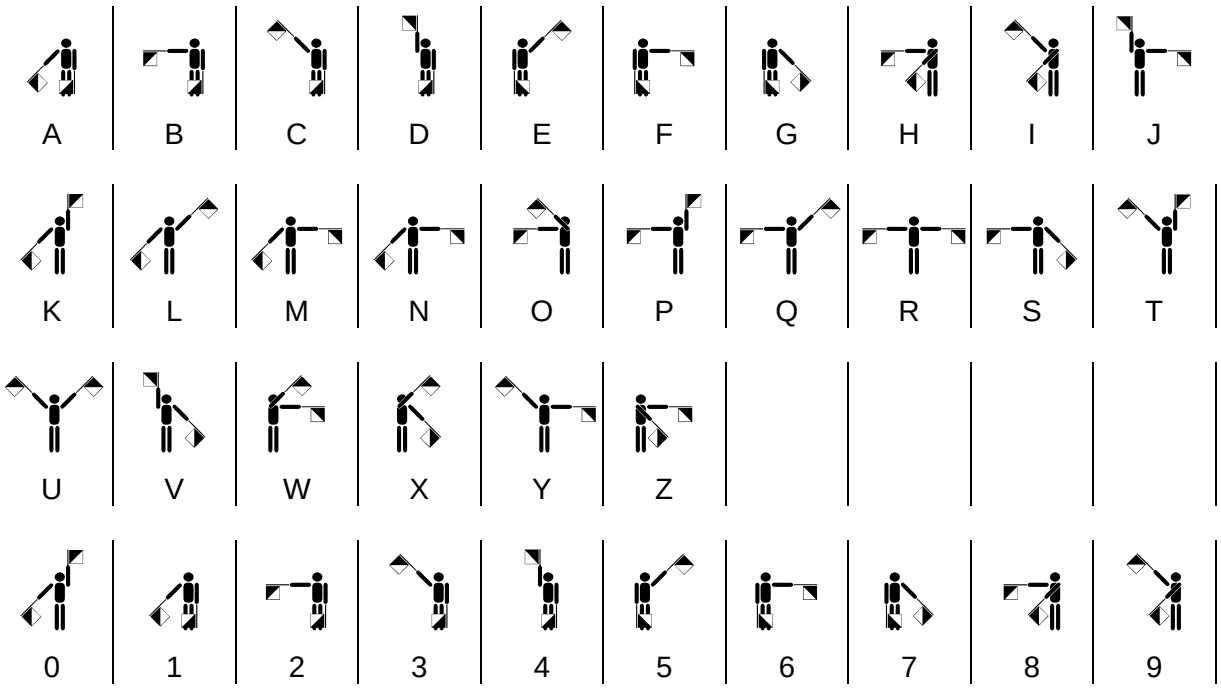
Im Alter von 43 Jahren erlitt der Chefredakteur der Zeitschrift "Elle" Jean-Dominique BAUBY eine Schlaganfall. Infolge dessen konnte er sich überhaupt nicht mehr bewegen. Auch das Sprechen war nicht mehr möglich. Als einzige Kommunikations-Möglichkeit blieb ihm, mit dem linken Auge zu blinzeln. Man spricht vom "Locked-in-Syndrom". Seine Logopädin Henriette DURAND führte ein einfaches Kommunikations-System ein. Eine Hilfsperson sagte immer langsam ein Alphabet auf. Beim richtigen Buchstaben blinzelte Jean-Dominique einmal. Auf diese Weise schrieb BAUBY auch ein Buch ("Schmetterling und Taucherglocke" ("Le scaphandre et le papillon")).

Aufgaben:

1. ***Denke Dir einen einfachen kurzen Satz aus! Übertrage diesen Satz an Deinen Nachbarn, indem dieser immer das Alphabet aufsagt und Du beim richtigen Buchstaben blinzelst!***
2. ***Recherchiere, was es mit dem Buchtitel "Schmetterling und Taucherglocke" auf sich hat!***
3. ***Welche Möglichkeiten gibt es die Kommunikation mit der Hilfsperson zu verbessern (Schnelligkeit und Korrektheit der "übertragenen" Buchstaben)? Erkläre Deine Verbesserungs-Vorschläge!***

Flaggen-Alphabet / Textübertragung mittels Signal-Flaggen (in der Marine)

Noch heute kommunizieren Marine-Schiffe mittels mehrerer Flaggen-Alphabete. Dabei macht man sich zunutze, dass feindliche Schiffe die Signal-Übertragung nur schwer einsehen und mitlesen können. Sie müssten in Sichtweite und praktisch fast immer zwischen den beiden kummunizierenden Schiffen liegen.
Grundsätzlich gibt es zwei unterschiedliche Flaggen-Alphabete. Das eine wird traditionell zur Kommunikation zwischen zwei Kriegsschiffen benutzt. Ziel ist die Übertragung von Texten (Befehlen). Die speziell ausgebildeten Signal-Gasten – so heißen die Flaggen-schwenkenden Matrosen – winken quasi den Text. Oft wird hier auch vom Winker-Alphabet gesprochen.



Das zweite System ist das internationale Signal-Flaggen-Alphabet. Mit ihm werden nur kurze Informationen oder Signale beschrieben. Diese Flaggen verbleiben länger am Mast. Jedes beliebige Mitglied der Schiffs-Besatzung kann die Flagge auf Befehl des Kapitän setzen. Jedes Schiff muss so einen Satz Flaggen in internationalen Gewässern mitführen. Das Internationale Flaggen-Alphabet besteht aus Flaggen für Buchstaben und Ziffern. Passend dazu gibt es ein Buchstabier-Wörter-Satz.

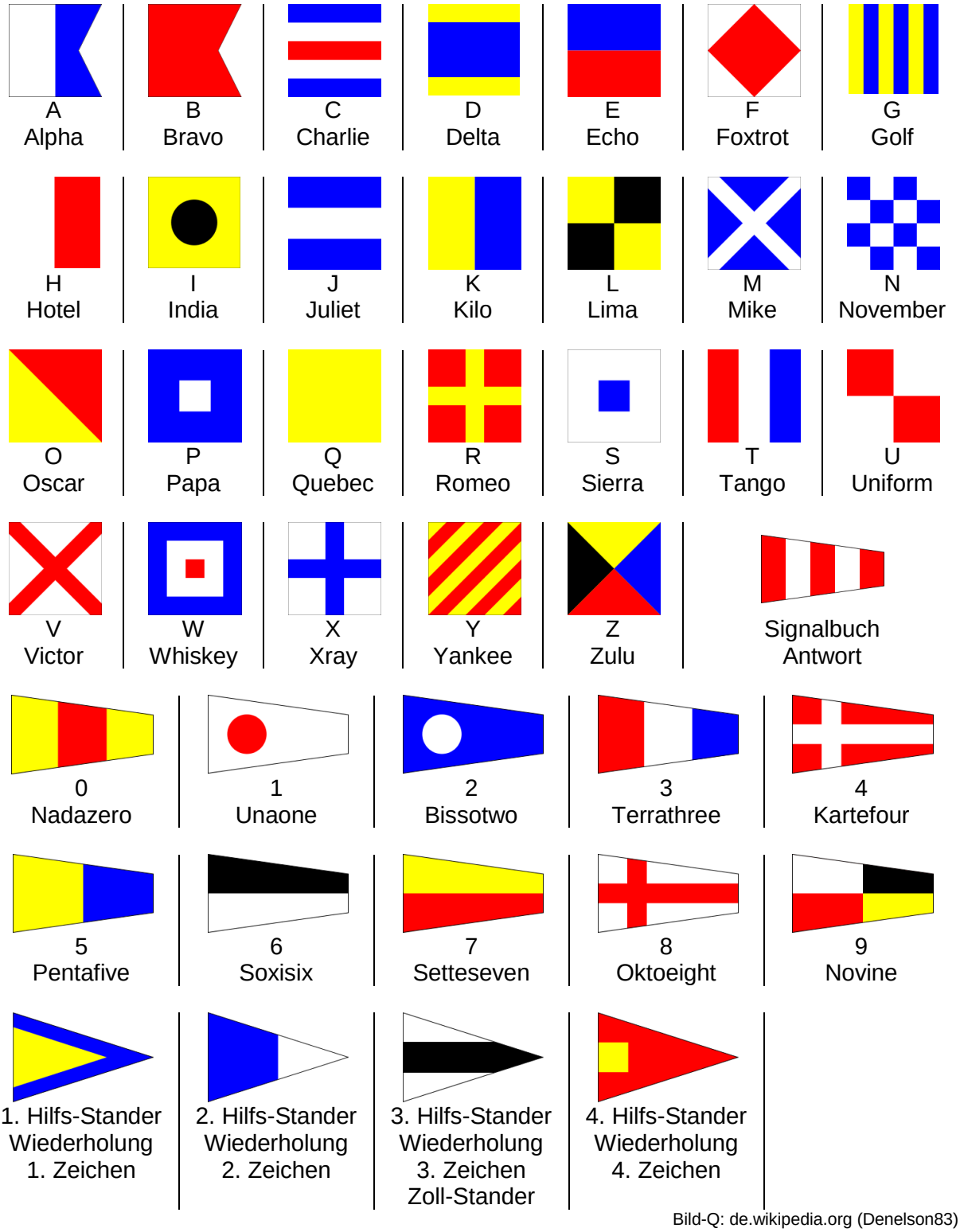


Bild-Q: de.wikipedia.org (Denelson83)

Heute werden diese Flaggen gerne auch mal zur Ausgestaltung eines Schiffes beim Anlegen in einem Hafen oder als festlicher Schmuck benutzt.

Aufgaben:

1. ***Welche Aussage steckt hinter dieser Beflaggung!***

   a) ***b)*** ***c)***

2. ***Einzelne Flaggen haben auch international vereinbarte Informationen. Was signalisieren die Schiffe mit den folgenden Flaggensignalen?***

   a) ***b)*** ***c)***

3.

das MORSE-Alphabet – Signal-Übertragung per Blinklampe, Telegraph oder Funk

die Arm-Telegraphen

Claude CHAPPE (1763 – 1805) entwickelte mit seinen Brüdern einen optischen Telegraphen (1791) und baute eine 225 km lange Telegraphenleitung von Paris nach Lille auf. Er benötigte dazu nur 22 Relaisstatione (Zwischen-Stationen). Die Stationen waren mit kleinen Türmen / Gebäuden ausgetattet an deren Spitze ein zwei-armiges Brett-Gestell montiert war. Über die Stellung der Arme wurde jeweils ein Buchstabe codiert..

Aufgaben:

1. ***Informiere Dich über drei der nachfolgenden Text-Übertragungs-Verfahren!***
   a)
   b)
2. ***Überlege Dir mit Deinem Nachbarn ein Chiffrierungs-System für die Ziffern 0 bis 9! Denke Dir eine sechs-ziffrige Zahl aus und übertrage diese an Deinen Nachbarn!***
3.

8.1.4. Codierung zur Erzeugung "geheimer" Nachrichten

Die Verwendung weniger oder unbekannter Codier-Tabellen erzeugt bei vielen Lesern den Eindruck, es handele sich um eine geheime Nachricht. Ein wenig ist das auch so, aber in der Praxis bleiben solche "Verschlüsselungen" einfach nur Codierungen. Meist sind sie ganz leicht mit einer Analyse der Buchstaben-Häufigkeit "zu knacken".
Wie effektiv schon Buchstaben oder Symbole aus anderen Sprache sein können, vermitteln vielleicht die nachfolgenden Beispiele:

ПӠƏr ЄҒs PЄəҒЃжӠєЃ ЪЯєє ЄЯє TЃrЃҒЄєҒSSЃ ӠЃяЃжѕжЯГЃєё
Αυξη μιτ Ξοδιερυνγ καννμαν Γεηειμνισσε υεβερτραγεν.
..ʜəɵ ʛɞɧ ˌɢə˞ɞɛɥʜɢɜ˞ ɸɗɢɢ ʛɗɢ ˛ɛɵɛɞʛɢɞɦɦɛ ʜɛɵɛɥɧɥɗɜ˞ɛɢ˯
ˇجﭦﭤ ﭤﭤج ˉﭢﭙﭤﭤﭼﭼﭺﭤ ﭢﭙﭢﭢ ﭤﭙﭢ ﭤﭤﭤﭤﭢﭢﭤﭼﭼﭤ ﭺﭤﭦﭤﭼﭼﭙﭤﭤﭢ˛

Obwohl es sich immer um den gleichen Satz handelt, ist er schon schwerer zu decodieren. Es wurde einfach nur eine andere Schrift-Art genutzt. Setzt man den Text wieder auf eine klassische lateinische Schrift-Art, dann ist der Text leicht zu lesen. Für den normalen Nutzer ist aber schon schwierig den Text zu dekodieren. Das liegt auch daran, dass wir die Zeichen nicht so einfach auf einer Tastatur eingeben können. Mit viel Probieren bekommt man es hin. Wer "ausländische" Zeichen öfter braucht, der kann sie sich z.B. mit Wasser-festen Stiften

auf einer hellen Tastatur notieren oder man legt sich Tastatur-Schablonen an. Für Schreibmaschinen-Tast-Kurse gibt es auch "blinde" – also unbeschriftetet – Tastaturen.
Ansonsten müssen wir uns eine Dekodier-Tabelle mit den lateinischen Zeichen und vielleicht auch mit den Ziffern erstellen und die Zeichen aus den anderen Schrift-Arten dagegen setzen.

✌◆♍♒ ❍♓♦ ✍□♎♓♏❒◆■♑ &♋■■ ❍♋■ ☝♏♒♏♓❍■♓♦♦♏ ◆♏♌♏❒♦ ❒♋♑♏■✍
↳▶⇨↑ ↘↓◀ ↬↕⇦↓⇨△▶⇔→ ↗⇨⇔⇔ ↘⇨⇔ ↧↾⇨↑⇨↓↘⇔↓▽▽⇨ ▶⇨⇦⇨△◀△ ⇨→⇨⇔↘

Aufgaben:

1. ***Gebe in einem Textverarbeitungs-Programm 5 kurze Sätze zu Deiner Person ein! (z.B. Name, Alter, Hobby's usw.; ruhig in abgewandelter Reihenfolge)***
   Formatiere den Text mittels Schriftart "Wingdings" auf die Schriftgröße 14 und drucke ihn einmal aus!
   Versuche den Text selbst zu lesen!
2. ***Alle Texte werden nun eingesammelt und wild gewürfelt wieder ausgeteilt! Versuche den Urheber des Ausdrucks zu ermitteln, den Du erhalten hast!***

Arial	Wingdings	Wingdings 3	WP Cyriilic A	Webdings	Sgreeck Fixed	WP Cyriilic A	Wingdings 2	Symbol	WP Phonetic	WP Arabic Script Sihafa					
A	✌	↵	П		A	П	☚	A	˸	˅	A	A	A	A	A
B	✍	↵	п		B	п	☛	B	˹	-	B	B	B	B	B
C	☝	↵	Р		Ξ	Р	☞	X	˻	ˍ	C	C	C	C	C
D	☟	↵	р		Δ	р	☜	Δ	˼	ˎ	D	D	D	D	D
E	☜	↵	С		E	С	☝	E	˽	ˏ	E	E	E	E	E
F	☞	↵	с		Φ	с	☟	Φ	˾	ː	F	F	F	F	F
G	☝	↵	Т		Γ	Т	✌	Γ	˿	ˑ	G	G	G	G	G
H	✌	↵	т		H	т	✋	H	̀	˒	H	H	H	H	H
I	✋	↵	У		I	У	✊	I	́	˓	I	I	I	I	I
J	☺	↵	у		ϑ	у	☜	ϑ	̂	۴	J	J	J	J	J
K	😐	↵	Ф		K	Ф	☞	K	̃	۳	K	K	K	K	K
L	☹	↵	ф		Λ	ф	☝	Λ	̄	۵	L	L	L	L	L
M	💣	↵	Х		M	Х	☟	M	̆	۶	M	M	M	M	M
N	☠	↵	х		N	х	✋	N	̇	۲	N	N	N	N	N
O	⚐	↵	Ц		O	Ц	✗	O	̈	۷	O	O	O	O	O
P	⚑	↵	ц		Π	ц	✓	Π	̋	۸	P	P	P	P	P
Q	✈	↵	Ч		Θ	Ч	☒	Θ	̌	ۭ	Q	Q	Q	Q	Q
R	☼	↵	ч		P	ч	☑	P	̊	ۮ	R	R	R	R	R
S	✒	↵	Ш		Σ	Ш	☒	Σ	̏	ۯ	S	S	S	S	S
T	❄	↵	ш		T	ш	☒	T	̑	۰	T	T	T	T	T
U	✠	↵	Щ		Y	Щ	⊗	Y	̒	۱	U	U	U	U	U
V	✠	↵	щ		¯	щ	⊗	ς	h	۲	V	V	V	V	V
W	✙	↵	Ъ		Ω	Ъ	⃠	Ω	ʁ	۳	W	W	W	W	W
X	✠	⇪	ъ		X	ъ	⃠	Ξ	ɐ	۴	X	X	X	X	X
Y	✡	⇪	Ы		Ψ	Ы	☙	Ψ	ɑ	۵	Y	Y	Y	Y	Y
Z	☪	⇦	ы		Z	ы	&	Z	ɒ	۶	Z	Z	Z	Z	Z

Arial	Wingdings	Wingdings 3	WP Cyrillic A	Webdings	Sgreeck Fixed	WP Cyrillic A	Wingdings 2	Symbol	WP Phonetic	WP Arabic Script Sihafa					
0	🗀	⇞	ж	–	0	ж	□	0	ˈ	۰	0	0	0	0	0
1	🗁	↔	З	□	1	З	🗐	1	ˌ	۱	1	1	1	1	1
2	🗎	↕	з	🗗	2	з	🗑	2	ˉ	۲	2	2	2	2	2
3	🗏	⇠	И	◂	3	И	🗔	3	ˊ	۳	3	3	3	3	3
4	🗐	⇢	и	▸	4	и	🖵	4	ˋ	۴	4	4	4	4	4
5	🗄	⇡	Й	▴	5	Й	🖳	5	ˍ	۵	5	5	5	5	5
6	🖴	⇣	й	▾	6	й	🖷	6	ˎ	۶	6	6	6	6	6
7	🖮	↯	К	◂◂	7	К	🖷	7	ˏ	۷	7	7	7	7	7
8	🖰	↵	к	▸▸	8	к	⊙	8	ˆ	۸	8	8	8	8	8
9	🖲	↳	Л	⏮	9	Л	🖭	9	ˇ	۹	9	9	9	9	9
a	♋	⇨	Я	✔	α	Я	♋	α	ɗ	ڀ	a	a	a	a	a
b	♌	⇦	я	🚲	β	я	♌	β	ɘ	ٻ	b	b	b	b	b
c	♍	⇨	Ə	□	ξ	Ə	♍	χ	ə	پ	c	c	c	c	c
d	♎	⇦	ə	●	δ	ə	♎	δ	ɚ	ڀ	d	d	d	d	d
e	♏	⇨	Ґ́	🖿	ε	Ґ́	♏	ε	ɛ	ٹ	e	e	e	e	e
f	♐	←	ґ́	🚌	ϕ	ґ́	♐	ϕ	ɜ	ٺ	f	f	f	f	f
g	♑	→	Ґ	■	γ	Ґ	♑	γ	ɝ	ٿ	g	g	g	g	g
h	♒	↑	ґ	🚜	η	ґ	♒	η	ɞ	ٹ	h	h	h	h	h
i	♓	↓	Ғ	ⓘ	ι	Ғ	⓪	ι	ɵ	ٹ	i	i	i	i	i
j	🙰	↖	ғ	🛩	ς	ғ	①	φ	ɟ	ڈ	j	j	j	j	j
k	🙵	↗	Ҕ	🚀	κ	Ҕ	②	κ	ɸ	ډ	k	k	k	k	k
l	●	↙	ҕ	✦	λ	ҕ	③	λ	g	ڊ	l	l	l	l	l
m	❍	↘	Є	❗	μ	Є	④	μ	ɠ	ڋ	m	m	m	m	m
n	■	⇔	є	●	ν	є	⑤	ν	ɢ	ڌ	n	n	n	n	n
o	□	↕	Ҽ	🛩	ο	Ҽ	⑥	ο	ɢ	ڍ	o	o	o	o	o
p	❒	▲	ҽ	🚍	π	ҽ	⑦	π	ɣ	ڎ	p	p	p	p	p
q	❑	▼	Ӂ	🕩	θ	Ӂ	⑧	θ	ɤ	ڏ	q	q	q	q	q
r	❐	△	ӂ	✕	ρ	ӂ	⑨	ρ	ɥ	ڐ	r	r	r	r	r
s	⬧	▽	S	?	σ	S	⑩	σ	ɦ	ڑ	s	s	s	s	s
t	⧫	◀	s	🛩	τ	s	❶	τ	ɧ	ڒ	t	t	t	t	t
u	◆	▶	Ӡ	🚔	υ	Ӡ	❶	υ	н	ڔ	u	u	u	u	u
v	❖	◁	ӡ	🚌	η	ӡ	❷	ϖ	ɨ	ڕ	v	v	v	v	v
w	⬩	▷	Ӣ	🚶	ω	Ӣ	❸	ω	ɪ	ږ	w	w	w	w	w
x	⊠	◣	ӣ	⊘	χ	ӣ	❹	ξ	ɩ	ڗ	x	x	x	x	x
y	🖻	◢	I	⊖	ψ	I	❺	ψ	ɻ	ڙ	y	y	y	y	y
z	⌘	◤	i	⊗	ζ	i	❻	ζ	ʞ	ڛ	z	z	z	z	z

BRAILLE-Schrift / Tastschrift für sehschwache und blinde Personen

Von Louis BRAILLE (1825) entwickelte Kontur-Schrift. Die Schrift-Zeichen bestehen aus einem Durchdrück-Raster auf Papier. Die Zeichen haben ein 3 x 2 Punkte-Raster. Zwischen den Buchstaben wir immer eine deutliche Lücke gelassen.
Für Computer gibt ein mechanische BRAILLE-Leiste, die als einzeilige Ausgabe dient.
Über Magnet und elektrische Spulen werden kleine Metallstifte aus der Oberfläche herausgeschoben. Diese Erhebungen können dann gut ertastet werden.

Q: de.wikipedia.org ()

Als Eingabe-Gerät wird eine spezielle Schreibmaschine / Tastatur benutzt, die mit 7 Tasten auskommt.
Bekannt sind diese Schreibmaschinen vielleicht aus älteren amerikanischen Gerichts-Filmen. In amerikanischen Gerichten sind vielfach sehschwache oder blinde Personen als Protokollanten beschäftigt. Sie können besser hören und lassen sich nicht von optischen Effekten usw. ablenken.

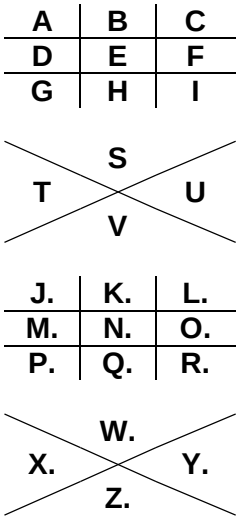
Q: de.wikipedia.org (Simon A. Eugster)

A A	B B	C C	D D	E E	F F	G G	H H	I I	J J	K K	L L
M M	N N	O O	P P	Q Q	R R	S S	T T	U U	V V	W W	X X
Y Y	Z Z	Ä	Ö	Ü	ß	ST	AU	ÄU	EU	EI	IE
CH	SCH										
,	;	:	.	?	!	'	"…	…"	-	()	§
Ziffer	0 0	1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 8	9 9	

der Freimaurer-Code

Mit einfachen – und auch recht systematischen – Symbolen arbeitet der legendäre **Freimaurer-Code**. Besonders im 18. Jahrhundert war dieser Code weit verbreitet. Zum Kodieren der Buchstaben werden die Rahmen vom Tabellen bzw. Kreuzen genutzt.
Die Freimaurer nutzten zwei 3x3-Tabellen und zwei Kreuze um die lateinischen Buchstaben zu kodieren. In jeder zweiten Struktur benutzten sie neben den Buchstaben noch Punkte.
Für die Symbol-Bildung werden nun einfach immer die angrenzenden Linien an den Buchstaben verwendet. Steht ein Punkt beim Buchstaben, dann wird dieser ebenfalls in das Freimaurer-Symbol eingezeichnet.
Der Buchstabe M ist z.B. von drei Rahmen-Linien umgeben: oben, rechts und unten. Dazu kommt der Punkt. Zusammen ergibt sich das Symbol:

Entsprechend geht man bei allen anderen Zeichen vor.



Aufgaben:

1. *Übernehme die nachfolgende Tabelle lateinischer Buchstaben (B) in Deine Mitschriften und ergänze die Freimaurer-Symbole (FS)!*

B	FS
A	
F	
K	
P	
U	
Z	

B	FS
B	
G	
L	
Q	
V	

B	FS
C	
H	
M	
R	
W	

B	FS
D	
I	
N	
S	
X	

B	FS
E	
J	
O	
T	
Y	

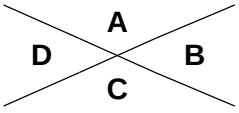
2. *Dekodieren den folgenden Satz mit dem Freimaurer-Code! Verwende dazu Deine Code-Tabelle!*

3. *Kodiere einen eigenen Satz im Freimaurer-Code und tausche diesem mit dem Banknachbarn! Dekodiere dann dessen Satz!*

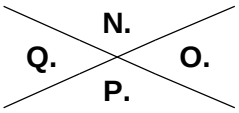
für die gehobene Anspruchsebene:

4. *Überlege Dir Varianten, wie man den Freimaurer-Code noch etwas "geheimer" machen kann!*
5. *Entwickle eine Idee, wie man z.B. auch Ziffern kodieren könnte!*

Durch ein leichtes Wechseln der Codier-Schemata kann man die "Entschlüsselung" schon deutlich erschweren. So könnte man immer an den geraden Tagen / Monaten / … die nebenstehenden Tabellen nutzen.
Hiermit codierte Texte unterscheiden sich symbolischer sehr stark.
Die "Geheim"-Texte, die mit dem originalen Freimaurer-Code erzeugt wurden. lassen sich mit dem nebenstehenden Alternativ-Code nicht dekodieren.
Mittels Häufigkeits- und Paar-Analysen lassen sich aber insgesamt schnell die gültigen Code-Tabellen rekonstruieren.
Schwieriger sind selbst erstellte und völlig frei erfundene / zusammengestellte Symbol-Tabellen. So etwas hat wohl jeder schon mal benutzt, um einer Freundin / einem Freund ein "geheimes" Briefchen zu schreiben.



E	**F**	**G**
H	**I**	**J**
K	**L**	**M**



R.	**S.**	**T.**
U.	**V.**	**W.**
X.	**Y.**	**Z.**

Aufgaben:

1.
2. ***Im Allgemeinen glaubt man, dass Codierungen immer leicht zu knacken sind. Gilt das auch für den folgenden Code? Decodiere den Text!***
   (Hilfe, wenn's gar nicht klappt: Es handelt sich um den A-tom-tom-Code. Dahinter steckt eine andere Zwei-Symbol-Codierung.)
3.

8.2. kleine Geschichte der Verschlüsselung (Chiffrierung) mit praktischen Beispielen

8.2.0. Grundlagen, Begriffe, …

Definition(en): Chiffrierung
Eine Chiffrierung / Verschlüsselung ist eine Umsetzung eines "Klartextes" in einen "Geheimtext" unter Verwendung eines "Schlüssels" (einer geheimen Vorschrift / eines geheimen Prinzips).
Chiffrierung ist die Kodierung mittels einer – nur einem begrenzten Nutzerkreis zugänglichen – (quasi geheimen) Umsetzungsverfahrens / Übertragungs-Prinzips.

Der Begriff Dechiffrierung wird sowohl für die ordentliche Rück-Übersetzung (Entschlüsseln) an der Empfänger-Stelle verstanden, als auch das geheimdienstliche Knacken ("Codeknacken" / Entziffern).

Definition(en): Dechiffrierung
Die Dechiffrierung / Entschlüsselung ist die Rück-Übertragung des "Geheimtextes" in den "Klartext".

Möglichkeiten für Chiffrierungen

- **Permutation / Transposition** Vertauschen der Symbol-Positionen
- **Substitution** Ersetzen von Symbolen durch andere
- **Expansion** Erweitern der Symbole um zusätzliche Zeichen / Textstellen / …

Zeichen ersetzen → Substitution

einfachste Variante für "verschlüsselte" Briefchen zwischen Schülern ist eine "ungewöhnliche" Chiffrier"-Tabelle. Jeder hat die wahrscheinlich schon mal aufgestellt, um seinem Freund oder Freundin einen geheimen Brief zu schreiben.
Ein solche Substitutions-Tabelle für Klar- und Code-Zeichen könnte so aussehen:

Klarz.	Codez.
A	✞
B	⌘
C	✉
D	⊙
E	♦
F	✓
G	♠
H	>
I	♓
J	☒

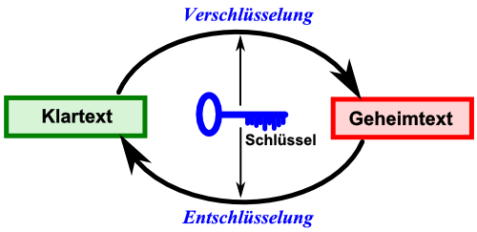
Klarz.	Codez.
K	ϟ
L	◡
M	♣
N	<
O	✶
P	⌤
Q	9
R	Γ
S	∑
T	◓

Klarz.	Codez.
U	ö
V	∏
W	α
X	▽
Y	≈
Z	⁂
0	Ψ
1	∞
2	?
3	♥

Klarz.	Codez.
4	m
5	β
6	§
7	Ω
8	%
9	▲
.	⊗
	#
?	=
!	Λ

Es handelt sich eigentlich nur solange um eine Chiffrierung, wie die Umwandlungs-Tabelle geheim bleibt. Ist diese bekannt, dann ist es eigentlich nur eine Codierung. Schließlich hätte man auch MORSE-Zeichen nutzen können.
Die Dechiffrierung kann mit der gleichen Tabelle – also dem gleichen Schlüssel erfolgen, weshalb man von einen symmetrischen Verschlüsselung spricht.

Benutzt man für Ver- und Entschlüsselung den gleichen Schlüssel bzw. das gleiche (ev. auch das umgedrehte) Verfahren, dann spricht man von symmetrischer Verschlüsselung.
Alle Geheimspachen bis in die Mitte des 21. Jahrhunderts funktionieren nach symmetrischen Verfahren.



Aufgaben:

1. *Entschlüssele den folgenden Geheim-Text! Wo steckt ein Fehler?*

♣	♦	♓	<	♦	#	<	♦	ö	♦	#	♠	♦	>	♦	♓	<	∑	✉	>	Γ	♦	✓	◓

2. *Überlege Dir einen Satz und verschlüssele ihn mit der obigen Code-Tabelle!*
3. *Tausch Deinen verschlüsselten Text mit dem vom Nachbarn und entschlüssle diesen!*

Wie knackt man so eine Chiffre?

Grundsätzlich haben wir mindestens zwei Möglichkeiten eine Chiffre zu knacken. Da ist zum einen das sture Ausprobieren. Man setzt nach und nach für jedes Zeichen (des Geheim-Alphabet) einen Klartext-Buchstaben ein. Der jeweils dechiffrierte Text wird auf Lesbarkeit geprüft. Was früher viele Tage dauerte, erledigen heute Computer in wenigen (Milli-)Sekunden. Es werden einfach alle Möglichkeiten erzeugt und mit Wörterbüchern (Wort-sammlungen) abgeglichen. Der Text, in dem die meisten Wörter gefunden werden, ist wahr-scheinlich der Klartext. Und hat man dann erst einmal ein gültiges Paar Klartext - Geheim-text, dann ist das Aufstellen der Chiffren-Tabelle kein Problem mehr.
Diese großen Durchprobier-Angriff wird auch **Brute Force** (für brutale Gewalt) genannt. Das Abgleichen mit Wort-Sammlungen bezeichne wir als **Wörterbuch-Analyse**. Manchmal bleibt einem Kryptoanalytiker keine andere Wahl.
Nachteil der "Brute Force"-Angriffe ist der Zeitaufwand. Im Allgemeinen muss man sehr viel probieren. Sachlich müsste man z.B. für einen deutschen Text mit 26 Buchstaben ungefähr

n = 26 * 25 * 24 * 23 * … * 3 * 2 * 1 = 26! =

Versuche machen. (Die 26 Buchstaben werden vor allem für Berechnungen und Vergleiche benutzt. In der Welt der Kryptologen spielen natürlich viel größere Alphabete eine Rolle.)
Allerdings könnte man auch schon beim 1. Versuch Glück haben. Der ungünstigste Fall ist es, wenn wir erst mit dem letzten Versuch die richtige Chiffre finden. Statistisch kommen wir auf durchschnittlich:

Versuche.

Einen Angriff durch Brute Force kann man nur durch große Alphabete und lange Schlüssel-wörter (wenn sie dann benutzt werden) behindert werden. Verhindern kann man sie nicht. Für den Chiffre-Knacker ist es natürlich imme wichtig, eine Nachricht rechtzeitig zu ent-schlüsseln. Tage, Wochen, Monate oder Jahre später nützt das nichts.

Aufgaben:

1. ***Wie lange bräuchte ein Computer durchschnittlich für einen Brute Force-Angriff auf einen Substitutions-Chiffre mit 26 Zeichen, wenn er 1'000 Varianten in einer sekunde ausprobieren kann? Schätze zuerst die Zeit! Berechne dann den exakten Wert! Gebe die Zeiten jeweils in sinnvollen Einheiten an!***
2. ***Wie lange würde der gleiche Angriff mit einem Super-Computer dauern, der 1'000'000'000 Varianten pro Sekunde berechnet?***

Eine zweite Möglichkeit zum Angriff auf eine einfache Substitutions-Chiffre ist die **Häufigkeits-Analyse**. Hierbei nutzt man die Eigenschaft von Buchstaben in Texten aus, dass diese nicht gleichmäßig häufig vorkommen. In der deutschen Sprache kommen zum Beispiel die Buchstaben ERNSTL viel häufiger vor, als PVJYXQ. Bei der Häufigkeits-Analyse stellt man die Häufigkeit der Geheimtext-Zeichen mit der der Klartext-Zeichen gegenüber. Meist reichen dann schon ein paar Proben und man kann die ersten Buchstaben zuordnen. Dabei macht man sich auch zunutze, das wiederum bestimmte Päarchen von Buchstaben häufiger sind als andere.

	deutsch					englisch	
Rang	**Bst.**	**Hfgkt. [%]**					
1	**E**	17,40				12,70	1
2	**N**	9,78				6,75	6
3	**I**	7,55				6,97	5
4	**S**	7,27				6,33	7
5	**R**	7,00				5,99	9
6	**A**	6,51				8.17	3
7	**T**	6,15				9,06	2
8	**D**	5,08				4,25	10
9	**H**	4,76				6,09	8
10	**U**	4,35				2,76	13
11	**L**	3,44				4,03	11
12	**C**	3,06				2,78	12
13	**G**	3,01				2,02	17
14	**M**	2,53				2,41	14
15	**O**	2,51				7,51	4
16	**B**	1,89				1,49	20
17	**W**	1,89				2,36	15
18	**F**	1,66				2,23	16
19	**K**	1,21				0,77	22
20	**Z**	1,13				0,07	26
21	**P**	0,79				1,93	19
22	**V**	0,67				0,98	21
23	**ß**	0,31					
24	**J**	0,27				0,15	23
25	**Y**	0,04				1,97	18
26	**X**	0,03				0,15	24
27	**Q**	0,02				0,10	25

Q: de.wikipedia.org;

Selbst die Häufigkeit der Anfangs- und End-Buchstaben ist im Deutschen sehr speziell. Für Text-Analyse ist es sehr hilfreich, auch diese Häufigkeiten zu beachten.

Anfangs-Buchstabe		
Rang	**Bst.**	**Hfgkt. [%]**
1	D	14,2
2	S	10,8
3	E	7,8
4	I	7,1
5	W	6,8

Q: de.wikipedia.org

Benutzt man z.B. die Häufigkeit der 5 wahrscheinlichsten End-Buchstaben, dann kann man mit ihnen fast 70 % der Wörter dahingehend "erraten".

End-Buchstabe		
Rang	**Bst.**	**Hfgkt. [%]**
1	N	21,0
2	E	15,1
3	R	13,0
4	T	10,3
5	S	9,6

Q: de.wikipedia.org

Wie verhindert man nun wieder eine Häufigkeits-Analyse?

Die Antwort heißt **homophone Chiffrierung** oder **Verschlüsselung**. Hierbei werden die Klartext-Zeichen, die besonders häufig sind und damit für eine Häuffigkeits-Analyse in Fragen kommen, durch weitere Symbole ergänzt. Ganz optimierte homophone Chiffren verwenden genausoviele Symbole, dass alle Häufigkeits-Unterschiede ausgeglichen sind. Kommt ein Klartext-Buchstabe also z.B. dreimal häufiger vor, als der unwahrscheinlichste Buchstabe, dann benötigt ein super homophones Alphabet für diesen Buchstaben drei Geheimtext-Symbole. Man braucht dann zwar sehr viele Symbole, aber der Aufwand heiligt die Mittel. Als Nebeneffekt wird nun auch der Brute Force-Angriff deutlich schwieriger – um nicht zu sagen, praktisch unmöglich.

Um die deutschen Buchstaben (einschließlich des ß) optimal homophon zu verschlüsseln bräuchte man also 172'173 Symbole. Echte Zeichen werden da vielleicht schwierig, aber in Zeiten der Computer können wir ja z.B. Bit-Folgen nutzen.

	deutsch				
Rang	**Bst.**	**Hfgkt. [%]**	**+ Ziffern (z.B.)**	**eff. Symbol-Anzahl**	**opt. Symbol-Anzahl**
1	**E**	17,40	1, 2, 3	10	580
2	**N**	9,78	4	5	326
3	**I**	7,55	5	4	252
4	**S**	7,27	6	4	242
5	**R**	7,00	7	4	233
6	**A**	6,51	8	4	217
7	**T**	6,15	9	4	205
8	**D**	5,08		4	169
9	**H**	4,76		4	159
10	**U**	4,35		4	145
11	**L**	3,44		3	115
12	**C**	3,06		3	102
13	**G**	3,01		3	100
14	**M**	2,53		3	84
15	**O**	2,51		3	84
16	**B**	1,89		2	63
17	**W**	1,89		2	63
18	**F**	1,66		2	55
19	**K**	1,21		2	40
20	**Z**	1,13		2	38
21	**P**	0,79		1	26
22	**V**	0,67		1	22
23	**ß**	0,31		1	10
24	**J**	0,27		1	9
25	**Y**	0,04			1
26	**X**	0,03			1
27	**Q**	0,02			1

Q: de.wikipedia.org;

Aufgaben:

1.

2. Wie könnte man eine Chiffre mit zusätzlichen Ziffern-Symbolen trotzdem angreifen? Überlege Dir mindestens eine Möglichkeit und erkläre sie den Kursteilnehmern!

3.

für die gehobene Anspruchsebene:

x. Wie lang müsste eine Binär-Zahl (Stellen-Anzahl) sein, damit die 27 oben genutzten deutschen Buchstaben unknackbar verschlüsselt werden können? Überlege Dir, ob so eine Verschlüsselung für heutige Computer und Internet-Leitungen machbar wäre!

Links:

gc.de/gc/buchstabenhaeufigkeit/ (diverse Analysen, Codierungen und Decodierungen)

Zeichen(-Reihen)folge verändern → Transposition

Hierbei handelt es sich schon um eine echte Chiffrierung, weil jetzt keine 1 : 1 Übersetzung von Zeichen oder Zeichen-Gruppen erfolgt.

Eine sehr einfache – aber schon sehr wirkungsvolle – Möglichkeit ist es, einfach zwei benachbarte Zeichen zu tauschen.

V	E	R	S	C	H	L	U	E	S	S	E	L	N		I	S	T		C	O	O	L		

→ Paartausch →

e	v	s	r	h	c	u	l	s	e	e	s	n	l	i		t	s	c		o	o		l	

Dieser einfache Transpositions-Code ist ebenfalls symmetrisch. Wir verwenden das gleiche verfahren des Paar-Tausches, um den Code-Text zu dechiffrieren.

e	v	s	r	h	c	u	l	s	e	e	s	n	l	i		t	s	c		o	o		l	

→ Paartausch →

V	E	R	S	C	H	L	U	E	S	S	E	L	N		I	S	T		C	O	O	L		

Aufgaben:

1. *Entschlüssele den folgenden Geheim-Text! Wo steckt / stecken der / die Fehler?*

e	j	e	d		r	m	a	h	c		t	m	a		l	e	f	l	h	r	e		

2. *Überlege Dir einen Satz und verschlüssele ihn mit der Paartausch-Methode!*
3. *Tausch Deinen verschlüsselten Text mit dem vom Nachbarn und entschlüssle diesen!*

für Experten und zum Knobeln:

4. *Überlege Die einen eigenen Code! Verschlüssele eine Nachricht damit und übergib die Nachricht Deinem Nachbarn!*
5. *Versuche den Code des Nachbarn zu knacken!*
6. *Wenn das Knacken nicht geklappt hat, dann lass Dir den Code vom Nachbarn geben und dechiffriere die Nachricht!*

Durch die Verlängerung der Austausch-Gruppe auf z.B 3, 4 oder 5 Zeichen lassen sich die Klartexte noch mehr verschleiern und es ergibt sich auch die Möglichkeit unterschiedliche Schlüssel festzulegen.
Betrachten wir mal als Beispiel Vierer-Gruppen. Wenn die Klartext-Zeichen dort in der Reihenfolge:

1	**2**	**3**	**4**

dann ergeben sich die folgenden veränderten Reihenfolgen – die wir hier Schlüssel nennen:

1	2	4	3
1	4	3	2
2	3	4	1
3	1	4	2
3	4	2	1
4	2	3	1

1	3	2	4
2	1	3	4
2	4	1	3
3	2	1	4
4	1	2	3
4	3	1	2

1	3	4	2
2	1	4	3
2	4	3	1
3	2	4	1
4	1	3	2
4	3	2	1

1	4	2	3
2	3	1	4
3	1	2	4
3	4	1	2
4	2	1	3

Nehmen wir als Beispiel den Schlüssel 2431. Wenn man die ersten vier Buchstaben des Alphabetes (also: ABCD) verschlüsseln würde, ergäbe sich: bdca. Das klingt nicht wirklich mehr nach Alphabet-Anfang.
Bei längeren Texten, werden die Zeichen in Spalten geschrieben. Aus dem Text:

MIT TRANSPOSITION WIRD ES SCHON SCHWERER ZU ENTZIFFERN

1	2	3	4
M	I	T	
T	R	A	N
S	P	O	S
I	T	I	O
N		W	I
R	D		E
S		S	C
H	O	N	
S	C	H	W
E	R	E	R
	Z	U	
E	N	T	Z
I	F	F	E
R	N		Y

Transposition →

2	4	3	1
I		T	M
R	N	A	T
P	S	O	S
T	O	I	I
	I	W	N
D	E		R
	C	S	S
O		N	H
C	W	H	S
R	R	E	E
Z		U	
N	Z	T	E
F	E	F	I
N	Y		R

Nachdem die Spalten "transponiert" wurden, schreiben wir den so verschlüsselten Text nun wieder in die Zeilen-Form zurück:

i tmrnatpsostoii iwnde r csso nhcwhsrreez u nztefefiny r

Angenommen wir würden wissen, das es sich um einen Transpositions-Algorithmus gehandelt hat, dann bleiben noch Fragen offen, auf die wir nur durch Probieren eine Lösung finden können:

Wieviele Spalten wurden verwendet? (Wieviel stellig ist der Schlüssel?)
Welches Austausch-Muster (welcher Schlüssel) wurde verwendet?

Rein theoretisch geht dabei alles, von 2 bis 26. Interessant ist nun die Frage, wieviele Schlüssel gibt es denn dann? Bei der Schlüssel-Länge 2 gab es zwei Möglichkeiten (12 und 21). Die originale Reihenfolge werden wir nicht nehmen, also bleit nur 1 Schlüssel.
Bei einer Schlüssel-Länge von 3 gibt es die Schlüssel:

1 2 3 2 3 1 3 2 1 1 3 2 2 1 3 3 1 2

(c,p) 2015 – 2020 lsp: dre

also schon 6 Schlüssel. Für die Schlüssel-Länge 4 haben wir die Schlüssel oben schon zusammengestellt. Das waren 24.
Die genaue Berechnung der möglichen Schlüssel ist etwas aufwändiger:

$N_S = n * (n-1) * (n-2) * \ldots * 1$

Die Mathematiker schreiben das so:

$N_S = n!$

und nennen es Fakultät. Bei einer Schlüssel-Länge von 5 sollte es also 5! (sprich: "**Fünf Fakultät"**) Schlüssel geben.
Aber wie kommt diese Rechnung zustande?
Betrachten wir wieder unseren 4er Code. Für das erste Zeichen können wir alle 4 Spalten nutzen. Bei der zweiten Spalte haben wir nur noch drei andere übrig. Also würden wir hier schon 4 * 3 = 12 rechnen.
In die dritte Spalte können jetzt noch 2 reinrutschen, also multiplizieren wir mit 2. Das ergibt schon 24 Möglichkeiten. Da die letzte Spalte nur noch eine übrig bleibt, haben wir hier keine Freiheitsgrade. Die Multiplikation mit 1 verändert das Ergebnis nicht.
Insgesamt kommen wir auf 24 Schlüssel mit effektiv 23 nutzbaren.

Schlüssel-Länge n	**Anzahl Schlüssel N_S**	**nutzbare Schlüsselzahl N_S -1**
2	2	1
3	6	5
4	24	23
5		
6		
7		
8		
9		
10		
11		
12		
13		
14	87'178'291'200	87'178'291'199
15	1'307'674'368'000	1'307'674'367'999
16	20'922'789'888'000	20'922'789'887'999
17	35'568'742'8096'000	35'568'742'8095'999
18	6'402'373'705'728'000	6'402'373'705'727'999
19	121'645'100'408'832'000	121'645'100'408'831'999
20	2'432'902'008'176'640'000	2'432'902'008'176'639'999
21	51'090'942'171'709'440'000	51'090'942'171'709'439'999
22	1'124'000'727'777'607'680'000	1'124'000'727'777'607'679'999
23	25'852'016'738'884'976'640'000	25'852'016'738'884'976'639'999
24	620'448'401'733'239'439'360'000	620'448'401'733'239'439'359'999
25	15'511'210'043'330'985'984'000'000	15'511'210'043'330'985'983'999'999
26	403'291'461'126'605'635'584'000'000	403'291'461'126'605'635'583'999'999
Summe gesamt	**419'450'149'241'406'189'412'940'313**	**419'450'149'241'406'189'412'940'287**

Aufgaben:

1. *Berechne die fehlenden Angaben!*
2. *Wie wird die ganz untere Zahl ausgesprochen?*
   (Wer es nicht alleine hinbekommt, schaut als Hilfe auf der Webseite https://de.wikipedia.org/wiki/Zahlennamen *nach, wie große Zahlen heißen!)*
3.

Was besagt nun die riesengroße Zahl in der letzten Zeile der Tabelle ("Summe gesamt")? Sie stellt die möglichen Schlüssel für eine Chiffrierung mittels einfacher Transposition dar. Mit anderen Worten soviele Versuch mus man machen bis man bei Code-Knacken alle Möglichkeiten ausprobiert hat. Die Zahl sagt uns wohl wenig.Deshalb wählen wir als Vergleich die Arbeitszeit, die jemand für das Code-Knacken maximal braucht. Nehmen wir an, für jede Möglichkeit bräuchten wir nur eine Sekunde, dann würde eine einzelne Person $1{,}33 * 10^{19}$ Jahre dafür brauchen. Unser Universum ist aber erst geschätzte $15 * 10^{9}$ Jahre alt. Also kein realistisches Szenario.
Die nächste Option ist die Nutzung eines Computers. Der arbeitet deutlich schneller und zu jeder Tageszeit. Nehmen wir für einen superschnellen Rechner an, dass er 1 Milliarde (109) Schlüssel ausprobieren kann. Selbst jetzt würde der Rechner noch $1{,}33 * 10^{10}$ Jahre brauchen.
Bei den Berechnungen muss noch im Hinterkopf bleiben, das wi hier mit einem einfachen deutschen Alphabet mit den üblichen 26 Buchstaben arbeiten. Wenn man noch die Ziffen und notwendige Satz-Zeichen mit einbezieht, dann verschlimmert sich der Aufwand nochmal.
Aber in unserem Fall kennen wir den Schlüssel. Also dechiffrieren wir die Nachricht nach dem gleichen Verfahren:

i tmrnatpsostoii iwnde r csso nhcwhsrreez u nztefefiny r

2	4	3	1
i		t	m
r	n	a	t
p	s	o	s
t	o	i	i
	i	w	n
d	e		r
	c	s	s
o		n	h
c	w	h	s
r	r	e	e
z		u	
n	z	t	e
f	e	f	i
n	y		r

Transposition →

1	2	3	4
M	I	T	
T	R	A	N
S	P	O	S
I	T	I	O
N		W	I
R	D		E
S		S	C
H	O	N	
S	C	H	W
E	R	E	R
	Z	U	
E	N	T	Z
I	F	F	E
R	N		Y

Letztendlich machen wir daraus wieder eine Zeile und erhalten als Klartext:

MIT TRANSPOSITION WIRD ES SCHON SCHWERER ZU ENTZIFFERN

(c,p) 2015 – 2020 lsp: dre

Aufgaben:

1. *Legt in einer kleinen Kommunikations-Gruppe (aus mindestens 2 Partnern oder kleinen Arbeitsgrüppchen) einen Transpositions-Schlüssel der Länge 5 fest! Chiffriert dann jeweils eine Nachricht und tauscht diese aus!*
2. *Dechiffriert die erhaltene Nachricht und antwortet darauf!*
3. *Wieviele Transpositions-Schlüssel gibt es für eine Schlüssel-Länge von 5, wenn das Alphabet um die Klein-Buchstaben und Ziffern erweitert wird? Begründe Deine Meinung / Berechnung / … ausführlich!*

für die gehobene Anspruchsebene:

4. *Welche Fakultät müsste man berechnen, wenn man für das Alphabet aus Aufgabe 3 die gesamte Alphabet-Länge als Transpositions-Schlüssel-Länge benutzen wollte? (Ermittle die Fakultät mit der Webseite fakultaet.net! Schreibe die Zahl als Zehner-Potenz auf!*

Tabellen-Verschlüsselung

Permutation mit einem Schlüsselwort

Benutzt wird dieses Mal eine Tabelle mit so vielen Spalten, wie das Schlüsselwort Buchstaben hat. Wie wir ja schon erfahren haben, ist ein reichlich langes Schlüsselwort die beste Wahl. Wir wählen hier als Beispiel KRYPTOGRAPHIE. Da im Verfahren die Buchstaben des Schlüsselwortes die Spalten beschriften, darf jeder Buchstabe nur einmal vorkommen. Somit bleibt KRYPTOGAHIE
Im gleich aufgezeigten Verfahren wird zudem noch ein Leerzeichen als Trenner zwischen Schlüsselwort und Text verwendet. Damit ergibt sich eine Schlüssel-Länge von 13.
Die Schlüssel-Länge wird mit in den Schlüssel aufgenommen. Damit das nicht ganz so offensichtlich wird, verwenden wir einfach eine Hilfs-Formel zum Berechnen:

SchlüsselCode = SchlüsselLänge - 5

also:

8 = **13** - 5

K	**R**	**Y**	**P**	**T**	**O**	**G**	**A**	**H**	**I**	**E**	**8**	

Nun tragen wir den Klartext zeilenweise in die Tabelle ein. Sollten noch Zeichen fehlen, dann können diese z.B. durch zufällige Zeichen erweitert werden.

K	**R**	**Y**	**P**	**T**	**O**	**G**	**A**	**H**	**I**	**E**	**8**	
H	E	U	T	E		G	E	H	E	I	M	E
R		T	R	E	F	F		1	5	0	0	
A	N		D	E	R		E	I	C	H	E	

Durch Vertauschen / Neuanordnen der Spalten kommt es zur Permutation. Die neuen Spalten-Positionen sollten ausgewürfelt werden. Nur so steckt genug Zufall in der Verschlüs-

g	**o**	**p**	**k**	**y**	**a**	**e**	**h**	**r**		**8**	**o**	**i**
g		t	h	u	e	i	h	e	e	m		e
f	f	r	r	t		0	1			0	f	5
	r	d	a		e	h	i	n		e	r	c

selung.
Die Tabelle wird dann sequentiell an den Empfänger übertragen.

gopkyaehr 8oig thueiheem effrrt 01 0f5 rda ehin erc

Da Entschlüsseln nutzt das umgedrehte Verfahren. Wir extrahieren zuerst die Spalten-Anzahl. Das muss die erste Zahl sein. In unserer Geheim-Nachricht ist das die 8. Bei einem sehr langen Text könnten es auch mehrere Ziffern sein. Mittels der Spalten-Formel berechnen wir die Spalten-Anzahl:

SchlüsselLänge = SchlüsselCode + 5

also:

13 = **8** + 5

Zur Kontrolle können wir die Länge der Geheim-Nachricht durch die Schlüssel-Länge teilen. Wenn das klappt, haben wir eine passende Spalten-Anzahl. In diese Tabelle verteilen wir wieder zeilenweise den Geheimtext.
Die oberste Zeile soll dann wieder unser Schlüsselwort werden. Dieses herauszubekommen kann sehr schwer werden.
Am sichersten ist es also bei diesem Verfahren, das Schlüssel-Wort auf einem anderen Weg zum Empfänger zu transportieren oder eine Schlüsselbuch zu verwenden.

g	**o**	**p**	**k**	**y**	**a**	**e**	**h**	**r**		**8**	**o**	**i**
g		t	h	u	e	i	h	e	e	m		e
f	f	r	r	t		0	1			0	f	5
	r	d	a		e	h	i	n		e	r	c

Wenn man das Schlüsselwort hat, dann kann das Umsortieren wieder beginnen und durch eine erneute Permutation erhalten wir den Klar-Text zurück.

K	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**H**	**I**	**E**	
H	E	U	T	E		G	E	H	E	I	M	E
R		T	R	E	F	F		1	5	0	0	
A	N		D	E	R		E	I	C	H	E	

Zeichenfolgen erweitern → Expansion

Auch bei den Expansions-Verfahren handelt es sich um eine echte Chiffrierung. Schöne Beispiele sind die Räuber-Sprache oder die BB-Sprache.
Bei der Räubersprache werden bestimmte (häufige) Buchstaben oder Buchstaben-Kombinationen durch andere (weniger häufige) Buchstaben-Kombinationen erweitert. Dadurch ist eine Häufigkeits-Analyse deutlich erschwert. Expandierte Texte sind zudem auch schwer zu lesen, da uns die typischen Lese-Muster fehlen.
Die Räubersprache geht auf den Kinderbuch-Helden Kalle Blomquist (von Astrid LINDGREN) zurück. Als Regel ist festgelegt, dass jeder Konsonant eines Wortes verdoppelt und dazwischen noch ein **o** ergänzt wird.
Aus dem Wort:

RAEUBERSPRACHE

wird so:

r**or**aeub**ob**er**or**s**os**p**op**r**or**a**c**oc**h**oh**e

Ein Wort-Ungetüm, dass man kaum aussprechen kann. Was dabei interessant ist, ist die Tatsache, dass mit einer sehr einfachen Regel ein kaum noch verständlicher Text erzeugt

wird. Natürlich kann man den Geheimtext bei Kenntnis der Regel schnell wieder dechiffrieren.
Cleveren Computern – oder besser ihren Programmierern wird bei der Text-Analyse natürlich die größere (verdoppelte) Häufigkeit der Konsonanten auffallen. Dazu noch die unzähligen O's. Da kommt man schnell auf ein Verfahren zum Knacken des Räuber-Code's.

Aufgaben:

1.

2. Beschreibe die Dechiffrierung der Räubersprache in einem Algorithmus!

3.

Bei der Löffel-Sprache sind die Vokale die auszutauschenden Zeichen. Um den Austausch nicht zu offensichtlich zu machen, verwendet man die nebenstehende Tabelle.
Man verdoppelt hier die Vokale und fügt in die Verdopplung die Silbe **lew** ein.
Schauen wir uns den Effekt beim Verschlüsseln von:

LÖFFELSPRACHE

an. Daraus wird:

Vokal, **Umlaut, Zwielaut**	**Ersatz in Löffelsprache**	**alternative Löffelsprache 2**
a	alewa	alefa
e	elewe	elefe
i	ilewi	ilefi
o	olewo	olefo
u	ulewu	ulefu
ä	älewä	älefä
ö	ölewö	ölefö
ü	ülewü	ülefü
ei	elewei eilewei	elefei eilefei
ie	ilewie ielewie	ilefie ielefie
au	alewau aulewau	alefau aulefau
eu	eleweu euleweu	elefeu eulefeu

lölewöffelewelspralewachelewen

Auch hier ist das ursprüngliche Wörtchen nicht wiederzuerkennen. Einem Kryptoanalytiker fallen natürlich die häufigen **lew**'s auf. Das vor und hinter dem lew immer der kgleiche Vokal steht ist auch leicht zu erkennen. So kann man Schritt für Schritt hinter dieses Verschlüsselungs-Verfahren kommen.

BB-Sprache / Bebe-Sprache / Baby-Sprache
Regeln:

- jedem Vokal oder Umlaut wird mit einem vorgestellten **b** wiederholt (e → ebe)
- ***außer*** vor Zwielauten (**ei**, **au**, **eu**, **äu** und **öu**): diesen wird ein **ab** vorgestellt (au → abau)
- und ***außer*** Zwielaut **ie**: hier wird nur ein **i** vor das **b** vorgesetzt (ie → ibie)

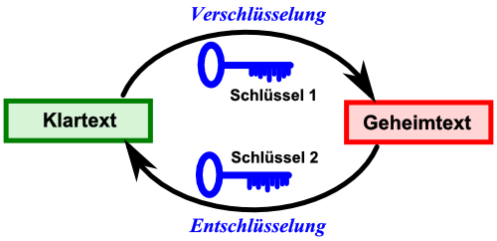
Aufgaben:

1.
2.
3. *Verschlüssele den folgenden Text mit der Hühner-Sprache! Diese funktioniert so ähnlich, wie die Löffel-Sprache.*
4. *Beschreibe das Verfahren zur Bildung der Ersatz-Zeichenketten für die Hühner-Sprache!*
5. *Entschlüssele den folgenden Text!*
6. *Erstelle einen Algorithmus für die Entschlüsselung eines Textes in der Hühner-Sprache!*

Vokal, Umlaut, Zwielaut	**Ersatz in Hühnersprache**	**alternative Hühnersprache 2**
a	ahadefa	ahadewa
e	ehedefe	ehedewe
i	ihidefi	ihidewi
o	ohodefo	ohodewo
u	uhudefu	uhudewu
ä	ähädefä	ähädewä
ö	öhödefö	öhödewö
ü	ühüdefü	ühüdewü
ei	eiheidefei	eiheidewei
ie	iehiedefie	iehiedewie
au	auhaudefau	auhaudewau
eu	euheudefeu	euheudeweu

Bei den letzten betrachteten Sprachen handelt es sich um sogenannte Spiel-Sprachen. Das sind Sprachen, die vor allem von Kindern und Jugendlichen benutzt werden. Für echte kryptographische Zwecke – vor allem im Zeitalter der Computer – eignen sie sich nicht. Sie sind durch recht einfache Analysen sehr leicht zu knacken.

Chiffrieren und Dechiffrieren mit verschiedenen Schlüsseln



Auf den ersten Blick könnte man glauben, dass das oben beschriebene Verfahren das einzige sei. Interessanterweise haben Kryptographen mehrere Verfahren gefunden, bei dem für die Ver- und Entschlüsselung verschiedene Schlüssel benutzt werden. Wie das funktioniert, schauen wir uns später an. Solche Verfahren werden asymmetrisch genannt und bestimmen heute die Kryptographie.
Sie basieren auf verschiedenen mathematischen Verfahren und lassen sich teilweise auch als sicher beweisen.

Aufgaben:

1.

2. **Von dem folgenden Satz ist nur bekannt, dass er in der Erbsen-Sprache geschrieben sein soll. Kannst Du ihn entziffern? Welche Regeln beschreiben diese Sprache?**

   erbse ierbse nerbse erbse erbse cerbse herbse terbse ierbse rerbse rerbse erbse serbse perbse rerbse aerbse cerbse herbse erbse.

3. **Informiere Dich über die Spielsprachen "Pig Latin" und "Mattenenglisch"! Erläutere die Regeln der Sprache anhand eines kurzen Satzes! Verschlüssle und entschlüssele einen Satz (ersatzweise den eines anderen Schülers)!**

4. **Verschlüssele den folgenden Text mit Hilfe der nebenstehenden Tabelle in die Sprache Grüfnisch! Diese funktioniert so ähnlich, wie die Löffel-Sprache.**

   TREFF MITTAG ALTE EICHE

5. **Verschlüssele einen kurzen Text mit dem Transpositions- und Expansions-Schlüssel: 3 5 4 3 2 1 4 !**

6.

Vokal, Umlaut, Zwielaut	**Ersatz in Grüfnisch**	**alternatives Grüfnisch 2**
a	anafa	
e	enefe	
i	inifi	
o	onofo	
u	unufu	
ä	änäfä	
ö	önöfö	
ü	ünüfü	
ei	enefei	
ie	inifie	
au	anafau	
eu	enefeu	

8.2.1. die Anfänge der geheimen Botschaften

Tätovieren der Nachricht auf die Kopfhaut
nachdem die Haare nachgewachsen waren, wurde der Kurier losgeschickt und an Ziel dann rasiert und nicht selten auch gleich noch getötet

Den Spartanern wird nachgesagt, dass sie ihre geheimen Botschaften auf Lederbändern notierten. Auffällig war vielleicht, dass die Buchstaben nicht waagerecht (- also quasi als Zeile -) geschrieben wurden, sondern untereinander (in einer Spalte). Schaut man sich das verwendete Verfahren an, dann wird auch klar, warum das so war.

Die Spartaner wickelten das Lederband um einen zylindrischen Stab. So entstand eine Spirale. Erst jetzt wurde der Text eingeprägt. Geschrieben wurde alledings waagerecht auf dem Stab. Danach wickelte man das Lederband ab und konnte ihn zum Empfänger transportieren.
Der Empfänger musste nun das Lederband um einen Stab wickeln, der exakt den gleichen Durchmesser hatte, wie der auf der Sender-Seite. Ausgehend vom ersten eingeprägten Buchstaben konnte der Empfänger den Text dann lesen.

T_HURFRDEE_LEEEHAUFRRRMSF_FE_S

Aufgaben:

1. ***Entschlüssele den unsichtbaren Teil der Nachricht!***
2. ***Schicke auf dem gleichen Stab (der gleichen Skytale) eine Bestätigung und als Zeitpunkt den Sonnenaufgang!***

notwendiges Material:
Chips-Dosen mit verschiedenen Durchmessern, Papp- oder Blech-Dosen z.B. von Whisky-Flaschen (im Folgenden auch Chips-Dosen genannt); DIN A4 Papier; Schere; Lineal; Klebstoff; Filzstift

Aufgaben:

1. ***Stelle Dir einen 1 cm breiten und ungefähr 1 m langen Papierstreifen her!***
2. ***Wickle den Papierstreifen spiralig um eine Chips-Dose! Notiere den folgenden Text in Großbuchstaben waagerecht auf den Streifen! Leere Zeilen werden mit beliebigen Buchstaben beschriftet.***
3. ***Versuche den Text mittels einer dickeren oder schmaleren Chips-Dose zu entschlüsseln!***
4. ***Verwende nun zum Entschlüsseln die Dose mit dem richtigem Durchmesser!***

Die Buchstaben des Klartextes werden mittels Skytale nicht verändert, sondern nur in der Position geändert. Ein solches Chiffrier-Verfahren nennen wir Transposition.
Da die Buchstaben erhalten bleiben, können solche Verfahren mittels Krypto-Analyse meist schenll geknackt werden. Für alle Sprachen sind bestimmte Häufigkeiten für die einzelnen Buchstaben bekannt. In der deutschen Sprache sind z.B. die Buchstaben besonders häufig. Praktisch gilt das zwar nur für beliebig lange Texte, aber auch bei kurzen Fachtexten ist die Analyse schon eine große Hilfe.
In der fortgeschrittenen Krypto-Analyse werden dann als nächstes Buchstaben-Folgen untersucht. Dabei sucht man z.B. nach typischen Zweier-Kombinationen: ck, ei, au usw. usf.
Auch fürdiese gibt es bestimmte Häufigkeiten.

Zick-Zack-Methode

Bei der Zick-Zack-Methode (ZickZack-Code) wird ein nummerischer Wert als Schlüssel festgelegt. Dieser bestimmt die Tiefe des ZickZack-Musters, in dem man den Klartext schreibt.
Der folgende zu chiffierende Klartext:

M	O	R	G	E	N		I	S	T		P	A	R	T	Y		B	E	I		L	E	N	A

soll mit ZickZack6 verarbeitet werden:

M																								
	O								T		P								I		L			
		R						S				A						E				E		
			G				I						R				B						N	
				E										T										A
					N										Y									

Heraus kommt ein unübersichtlicher Buchstaben-Salat.

m			o	t	p	i	l	r	s	a	e	e	g	i	r	b	n	e		t		a	n	y

In der Chiffrier- und Funk-Technik verwendet man gerne Fünfer-Blöcke. Das verwirrt nochmal und man kann bei Funk-Übertragungen kleinere Übertragungs-Fehler schnell bemerken und bei Bedarf einen Fünfer-Block noch einmal senden.

m			o	t
p	i	l	r	s
a	e	e	g	i
r	b	n	e	
t		a	n	y

Aufgaben:

1. *Überlege Dir eine Nachricht für Deinen Arbeitsplatz-Nachbarn! Chiffriere ihn mit einem 4er ZickZack! Notiere die Geheim-Nachricht auf einem karierten Zettel!*
2. *Tauscht die Zettel und dekodiere die empfangene Geheim-Nachricht!*

für Experten und zum Knobeln:

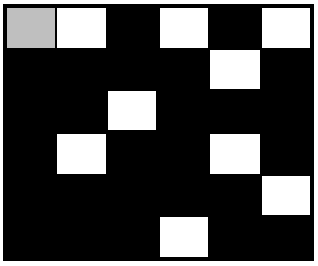
3. *Codiere eine Nachricht an Deinen Arbeitsplatz-Nachbarn mit einem beliebigen ZickZack! Tauscht die Geheim-Nachrichten und versucht sie zu dechiffrieren!*
4. *Überlege Dir Abwandlungen des ZickZack-Codes!*

Knacken des ZickZack-Codes

Ausprobieren der Varianten
Wieviele muss man ausprobieren?

Analyse der Buchstaben-Häufigkeit
das Geheimnis des ERNSTL

Schablonen nach FLEIßNER



Das jetzt beschriebene Verfahren wurde 1881 vom östereichischen Oberst Eduard FLEIßNER entwickelt. Er entwickelte eine Schablone, die aus 6 x 6 Kästchen bestand. Sie bestand dabei aus vier 3 x 3 Mustern, die so angeordnet sind, dass sie sich beim Drehen der Schablone zu einem Voll-Quadrat ergänzten.
Man legt die Schablone zuerst mit der gekennzeichneten (grauen) Ecke links oben auf ein Blatt. Nun können die ersten Buchstaben des Klartextes durch die ausgestanzten (hellen) Felder geschrieben werden.
Ist man am Ende angekommen, wird die Schablone um 90° (z.B. immer nach rechts) gedreht. Die Schablone muss aber auf dem ursprünglichen "6 x 6"-Schreibfeld positioniert werden. Nun werden die nächsten Buchstaben geschrieben. Dieses Verfahren wird dann noch zweimal wiederholt, bis die Scheibe wieder in der Ausgangs-Position ist. (Jetzt sieht man auch die zuerst geschriebenen Buchstaben wieder.)
Für die nächsten Zeichen wird die Schablone nun neben dem ersten "6 x 6"-Schreibfeld gelegt und das gesamte Verfahren solange wiederholt, bis der Klartext zuende ist. Leer-Stellen werden mit beliebigen Buchstaben ausgefüllt.
Echte Kryptologen würden die Buchstaben-Quadrate jetzt wieder in Fünfer-Blöcker zerlegen. Daruf verzichten wir hier mal.
Ein mit der obigen Schablone verschlüsselter Text könnte also z.B. so aussehen:

In Jules VERNE's utopischen Roman "Mathias Sandorf" wird die Verschlüsselung mittels einer FLEIßNERschen Schablone erwähnt.

Aufgaben:

1.
2.
3. Entwickle eine eigene "4 x 4"-Schablone nach FLEIßNER!

Neben der Positions-Veränderung von Buchstaben kann man natürlich auch ihre Symbole verändern. Solche Chiffrier-Verfahren nennen wir Substitutionen.

Aufgaben:

1.
2.
3.

Aber kommen wir zu unserer kleinen Geschichts-Reise zurück.

Atbash

exakter ATBaSch
bezeichnet das Prinzip der Chiffrierung
diente der Suche nach versteckten Nachrichten in religiösen Texten
hebräisches Alphabet umfasste 22 Buchstaben
aus 1. Buchstaben (**A**leph) wird der letzte (**T**aw)
aus dem zweiten (**B**eth) wird der vorletzte (**Sch**in)
usw. usf.
praktisch wird das Alphabet umgedreht als Ersetzungs-Alphabet verwendet

typische Substitutions-Chiffre
es werden Zeichen vertauscht (substituiert)
monoalphabetisch, da Verschlüsselungs- und Entschlüsselungs-Methode gleich sind

wendet man das Atbasch-Codierung auf das lateinische Alphabet an, dann ergibt sich folgendes Schema

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Geheimalph.	**z**	**y**	**x**	**w**	**v**	**u**	**t**	**s**	**r**	**q**	**p**	**o**	**n**	**m**	**l**	**k**	**j**	**i**	**h**	**g**	**f**	**e**	**d**	**c**	**b**	**a**

zu beachten ist hierbei, dass es in der einfachen (- mehr theoretischen -) Kryptologie üblich ist, auf Leerzeichen und andere Zeichen und Ziffern zu verzichten. Wir nennen diese minimale Alphabet hier lateinisches Kryptographie-Alphabet.
häufig werden auch die Buchstaben i und J zusammengefasst
Fehlen Zeichen in der Nachricht oder sind Trennungen unbedingt notwendig, wir das X benutzt

Klartext schreiben wir immer mit Groß-Buchstaben
ev. wird auch die Textfarbe **grün** benutzt
Geheimtexte werden mit (**roten**) Klein-Buchstaben notiert

mechanische Chiffrier-Scheibe für ATBaSch
Q: de.wikipedia.org (Hubert Berberich)

Verschlüsselung und Entschlüsselung sind beim ATBaSch gleich, das heißt auch, das wir den Geheimtext wie einen Klartext behandeln können. Nach dem Verschlüsseln dieses (geheimen) Klartext erhalten wir den richtigen Klartext zurück. Wir sprechen auch von einer symmetrischen Verschlüsselung.
die Prinzipien lassen sich dann schnell auf größere Alphabete mit viel mehr Zeichen anwenden
in solche Alphabete gehören dann auch Ziffern, Sonderzeichen und Symbole
dadurch wird das Knacken des Codes deutlich schwieriger

Aufgaben:

1. ***Verschlüssele mit ATBaSch und dem lateinischen Kryptographie-Alphabet den Satz:***

   Es gibt viele geheime Botschaften.

2. ***Entschlüssele den Satz:***

   pibkgltizksrvdriwmrvozmtdvrort.

   Als Verschlüsslungs-Verfahren wird das klassische ATBaSch und das lateinische Kryptographie-Alphabet benutzt.

3.

Weiterentwicklungen der ATBaSch-Methode

ALBaM

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13
	A	B	C	D	E	F	G	H	I	J	K	L	M
	Z	Y	X	W	V	U	T	S	R	Q	P	O	N

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13
Klaralphabet	A	B	C	D	E	F	G	H	I	J	K	L	M
Geheimalph.	z	y	x	w	v	u	t	s	r	q	p	o	n
Geheimalph.	a	b	c	d	e	f	g	h	i	j	k	l	m
Klaralphabet	Z	Y	X	W	V	U	T	S	R	Q	P	O	N

(c,p) 2015 – 2020 lsp: dre

CÄSAR-Chiffre

Gaius Julius CÄSAR (100 – 44 v.u.Z) benutzte für seine Geheimbotschaften einen genial einfachen und scheinbar auch sehr sicheren Code. Es nicht bekannt geworden, dass der Code zu CÄSARS Zeiten jemals geknackt wurde.
Der CÄSAR-Code bestand darin ein Zahl zur Verschiebung der beiden Alphabete zueinander zu benutzen. Julius CÄSAR benutzte wohl immer die 3. Selbst schon die Verschiebung um eine Position bringt einen Buchstaben-Salat hervor.

Beispiel: CÄSAR3

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Geheimalph.	**y**	**z**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**	**u**	**v**	**w**	**x**

Beispiel: CÄSAR7

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
Geheimalph.	**u**	**v**	**w**	**x**	**y**	**z**	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**	**q**	**r**	**s**	**t**

ATBaSch ist praktisch eine spezieller CÄSAR-Code. Bei ihm wird das Geheim-Alphabet nur umgedreht (invertiert).

notwendiges Material:
Codier-Scheibe für CÄSAR-Codierung oder 2 Papierstreifen mindestens 1 cm dick und 40 cm lang; größere Büro-Klammern

Vorbereitung (Herstellung von Codier-Streifen):

1. *Teile den Streifen in 0,75 cm breite Abschnitte! Schreibe dann jeweils zweimal das Kryptographie-Alphabet hintereinander auf die Streifen! Auf einem Streifen wird das erste Alphabet auch durchnummeriert!*

Aufgaben:

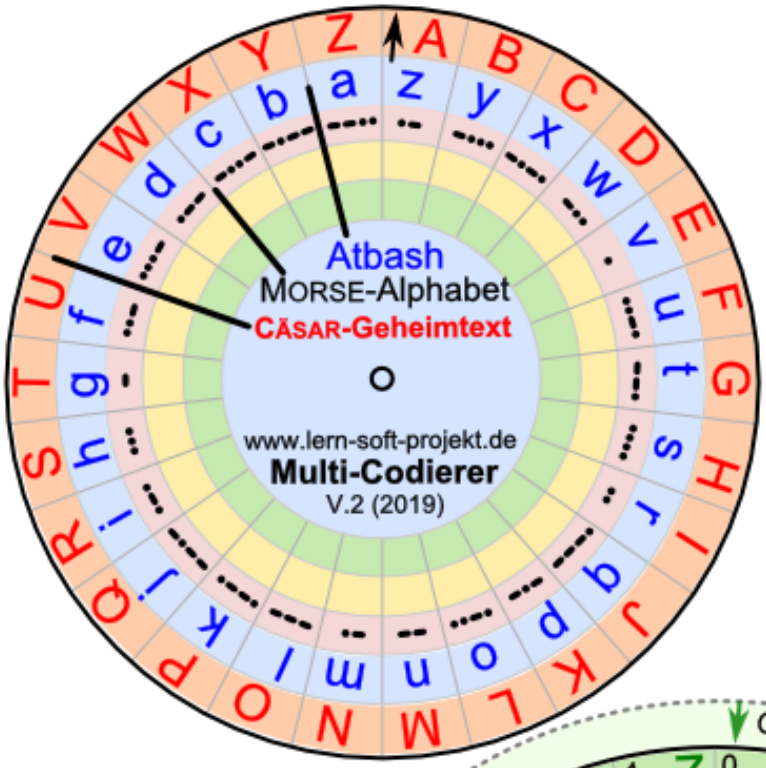
1. *Verschlüssele mit CÄSAR4 und dem lateinischen Kryptographie-Alphabet den Satz:*

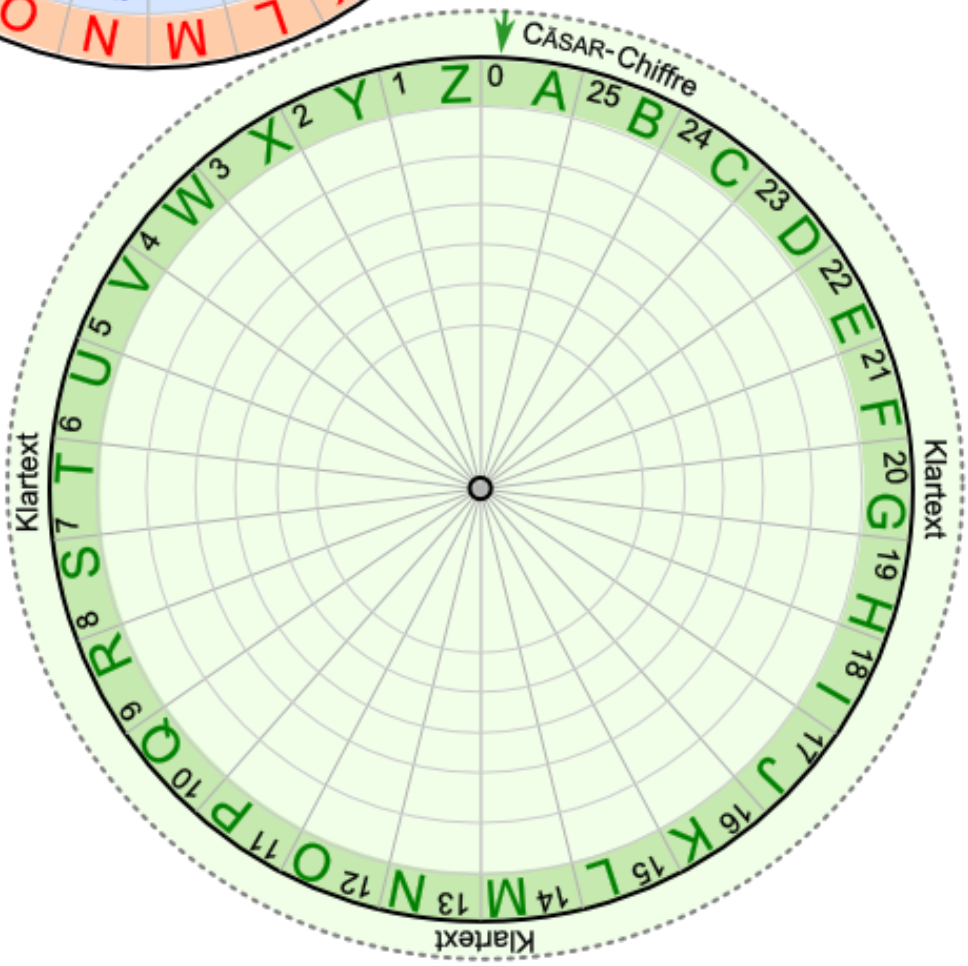
Es gibt viele geheime Botschaften.

2. Entschlüssele den Satz:

Als Verschlüsslungs-Verfahren wird das klassische CÄSAR16 und das lateinische Kryptographie-Alphabet benutzt.

3.





Ev. diese Seite farbig ausdrucken und laminieren.
Scheiben ausschneiden und ev. nicht-laminierte Scheiben auf Karton oder dünne Pappe aufkleben. In der Mitte ein Loch eindrücken und die Scheiben mit einem größeren Druckknopf oder einer Musterbeutel-Klammer verbinden.

Bedienung der Multi-Kodier-Scheibe

Z.B. für die CÄSAR-Verschlüsselung zuerst den schwarzen Pfeil (auf rötlichem Geheimtext) auf die richtige Chiffre-Nummer einstellen! Dann Buchstaben-weise den grünlichen Klartext in den rötlichen Geheimtext – oder umgedreht – übertragen.

Auf laminierten Kodier-Scheiben können eigene Geheim-Alphabete in die freien Ringe geschrieben werden. Mit Folienstiften sind die Beschriftungen gut haltbar. Zum Entfernen reicht ein mit Brennspiritus angefeuchteter Lappen (bzw. eine Ecke davon).

Weiterentwicklungen der CÄSAR-Chiffrierung

Verwendung eines Schlüsselwortes

Schlüsselwort sollte aus möglichst verschiedenen Buchstaben bestehen – also ohne doppelte Buchstaben. Kommen Buchstaben doppelt vor, dann werden die Wiederholungen vor dem Einsetzen als Schüsselwort rausgestrichen.
Aus SEELENVERKAEUFER wird dann SELNVRKAUF.

Beispiel: CÄSAR mit Schlüsselwort: BUCHLISTE

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Geheimalph.	b	u	c	h	l	i	s	t	e	a	d	f	g	j	k	m	n	o	p	q	r	v	w	x	y	z
	Schlüssel																									

Es tritt hier das Problem auf, das am Ende die Buchstaben nicht selten mit sich selbst verschlüsselt werden. Das ist aber insofern nicht ganz so dramatisch, da diese Buchstaben in Wörtern sowieso recht selten vorkommen.
Diese Unschönheit kann man aber umgehen, wenn man zum ursprünglichen Verfahren mit der verschiebung zurückkehrt und zusätzlich das Schlüsselwort verwendet.

Verwendung von Schlüsselwort und Verschiebung

Beispiel: CÄSAR7 mit Schlüsselwort BUCHLISTE

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Geheimalph.	r	v	w	x	y	z	b	u	c	h	l	i	s	t	e	a	d	f	g	j	k	m	n	o	p	q

Verwendung eines zufällig geordneten Zeichensatz

Statt der alphabetischen Folge benutzt man nun einen zufällig geordneten Zeichensatz. Dieser muss dem Dechiffrierer auch bekannt sein.
Auf diesen Zeichensatz wird nun die klassische CÄSAR-Methode angewendet. D.h. wir verschieben um sosundsoviele Zeichen.

Beispiel: CÄSAR7 auf ein Zufalls-Alphabet

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	K	T	G	E	P	S	W	Y	J	Q	H	Z	I	A	V	O	R	C	M	N	U	X	B	D	F	L
Geheimalph.	u	x	b	d	f	l	k	t	g	e	p	s	w	y	j	q	h	z	i	a	v	o	r	c	m	n

Natürlich ließe sich die Verschlüsselung auch noch verstärken, wenn man zusätzlich noch ein Schlüsselwort benutzt. Dann wäre es selbst für alle anderen Empfänger – die mit dem

gleichen zufälligen Zeichensatz arbeiten – nicht ganz so einfach den Geheimtext zu entziffern.
Eine weitere Möglichkeit ist die Verwendung von zwei unterschiedlichen Zufalls-Alphabeten für den Klartext und den Geheimtext.

Beispiel: CÄSAR7 auf zwei verschiedenen Zufalls-Alphabeten

Symbole lfd. Nr.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Klaralphabet	V	J	N	X	T	G	Z	K	O	B	S	R	A	Y	E	I	H	P	D	F	C	U	L	Q	W	M
Geheimalph.	y	a	k	q	b	m	p	s	h	d	c	l	x	j	v	z	u	n	t	r	w	g	f	i	o	e

Auf die Verschiebung könnte man eigentlich verzichten.

Aufgaben:

1. *Erzeuge mittels einer Tabellkalkulation (z.B.: EXCEL oder CALC) ein Zufalls-Alphabet!*
   a) *Öffne eine neue Tabelle und speichere sie unter "Zufallsalphabet" ab!*
   b) *Gebe z.B ab Zeile 5 in der Spalte A die Liste des regulären Alphabetes ein! Kopiere diese Liste auch in Spalte D!*
   c) *In Spalte B daneben eine Liste von Zahlen von 1 bis 26 als Positions-Angabe eingeben!*
   d) *Mit Hilfe der Funktion ZUFALLSZAHL() werden in der Spalte C Zufallszahlen zwischen 0 und 1 erzeugt!*
   e) *Markiere nun die Zufallszahlen und das kopierte Alphabet (in Spalte D) und wähle "Daten sortieren"!*
   f) *bei einer Wiederholung von e) wird eine neue Zufallsfolge erzeugt.*
2. *Verschlüssele den folgenden Text mit dem obigen CÄSAR-Verfahren mit einem Zufalls-Alphabet!*

3. *Versuche den entstandenen Geheimtext mit einem Knack-Programm für CÄSAR zu dechiffrieren!*
4. *Dechiffriere den Geheimtext:*
   mit dem obigen CÄSAR7-Verfahren und den zwei angegebenen Zufalls-Aphabeten

Verschlüsselung mit mehreren Schlüssel-Alphabeten

Das große Problem der einfachen Substitutions-Verfahren ist immer die mögliche Krypto-Analyse über die Buchstaben-Häufigkeit.
Johannes TRITHEMIUS (1462 – 1516) erstellte für sein Verschlüsselungs-Verfahren eine sogeannte Transpositions-Tabelle – diese nannte er **Recta transpositionis tabula** oder kurz auch **Tabula recta**. In der originalen Tabelle fehlen die Buchstaben **j** und **v**, da im Mittelalter in der deutschen Sprachen **u** und **v** sowie **i** und **j** nicht unterschieden wurden. Wir nehmen hier eine an unser heutiges Alphabet angepasste Tabelle:

		G	**E**	**H**	**E**	**I**	**M**	**B**	**L**	**E**	**I**	**B**	**T**	**G**	**E**	**H**	**E**	**I**	**M**	X	X	X	A	B	C	D
A	1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	25	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	26	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

h	**g**	**k**	**i**	**n**	**s**	**i**	**t**	**n**	**s**	**m**	**f**	**t**	**s**	**w**	**u**	**z**	**e**	q	r	s	w	y	a	c

Man kann gut erkennen, dass ein Buchstabe jedes Mal ein neues Geheim-Zeichen bekommt. Obwohl wir fünfmal ein E im Klartext hatten, ergibt sich jedes Mal ein anderes eheim-Symbol. Anders herum kann man aus dem mehrfachen Auftreten eines Geheim-Symbol's nicht auf den Klartext-Buchstaben zurückschließen. Wir haben im Geheimtext z.B. zweimal ein i. Jedes Mal war es aber ein anderer Klartext-Buchstabe, der verschlüsselt wurde.
Im Prinzip benutzte TRITHEMIUS als Schlüssel den Klartext. Mit anderen Worten: er verschlüsselte einen Text mit sich selbst.
Daraus ergibt sich ein Problem: ein Buchstabe wird niemals durch sich selbst kodiert. Dies bietet eine Chance, den Code zu knacken.
Die Verwendung von Füllzeichen ist bei TRITHEMIUS eher ungünstig. Benutzt man das gleiche Füllzeichen (s. oben: XXX), dann ergibt sich eine alphabetische Symbolfolge. Die Verwendung von fortlaufenden Buchstaben (s. oben: ABCD) erzeugt auch eine charakteristische (springende) Symbol-Folge.
Der Franzose Blaise DE VIGINÉRE (sprich: de wischineer,) entwickelte eine ähnliche Chiffre. Er ordnete den Buchstaben in Abhängigkeit von ihrer Position im Klartext unterschiedliche

Chiffren zu. Nehmen wir z.B. das Schlüsselwort: **geheim**, dann wird der erste Buchstabe des Klartextes mit CÄSAR**7** verschlüsselt, weil **g** an der **7**. Position im Alphabet steht. Der zweite Buchstabe wird dann mit CÄSAR5 (e ist an der 5. Position) usw. verschlüsselt. Am Ende des Schlüsselwortes beginnt man wieder von vorn.

		K	**R**	**Y**	**P**	**T**	**O**	**G**	**A**	**F**	**I**	**I**	**S**	**T**	**S**	**C**	**H**	**O**	**N**	**T**	**O**	**L**	**L**	X	X	X
A	1	**G**	**E**	**H**	**E**	**I**	**M**	G	E	H	E	I	M	G	E	H	E	I	M	G	E	H	E	I	M	G
B	2	H	F	I	F	J	N	H	F	I	F	J	N	H	F	I	F	J	N	H	F	I	F	J	N	H
C	3	I	G	J	G	K	O	I	G	J	G	K	O	I	G	J	G	K	O	I	G	J	G	K	O	I
D	4	J	H	K	H	L	P	J	H	K	H	L	P	J	H	K	H	L	P	J	H	K	H	L	P	J
E	5	K	I	L	I	M	Q	K	I	L	I	M	Q	K	I	L	I	M	Q	K	I	L	I	M	Q	K
F	6	L	J	M	J	N	R	L	J	M	J	N	R	L	J	M	J	N	R	L	J	M	J	N	R	L
G	7	M	K	N	K	O	S	M	K	N	K	O	S	M	K	N	K	O	S	M	K	N	K	O	S	M
H	8	N	L	O	L	P	T	N	L	O	L	P	T	N	L	O	L	P	T	N	L	O	L	P	T	N
I	9	O	M	P	M	Q	U	O	M	P	M	Q	U	O	M	P	M	Q	U	O	M	P	M	Q	U	O
J	10	P	N	Q	N	R	V	P	N	Q	N	R	V	P	N	Q	N	R	V	P	N	Q	N	R	V	P
K	11	Q	O	R	O	S	W	Q	O	R	O	S	W	Q	O	R	O	S	W	Q	O	R	O	S	W	Q
L	12	R	P	S	P	T	X	R	P	S	P	T	X	R	P	S	P	T	X	R	P	S	P	T	X	R
M	13	S	Q	T	Q	U	Y	S	Q	T	Q	U	Y	S	Q	T	Q	U	Y	S	Q	T	Q	U	Y	S
N	14	T	R	U	R	V	Z	T	R	U	R	V	Z	T	R	U	R	V	Z	T	R	U	R	V	Z	T
O	15	U	S	V	S	W	A	U	S	V	S	W	A	U	S	V	S	W	A	U	S	V	S	W	A	U
P	16	V	T	W	T	X	B	V	T	W	T	X	B	V	T	W	T	X	B	V	T	W	T	X	B	V
Q	17	W	U	X	U	Y	C	W	U	X	U	Y	C	W	U	X	U	Y	C	W	U	X	U	Y	C	W
R	18	X	V	Y	V	Z	D	X	V	Y	V	Z	D	X	V	Y	V	Z	D	X	V	Y	V	Z	D	X
S	19	Y	W	Z	W	A	E	Y	W	Z	W	A	E	Y	W	Z	W	A	E	Y	W	Z	W	A	E	Y
T	20	Z	X	A	X	B	F	Z	X	A	X	B	F	Z	X	A	X	B	F	Z	X	A	X	B	F	Z
U	21	A	Y	B	Y	C	G	A	Y	B	Y	C	G	A	Y	B	Y	C	G	A	Y	B	Y	C	G	A
V	22	B	Z	C	Z	D	H	B	Z	C	Z	D	H	B	Z	C	Z	D	H	B	Z	C	Z	D	H	B
W	23	C	A	D	A	E	I	C	A	D	A	E	I	C	A	D	A	E	I	C	A	D	A	E	I	C
X	24	D	B	E	B	F	J	D	B	E	B	F	J	D	B	E	B	F	J	D	B	E	B	F	J	D
Y	25	E	C	F	C	G	K	E	C	F	C	G	K	E	C	F	C	G	K	E	C	F	C	G	K	E
Z	26	F	D	G	D	H	L	F	D	G	D	H	L	F	D	G	D	H	L	F	D	G	D	H	L	F

q	**v**	**f**	**t**	**b**	**a**	**m**	**e**	**m**	**m**	**q**	**e**	**z**	**w**	**j**	**l**	**w**	**z**	**z**	**s**	**s**	**p**	**f**	**j**	**d**

Die Stärke dieses Verfahrens wird schon bei den Füllzeichen am Ende sichtbar. Kein X wurde gleichartig oder mit einer Buchstabenfolge verschlüsselt. Noch besser wäre es natürlich, ganz auf die Füllzeichen zu verzichten, da sie ein guter Angriffs-Punkt für eine Krypto-Analyse sind. Wenn man weiss, dass am Ende sehr wahrscheinlich Xe stehen, dann kann bei genügend Geheimtexten das Passwort teilweise geknackt werden.

Heute wissen wir, wenn man einen zum Klartext gleichlangen Schlüssel verwendet und diesen nur ein einziges Mal benutzt, dann ist die VIGINÉRE-Chiffre unknackbar. Außer natürlich man versucht es mit einem Brute-Force-Angriff.

Man braucht also auch heute keine komplizierte Technik oder gar Computer, um absolut sicher Texte zu verschlüsseln. Das einzige Problem ist der Transport der Schlüssel und die Absprache, welcher Schüssel genau benutzt werden soll.

Werden allerdings kürze Schlüsselwörter benutzt, dann kann der Geheimtext ev. entschlüsselt werden. Dabei ermittelt man zuerst mit dem KASISKI-Test die wahrscheinliche Schlüssellänge. Dann zerlegt man den Text in die Teile, die mit dem gleichen Schlüssel-Zeichen codiert wurden. Sie werden einer Häufigkeits-Analyse unterzogen. Ab hier ist es dann nur noch Rechen- oder Such-Aufwand. Gute Code-Knacker erschließen daneben noch das verwendeten Schüsselwort.

(c,p) 2015 – 2020 lsp: dre

Aufgaben:

1. *Verschlüssele den folgenden Text mittels VIGENÉRE-Verfahren und dem Schlüsselwort "DAMENSCHUH"!*

   Mein Geheimnis ist: Ich mag gerne Tee.

2. *Denke Dir nun ein neues Schlüsselwort mit mindestens 8 Zeichen aus und verschlüssele damit einen Text von maximal 25 Zeichen! Die nicht benötigten Zeichen werden mit X aufgefüllt.*
3. *Gebe den Geheimtext und den Schlüssel an Deinen Nachbarn weiter! Dechiffriere den Geheimtext Deines Nachbarn!*

für Experten und zum Knobeln:

4. *Der folgende Text wurde mittels VIGENÉRE-Verfahren verschlüsselt. Die letzten – nicht gebrauchten – Zeichen wurden mit X aufgefüllt. Wie lautet das Passwort und wie der Klartext?*

Verbesserungen

Giovan Battista BELLASO (~ 1505 ~ 1568/81) benutzte statt der klassischen (sortierten Alphabete z.T. gewürfelte Symbol-Listen (1555). So z.B. für die Buchstaben A und R die folgende Liste.

A R	→	r	m	d	a	c	n	e	u	p	s	b	t	d	f	g	e	h	l	x	o	y	z

Aufgaben:

1. *Erstelle Dir eine eigene Liste von 5 gewürfelten deutschen Alphabeten! Ordne Sie den möglichen Schlüssel-Buchstaben zu, so dass eine private Verschlüsselungs-Tabelle entsteht!*
2. *Verschlüssele nun mit Deiner Tabelle und einem Schlüsselwort einen kurzen Text!*
3. *Tausche den Geheimtext und das Schlüsselwort auf zwei verschiedenen Wegen (schriftlich, mündlich, per eMail, …) mit einem Kursteilnehmer!*
4. *Entschlüssele die getauschte Nachricht!*

Von Sir Francis BEAUFORT (1774 – 1857) kam eine weitere Verbesserung. Durch die Verwendung eines umgekehrten (revertierten) Alphabet's kann man erreichen, dass nun nicht mehr zwischen dem Verfahren der Ver- und Entschlüsselung unterschieden werden muss. Für beide Vorgänge kann das gleiche Vorgehen benutzt werden.

(böse) Frage zwischendurch:
Durch was ist BEAUFORT heute noch jedem bekannt?

Eine Tabelle für das BEAUFORT-Verfahren sähe dann z.B. so aus:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B
2	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C
3	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D
4	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E
5	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F
6	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G
7	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H
8	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I
9	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J
10	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K
11	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L
12	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M
13	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O	N
14	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P	O
15	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q	P
16	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R	Q
17	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S	R
18	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T	S
19	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U	T
20	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V	U
21	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W	V
22	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X	W
23	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y	X
24	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z	Y
25	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A	Z
26	Z	Y	X	W	V	U	T	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A

Enigma – eine Legende und ein Mysterium

Bei der Enigma – der wohl berühmtesten Chiffrier-Maschine – aus dem 2. Weltkrieg wird das Ersetzen von Zeichen noch verstärkt. Man könnte sich das so vorstellen, dass man den ersten Buchstaben mit einem CÄSAR-Verfahren chiffiert, das mit zwei zufälligen Alphabeten arbeitet. Für den zweiten Buchstaben benutzt man nun zwei neue Zufalls-Alphabete und eine verschiebung, die sich aus dem ersten verschlüsselten Zeichen ergibt. Mit dem dritten, vierten und ev. auch dem Fünften Zeichen verfährt man genauso. Da beginnt man wieder mit dem ersten Zufalls-Alphabet-Paar – allerdings mit einer Verschiebung. Was dabei rauskommt ist der absolute Buchstaben-Salat.
In der orginal-Enigma-Maschine übernehmen Walzen die Rolle der Zufalls-Alphabet-Paare. Je nach Geheimhaltungs-Stufe wurden Maschinen mit 3 oder 4 Walzen verwendet.
Insgesamt gab es 5 verschiedene Walzen in in belieber Reihenfolge gesteckt werden konnten. Die Walzen-Reihenfolge und die Anfangs-Positionen wurden jeden Tag um Mitternacht gewechselt.
Die gesamte Verschlüsselung durch die Maschine war noch komplizierter, da praktisch eine zweimalige Chiffrierung für einen Buchstaben durchgeführt wurde. Dazu kamen Steck-Verbinder, die nochmals neue Buchstaben-Zuordnungen ermöglichten.
Für eine Enigma mit 3 von 5 Walzen, 1 von 2 Umkehrwalzen und 10 Steck-Verbindungen ergeben sich 214'917'374'654'501'238'720'000 mögliche Schlüssel.
Nehmen wir an die Dechiffrierer würden alle Schlüssel auf einen Geheimtext ausprobieren und für jede Prüfung nur 1 Sekunde benötigen, dann wären sie damit 6'814'985'243'991'033 Jahre beschäftigt. Das sind rund 7 Millionen Milliarden Jahre. Unser Weltall ist wahrscheinlich mal gerade 14 Milliarden Jahre alt. So wird das also nichts!
Na gut, ein Computer rechnet und prüft viel schneller. Um im Lebensalter unseres Sonnensystems fertig zu werden, muss er schon mehr als 1 Millionen Schlüssel pro Sekunde prüfen. Das ist schon eine reife Leistung für einen Computer. Dabei sollte aber immer bedacht werden, dass nach 24 Stunden ein neuer Schlüssel gilt. Das heißt, man bräuchte 2'487'469'614'056 Computer, die parallel arbeiten müssten. Das macht die theoretische Stärke der Enigma vielleicht klar.
Durch einige kryptographische Schwächen in der Konstruktion der Maschien, wie auch in der praktischen Verwendung konnte die Enigma von Alan TURING geknackt werden. Seine TURING-Bombe genannte Maschine reduzierte die Schüssel auf eine Zahl, die ein Brechen eines Schlüssels innerhalb eines Tages (praktisch weniger als 10 Stunden) ermöglichte.
Zum Erfolg haben aber auch andere Forscher und Agenten beigetragen. Ohne die Eroberung von Code-Büchern und Enigma-Maschinen wäre die Entschlüsselung im 2. Weltkrieg wohl nicht möglich gewesen.
Ähnlich gebaute Enigma-Maschinen werden immer noch von einigen Ländern zum Verschlüsseln von diplomatischer Post verwendet. Das Verfahren ist seit vielen Jahren auch auf PC's verfügbar. Aber Computer und Programme sind manipulierbar, hinterlassen elektromagnetische Spuren und deshalb nicht unbedingt geeignet.
Als Konsequenz aus den gelungenen Knack-Versuchen der Enigma wurde das KERCKHOFFsche Prinzip bestätigt, welches dieser schon 1883 formulierte:

> **Die Sicherheit eines Kryptosystems darf nicht von der Geheimhaltung ihres Verfahrens abhängen. Die Sicherheit basiert immer nur auf der Geheimhaltung des Schlüssels.**

Viele Abwandlungen der Enigma nach dem II. Weltkrieg machten das System viel sicherer. U.a. wurde mehr Walzen eingesetzt. Solche Maschinen lassen sich **praktisch** nicht mehr knacken, selbst wenn man die gesamte Rechen-Leistung auf der Erde dafür benutzen würde.

Links:

http://users.telenet.be/d.rijmenants/en/enigmasim.htm (Enigma-Simulator; engl.)

Das Geheimnis sicherer Passwörter

> Schaut eine Blondine einem Nerd bei der Passwort-Eingabe über die Schulter. "Du hast ja das gleiche Passwort wie ich – sechs Sternchen.

So ein Beobachten bei der Passwort-Eingabe ist sehr unhöflich. Man sollte einen solchen Beobachter durch Anzeigen von Unbehagen (z.B. durch Ansehen oder betontes Warten vor der Eingabe) klar machen, dass er sich jetzt mal bitte wegdrehen möchte. Wenn man selbst neben einem Passwort-Eingeben sitzt, dann sollte man sich gleich unaufgefordert wegdrehen. Das erhöht das Vertrauen.
Wir haben gesehen, dass man mit sehr langen Passwörtern sehr sicher verschlüsseln kann. Es ist aber sehr schwierig diese zu handhaben. Aus praktischen Gründen brauchen wir also eher kurze Passwörter, die aber trotzdem möglichst sicher sein sollen.
Es gibt verschiedene Regeln und Empfehlungen, wie man sich relativ sichere Passwörter kreieren kann. Die Länge eines Passwortes sollte also immer möglichst lang sein, wobei ein Kompromiss zwischen der Länge und der Merkbarkeit gefunden werden muss.
Zum Zweiten schließen wir gängige Wörter der eigenen und der englischen Sprache aus. Nichts ist einfacher, als einfach alle Wörter eines Wörterbuch's (z.B. des Duden's) auszuprobieren. Solche Knackversuche nennt man Brute Force. Man versucht mit roher Gewalt ein Passwort zu erhalten.
Als Worte kann man sich auch Dinge ausdenken, die es in der reelen Welt nicht gibt. Da bieten sich Tier- oder Pflanzen-Namen an. Einen Tulpen-Adler oder eine Berg-Makrele gibt es wohl in keinem Wörterbuch. Auch andere Dinge aus anderen Lebensbereichen – also solche mit denen man normalerweise nichts zu tun hat – bieten sich an: Mercedes-Blüte, Zangen-Zarge od. Scheren-Poller sind da gute Beispiele.
Häufig verlangen die Passwort-Kontroll-Mechanismen von Webseiten usw., dass man z.B. sowohl Groß- und Klein-Buchstaben verwenden muss. Natürlich sind damit nicht die üblichen Groß-Schreibungen von Substantiven oder Eigennamen gemeint. Vielmehr soll man die Groß-Buchstaben da einsetzen, wo sie normalerweise nicht vorkommen.
Würden wir z.B. das Passwort "TulpenAdler" verbessern wollen, dann bieten "tUlPeNaDlEr" oder "TUlpENadLEr" eine deutlich höhere Sicherheit. Diese Wörter stehen in keinem Wörterbuch und ein rumprobieren würde beachtlich mehr Rechenaufwand beim Passwort-Knacken bedeuten.
Eine weitere Möglichkeit ist es, einzelne Buchstaben durch Zahlen zu ersetzen. Da bieten sich z.B. die 1 als Ersatz für ein i an. Der Vorteil ist, dass man solche Passörter immer noch gut lesen kann. Aus "TulpenAdler" könnte so z.B. "Tu1p3n4d13r" werden. Computer-Programme können diese Ersetzungen natürlich auch in ihre Knack-Algorithmen übernehmen, was diese Austausche etwas nachteilig macht. Trotzdem verbessern die gemeinsame Nutzung von Groß- und Klein-Buchstaben nochmals nenneswert.
Statt Wörtern empfehlen Kryptologen lieber Sätze (Sprichwörter, Redewendungen usw.) zu nehmen. Von den einzelnen Worten nimmt man dann nur die ersten Buchstaben. Aus dem Sprichwort:

Wer **a**nderen **e**ine **G**rube **g**räbt, **f**ällt **s**elbst **h**inein.

wird dann zuerst einmal:

WaeGgfsh

Vielfach sind auch Sonderzeichen in Passwörter zugelassen, gewünscht oder gefordert. Im Beispiel könnte man die Satz-Zeichen mit einbeziehen:

WaeGg,fsh.

Kombinieren wir jetzt noch unsere ersten Empfehlungen mit hinein, dann könnte schon:

W4e69,f5h.

erhalten. Und mit weiteren Sonderzeichen oder speziellen Ersetzungen käme man z.B. auf:

W@e&9,f5h.

Echte sichere Passwörter sind deutlich länger als 12 Zeichen und zufällig zusammengewürfelt.
Aber Achtung! Zu komplizierte Passwörter verleiten zum Aufschreiben oder Anlegen von Spickzetteln. Ein noch so sicheres Passwort unten auf die Tastatur geklebt oder gar als Post-it am Monitor sind dann völlig umsonst. Da kann man auch gleich "123456" nehmen.

Aufgaben:

1. *Formuliere Regeln zur Passwort-Kreation in Form der biblischen Gebote ("Du sollst …! / Du sollst nicht …!)*
2. *Recherchiere die am häufigsten benutzten Passwörter!*
3. *Erstelle aus dem Sprichwort "Ein Unglück kommt selten allein." mindestens fünf unterschiedliche Passwörter! Sortiere sie nach Stärke (gegenüber Brute Force und Zeichen-Ausprobier-Algorithmen)!*

Mein ganz geheimes Passwort ist übrigens "falsch". Immer, wenn ich dieses vergessen habe und irgend ein falsches eingebe, dann meldet sich das Anmeldesystem zurück und hilft mir mit der richtigen Antwort: "Ihr Passwort ist falsch."

Steganographie – die Kunst des Versteckens

Nachricht auf Glatze tätowieren und nachdem die Haare nachgewachsen sind, die Kurier losschicken, beim Empfänger wir die Haarpracht wieder entfernt und man kann die Nachricht ablesen
(Einmal-Code, symmetrisch)

präparierte Wachstafeln
Wachskugeln verschlucken
doppelter Boden in Koffern oder Paketen
Schmugel-Fächer in Thermoskannen, Schuh-Absätzen usw.
Mikro-Punkt (fotographische Verkleinerung eines Textes bis auf die Größe eines Punktes in einem (öffentlichen) Text. Punkt wird bei Empfänger wieder vergrößert, z.B. mit einem Mikroskop oder Foto-Vergrößerungungs-Apparat.)

Wasserzeichen auf Banknoten oder in Bildern, damit Urheber ihre Bilder wiedererkennen können und z.B. eine mißbräuchliche Nutzung dokumentieren können

Stenographia ist eine im 17 Jhd. praktizierte Technik, um Nachrichten mittels Licht zu übertragen. Für die Stenographia wird eine Nachricht auf / in einen Hohlspiegel geschrieben. Mit einer starken Lichtquelle wird ein Lichtstrahl erzeugt, der auf den Empfänger gerichtet wird. Im Empfänger-Holspiegel kann die Nachricht dann gelesen werden.

Kurzschriften, wie z.B. Steno – eigentlich Stenographie – werden heute kaum noch verwendet. Statt der Wörter werden nur Linien-Schwünge notiert. Das geht extrem schnell, muss aber gelernt sein. Früher bei Sekretärinnen und Protokollanten üblich. Heute oft durch Diktier-Systeme ersetzt. Die Stenogramme bzw. aufgesprochenen Texte wurden dann später in Ruhe eingetippt.

modern Tintenkiller

einfach Informationen in gemalten Bildern verstecken
z.B. durch Aufmalen der Situation: Treff an der Bushaltestelle um 13 Uhr

z.B. durch Verstecken von Zeichen und Symbolen z.B. Buchstaben und Ziffern, ev. geordnet von unten nach oben oder rückwärts zu lesen
Buchstaben z.B. in Bild-Elementen etwas dicker zeichnen (ev. auch leicht drehen usw.)

Das Trägermedium wird Container genannt.
Steganogramm (auch verk.: Stegogramm) ist das fertige geheime Schreiben, Bild oder Video.

symmetrische Steganographie

asymmetrische Steganographie
auch Public-key-Steganographie

Grenzen zwischen kryptographie und Steganographie verschwimmen immer mehr, vor allem auch deshalb, weil beide Verfahren oft kombiniert werden. Nichts ist besser als eine Nachricht, die keiner als solche erkennt und dann auch noch unknackbar chiffriert ist.

Postkarten-Chiffren

Da Postkarten immer mehr oder weniger öffentliche Texte enthalten, muss eine darin versteckte Nachricht in einer nur den beiden Kommunikanten bekannten Form verschlüsselt werden.
Natürlich kann man z.B. die CÄSAR-Chiffre nutzen und schon wird es schwerer, die versteckte Nachricht zu entschlüsseln. Der Nachteil is dabei, dass praktisch jeder sieht, dass der Text wohl verschlüsselt ist. Das spornt Neugierige meist erst so richtig an. Besser ist es die Klar-Nachricht irgendwie unauffällig in den öffentlichen Postkarten-Text einzuarbeiten.
Z.B. könnte man einen Text mit verschiedenen Farb-Stiften schreiben. Nur eine Farbe enthält aber die verschlüsselte Nachricht. Die anderen Farben sind nur zur Tarnung.

Ältere "Postkarten"-Chiffren basierten darauf, dass immer nur der erste Buchstabe jedes Wortes genutzt wurde. Aus einem Klartext dann aber einen Steganogramm zu machen, welches wirklich unauffällig ist, kann schon recht schwierig werden.

Aufgaben:

1. *Erstelle für das Wörtchen "Steganogramm" ein Steganogramm im Erster-Buchstabe-Postkarten-Code!*
2. *Verschlüssle eine eigene Nachricht durch z.B. den Zweite-Buchstaben-Postkarten-Code! Tausche Dein Steganogramm mit dem eines Kurs-Teilnehmer's und entschlüssle dieses!*
3.

Eine weitere Möglichkeit sind auffällige Buchstaben. Diese könnten schief oder etwas höher oder niedriger geschrieben werden. Auch könnte man bei gedruckten Texten mit verschiedenen Schrift-Typen arbeiten. So könnte die geheime Nachricht mit Serifen-losen Buchstaben in einem "normalen" Text versteckt werden, der eben mit einer Serifen-Schrift gesetzt ist.

Anagramme
sind Schüttel-Wörter, also Neukombinationen der Buchstaben eines Klarwortes, wobei das Geheimwort auch wieder ein Wort der Sprache sein muss

so kann aus **LEBEN** das Wörtchen **nebel** werden
so kann aus **BESENSTIEL** das Wörtchen **liebesnest** werden

praktisch ist es die Ersetzung eines Wortes durch ein anderes
leicht zu knacken, da man sich nur Wort-Listen anlegen muss, die schon bekannte Wörter enthält
für komplexere Fachwörter kaum Anagramme zu finden
bei unbekannten Wörtern durchsucht man entweder das bekannte Wörterbuch (aller Wörter) der Sprache oder man generiert alle Kombinationen und prüft diese gegen die Wörter im Wörterbuch
erstere Variante wahrscheinlich effektiver
um die Suche in Wörterbüchern zu vereinfachen kann man mit Stellen-unanhängigen Hash-Werten arbeiten, aus den Buchstaben wird eine Kennzahl berechnet – z.B. wie eine Quer-summe – im Wörterbuch sind dann neben den Wörtern auch deren Hash's gespeichert. Zur Suche wird dann zuerst in den Hash-werten verglichen und bei einer Übereinstimmung detaulliert kontrolliert, ob das Wörtchen passt

Links:
http://www.sibiller.de/anagramme/ (etwas komplexerer Anagramm-Generator) (http://anagramm generator.de/)
https://www.wort-suchen.de/anagramm-generator/ (Anagramm-Generator)

Verschlüsselung und Verstecken mit Snap!

Chiffrierung
ASCII, UniCode – Umwandlung in Zahlen od. Symbole, … ineinander

Verschlüsselung
CÄSAR-Code

Steganographie
z.B. indem der Transparenz-Wert (normal immer 255 zum Verstecken genutzt wird)
s.a. Informatik-Kurs (OpenSAP) 1. Woche Unit 6 VS
… Verstecken könnte man noch mit Verschlüsseln kombinieren
… Verstecken erweitern auf Gültigkeit nur aller soundsovielten Pixel, die anderen werden zufällig belegt

komplexe Aufgaben (zu diesem Kapitel):

1.
2. Überlege Dir eine möglichst effektive Text-Übertragung mittels der "Blinzel"-Methode (nach DURAND)! Übertrage einen Text einmal über das normale Alphabet und einmal nach Deiner verbesserten Methode an Deinen Nachbarn. Zähle dabei mit, wieviele Buchstaben die Hilfsperson (Dein Nachbar) immer aufsagen muss, um einen Buchstaben zu übertragen!
3.

8.3. Nachrichten verstecken

historisch

auf Kopfhaut tätowierte Nachricht
Melder wird losgeschickt, wenn Haare nachgewachsen sind

Wachstafeln
Nachricht wird auf Träger-Material (z.B. Holz) eingeritzt
Brett mit Wachs-Schicht überzogen und eine Schein-Nachricht in das Wachs geritzt

maskierte Text / Geheimschriften / Jargon-Code
Kohle, Kies → Geld

Geheim-Tinten

doppelte Böden in Kisten / Truhen / Paketen
in Hochzeiten sogar in Briefumschlägen versteckte Nachrichten

Wasserzeichen
in Korrespondenzen
Banknoten

Mikropunkt

Machine Identification Code
fast unsichtbare Kennzeichnungen von Ausdrucken mit Informationen des Drucker's bzw. des Kopierer's

9. Daten erfassen und darstellen

Aufbau einer Tabellenkalkulation

- *die Begriffe Tabellenblatt, Zeile, Spalte und zelle sicher verstehen*
- *die Adressen von Zellen und Zellbereichen angeben und verwenden*

Daten verarbeiten

Es bieten sich zweispaltige Tabellen an, die Zuordnung einer einzigen Wertereihe zu einer Reihe von Argumenten beinhalten.

- *Daten in vorhandenen Tabellen interpretieren, verändern und ergänzen*
- *den Datentyp für eine Zellinhalt bestimmen*

Es sind die Datentypen Zeichenkette und Zahl zu unterscheiden.
nummerische Daten formatieren
Die SuS formatieren Zellinhalte als Währung oder als Zahl mit einer festen Anzahl von Dezimalstellen.

- *Tabellen planen und erstellen*

Das Augenmerk liegt bei der Struktur der Tabelle und der genauen inhaltlichen Bezeichnung der Datenreihen.

Diagramme

- *Diagramme erstellen und beschriften*

Es erfolgt eine Reduktion auf Balken-, Säulen und x-y-Diagramme. die Erstellung des Diagramms sollte mithilfe eines Diagrammassistenten erfolgen.

- *das Referenzprinzip beispielhaft erklären*

Das Referenzprinzip bedeutet hier, dass die Änderung der Daten in der Tabelle eine Anpassung des Diagramms nach sich zieht.

- *die Eignung von Diagrammdarstellungen bewerten*

Die Bewertung der Darstellungen sollte sowohl eigene als auch von anderen erstellte Diagramme umfassen und hinsichtlich der Korrektheit un der Eignung für den beabsichtigten Zweck erfolgen.

Formeln

- *Formeln mit Grundrechenoperationen entwickeln, implementieren und prüfen*

Die SuS beschreiben funktionale Abhängigkeiten zwischen Daten und entwickeln daraus Formeln unter Verwendung relativer Zellbezüge. Sie erläutern den Vorteil von Formeln gegenüber manuell berechneten, konstanten Werten.

Vorschläge zur inhaltlichen Vertiefung:

- *weitere Formatierungsoptionen*
- *benutzerdefinierte Datenformate*

Problem-Fragen für Selbstorganisiertes Lernen

Wer war zuerst da - die Textverabeitung, eMail-Programme, das Internet oder Tabellenkalkulationen?

Was unterscheidet Tabellen in einer Textverarbeitung von Tabellen in einer Tabellenkalkulation?

Was ist das Erfolgsgeheimnis von Tabellenkalkulationen?
Ist eine Tabellenkalkulation wirklich ein Werkzeug für "faule" Daten-Verarbeiter?

Wie kann man in Tabellenkalkulationen rechnen?

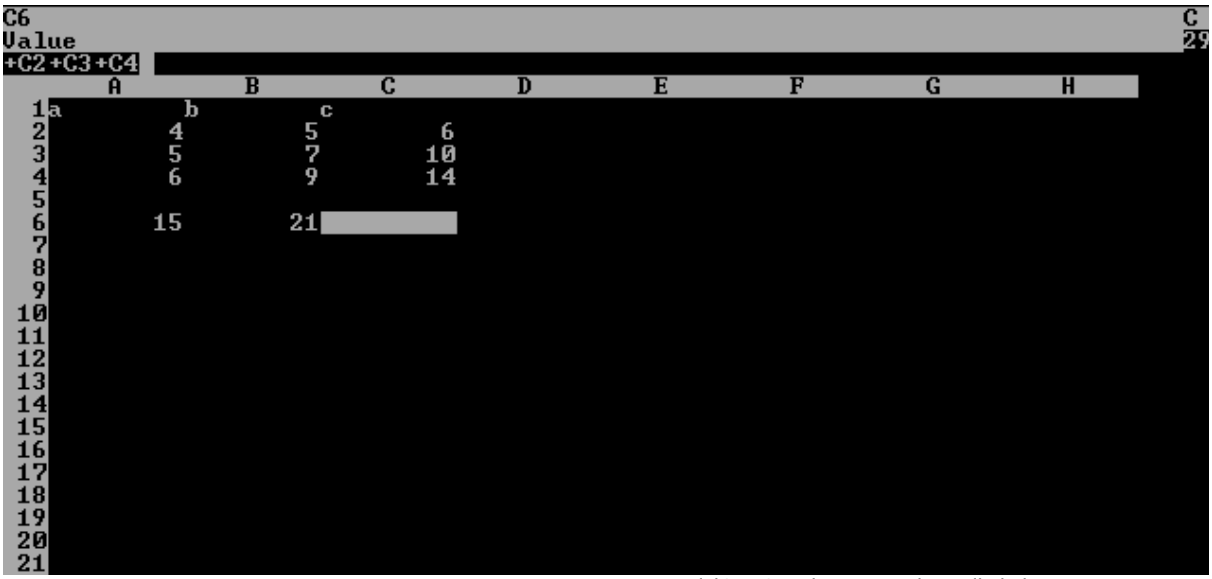
Ein Diagramm / Bild sagt mehr als 1000 Worte oder eine Tabelle – stimmt das?

Welche Arten / Typen von Diagrammen gibt es?
Kann man die Art des Diagramm frei wählen?

Lassen sich mit Tabellenkalkulationen und Diagrammen Daten und Präsentationen verfälschen? Ist immer alles wahr, was in Tabellen steht und in Diagrammen angezeigt wird?

moderne Adressierung mit Zell-Adresse (Zell-Name)
üblich Spalte mit Buchstabe(n) und die Zeilen mit Nummern
je nach Programm-Version und Bit-Breite (32 od. 64 bit) ergeben sich unterschiedliche Spaltenzahlen und Zeilenzahlen

im Schulbrereich spielt diese Grenze keine Rolle, so dass praktisch alle Tabellenkalkulationen diesbezüglich gleichwertig sind



VisiCALC – ein Urvater der Kalkulations-Programme

ältere / historische Zell-Adressierung
Zeilen und Spalten durchnummeriert
Zeile wird zuerst genannt, dann folgt die Spalte
Zell Adresse heißt in dieser Schreibweise z.B. Z6S2 (für B6)
auch heute noch als Option in EXCEL einstellbar (unübersichtllich und schwer verständliche Formeln)
gemeint sind dabei immer nur die anderen Zellen, hier wird der Weg zu den anderen zellen beschrieben, wieviel Zeilen / Spalten nach vorne / hinten muss man sich bewegen, um die gewünschte Zelle zu erreichen
intern werden Spalten und Zeilen auch heute immer noch als Nummern verarbeitet, weil dass für den Computer einfacher ist, als Buchstaben bzw. dern Kombinationen zu benutzen
für die Anzeige auf dem Bildschirm wird die Adresse umgerechnet (das kann der Computer viel besser als wir, wir verstehen dafür die Buchstaben-Zahlen-Adressen besser)

Daten in vorhandenen Tabellen interpretieren, verändern und ergänzen

Tabellen für Diagramme planen und erstellen

den Datentyp für einen Zellinhalt bestimmen

numerische Daten formatieren

Zellen-Formate

Objekte und Strukturen in der Tabellen-Kalkulation

drei Schichten (3-Ebenen-Modell (GIEDING 2003)) (hier von oben nach unten)

- Format-Ebene
- Werte-Ebene
- Formel-Ebene

sachlich logischer und informatisch besser strukturiert:

- Werte-Ebene (→ Eingabe)
- Formel-Ebene (→ Verarbeitung)
- Format-Ebene (→ Ausgabe)

ergibt auch eine logisch besser verständliche Kombination der Einzelebenen zu Zwischen-Lösungen beim Erstellen einer Tabellenkalkulation

Wie schon bei anderen Programm-Typen (Grapfik- und Textverarbeitungs-Programme) gibt es auch in der Tabelle-Kalkulation verschiedene **Objekte**, denen wieder **Attribute** (Eigenschaften) und **Methoden** (Funktionen, Arbeitsschritte) zugeordnet werden.

Objekt:			**Bemerkungen / Hinweise**
Kalkulations-Mappe			
→ Unterobjekt(e):			
• Tabelle			
• Diagramm			
•			
→ Attribut(e):		***Wert***	
• Dateiname	=	*Lineare Funktion*	dieses Attribut kann der Nutzer direkt ändern
• Dateityp	=	*XLSX*	
• geändert	=	*wahr*	dieses Attribut wird vom Programm verwaltet
• Schreibschutz	=	*falsch*	
• Passwort	=	*p6$Rq=%*	Passwort wird verschlüsselt abgespeichert
•	=		
→ Methode(n):			
• Neu erstellen			
• Speichern unter …			
• Speichern			
• Öffnen			
• Schließen			
•			

Objekt:			**Bemerkungen / Hinweise**
Kalkulations-Tabelle			
→ Unterobjekt(e):			
• Zelle			
• Diagramm			
• Textfeld			
•			
→ Attribut(e):		***Wert***	
• Seitenformat			
○ Seitengröße	=	*A4*	
○ Orientierung	=	*hoch*	
○ Randabstände			
▪ links	=	*2 cm*	
▪ rechts	=	*3 cm*	
▪ oben	=	*2 cm*	
▪ unten	=	*2 cm*	
• gefüllt / nächste Seite	=	*falsch*	
•	=		
→ Methode(n):			
• Neu erstellen			erzeugt eine neue Seite mit dem akt. Format
• Seitenformat festlegen / ändern			
• Ansicht aktualisieren			
• Druckvorschau erzeugen			
•			

Objekt:			Bemerkungen / Hinweise
Zelle			meint hier nur druckbare Zeichen
→ Attribut(e):		***Wert***	
• Zeichen-Code	=	*0041h*	Windows-interner Code für "**A**"
• Zeichenformat			
○ Schriftart	=	*Arial*	
○ Schriftstil			
▪ fett	=	*wahr*	
▪ kursiv	=	*falsch*	
▪ unterstrichen	=	*falsch*	
○ Schriftfarbe	=	*rot*	intern auch Nummer: RGB = 255,0,0
○ Hintergrundfarbe	=	*keine*	RGB = 255,255,255 (weißer Hintergrund)
•			
→ Methode(n):			
• Neu erstellen			Übernahme von der Tastatur
• Schriftart setzen			
• Schriftstil ändern			
• Schriftfarbe setzen			
• Hintergrundfarbe setzen			
•			

Objekt:	Bemerkungen / Hinweise
Zelle	
→ Unterobjekt(e):	
• Formel	
•	
→ Attribut(e):	
• Zell-Adresse	
• Anzeige	sichtbare Belegung
• Zell-Belegung	Wert, Text, Formel
• Schreibschutz	
• …	
→ Methode(n):	
•	
•	
•	
• …	

Objekt:	Bemerkungen / Hinweise
Zell-Bereich	
→ Unterobjekt(e):	
• Zelle	
•	
→ Attribut(e):	
• Start-Zelle (Adresse) (Startzelle)	
• End-Zelle (Adresse)	
• Schreibschutz	
• …	
→ Methode(n):	
•	
•	
•	
• …	

Aufgaben:

1. *Überlege Dir, wie man die Daten einer Tabellen-Kalkulation speichern könnte! Stelle mindesten zwei prinzipiell unterschiedliche Möglichkeiten vor!*
2. *Erstelle eine Tabellenkalkulations-Datei (Mappe) mit einer einzigen Tabelle! Trage in die Zelle A1 die Zahl 100 ein und speichere die Datei als A1_100 ab!*
3. *Lösche den Wert aus der Zelle A1 und trage ihn in eine weiter rechts / unten (vielleicht die allerletzte) Zelle ein! Speichere die Datei unter dem Namen ZZxx_100 ab!*
4. *Erstelle in der letzten Datei noch 2 weitere Tabellen-Blätter und trage dort ebenfalls in eine sehr weit rechts 7 unten liegende Zelle eine 100 ein! Speichere die Datei nun unter dem Namen 3xZZxx_100!*
5. *Verlasse das Tabellenkalkulations-Programm und öffne den Datei-Manager (Windows-Explorer, Arbeitsplatz, Computer, …)! Vergleiche die Datei-Größen der abgespeicherten Dateien! Welche Rückschlüsse auf das Speicher-Verfahren (Datei-Format) kannst Du ziehen?*

Dateien sind Objekt-Listen, es werden nur die Dinge gespeichert, wo Änderungen vorgenommen wurden (bezogen auf eine leere Tabelle!)

Diagramme erstellen

das Referenzprinzip beispielhaft erklären

die Eignung von Diagrammdarstellungen bewerten

Tabellen für Berechnungen mit mehreren Datenreihen planen, erstellen und bewerten

Formeln mit Grundrechenoperationen entwickeln, implementieren und prüfen

Formeln mit Funktionen entwickeln, implementieren und prüfen

F. Textverarbeitung für den Anwendungsalltag

ein Seriendokument erstellen

Attributwerte für Seitenausrichtung und –ränder bestimmen, darstellen und begründet ändern

Jahrgangsstufe 8

10. Sensorgesteuerte Anwendungen entwickeln

Daten verarbeiten

- *Sensorwerte abfragen und verarbeiten*

Die SuS identifizieren Sensoren in einem Informatiksystem und ermitteln den Wertebereich eines Sensors.

- *das Prinzip der Analog-Digital-Wandlung erläutern*

Software für ein Informatiksystem entwickeln

- *eine Spezifikation prüfen und erweitern*

Eine Spezifikation ist die exakte Beschreibung der gewünschten Funktionalität eines Informatiksystems.

- *eine Anwendung realisieren und dokumentieren*

Die Dokumentation kann als Kommentarfunktion in der Implementierung, als Prozessdokumentation oder als Bedienungsanleitung realisiert werden.

- *eine Anwendung systematisch testen*

Anhand der Testergebnisse sich Schlussfolgerungen für die Entwicklung der Anwendung zu ziehen.

Problem-Fragen für Selbstorganisiertes Lernen
Wie messen Computersystem Umweltdaten? Warum müssen Umweltdaten in digitale Daten für den Computer umgewandelt werden? Wie kann man Umwelt-Daten in digitale Daten umwandeln? Kann man auch digitale Daten wieder in analoge Daten umwandeln? Was unterscheidet Computer- und Informatik-Systeme? Wer bestimmt wie, was ein Datenverarbeitungfs-System leisten muss? Legt man die Anforderung an ein Informatik-System vorher fest oder nimmt ein fertiges System und beschreibt die zufällig entstandenden Funktionen für den Vertrieb? Sind digitale Meßgeräte immer fehlerfrei?

10.x. Umwelt – Computer – Umwelt

10.x. Internet of Thinks, Smarthome und kein Ende

Sensorwerte abfragen, verarbeiten und bewerten

das Prinzip der Digitalisierung erläutern

Tabellen für Berechnungen unter Verwendung absoluter und relativer Zellbezüge planen und erstellen

11. Daten automatisiert auswerten

Daten

- *Tabellen mit mehreren Datenreihen planen, erstellen und interpretieren*

Das automatisierte Erzeugen oder Fortsetzen von Datenreihen ist zu thematisieren.

- *bedingte Formatierungen zweckmäßig einsetzen*
- *Daten nach mehreren Kriterien sortieren*
- *Daten über mehrere Attribute filtern*

Formeln

- *absolute und relative Zellbezüge verwenden*

Das Kopieren von Formeln ist Anlass für Betrachtungen zu absoluten und relativen Zellbezügen.

- *Formeln mit arithmetrischen Funktionen interpretieren und entwickeln*

Einfache Funktionen sind min(), max(), anzahl(), mittelwert(), summe().

- *die WENN-Funktion zweckmäßig interpretieren und verwenden*
- *eine Verweis-Funktion interpretieren und verwenden*

das Prinzip eines Sortierverfahrens beschreiben

Das Prinzip eines Sortierverfahrens sollte anaktiv und modellhaft veranschaulicht werden.

Vorschlag zur inhaltlichen Vertiefung:
Die Planung einer Tabellenstruktur sollte auch ohne das Vorliegen konkreter Daten erfolgen.

Problem-Fragen für Selbstorganisiertes Lernen

(c,p) 2015 – 2020 lsp: dre

14.x. Sortieren im Alltag

das Prinzip eines Sortierverfahrens beschreiben

geschachtelte algorithmische Strukturen interpretieren und implementieren

14.x. Sortier-Algorithmen

interessante Parameter in der Informatik:
Speicherplatz-Bedarf
Sortier-Zeit / Zeit-Bedarf
Sicherheit, dass Ergebnis immer stimmt

out-of-place-Verfahren

ursprüngliche Daten werden nach und nach zerlegt und in einer neuen Struktur (neu) zusammengestellt

Vorteil(e):

- relativ einfach (meist recht primitive Algorithmen)
-

Nachteil(e):

- größerer Speicher-Platz-Bedarf
- benötigen (wegen der Einfachheit) meist mehr Arbeitszeit
- originale Daten gehen verloren (ev. muss also eine Kopie erstellt werden)
-

Beispiel: Select-Sort

man sucht aus der ursprünglichen Liste immer den kleinsten oder größten Wert heraus (je nach Sortier-Richtung) und hängt dieses an die Ergebnis-Liste an
dieses macht man solange, bis die Quell-Liste leer ist

interatives Verfahren (wiederholt immer wieder den gleichen Schritt, bis nichts mehr zu verarbeiten ist)

Vorteil(e):

-

Nachteil(e):

-

inplace-Sortierung

Sortierung erfolgt innerhalb der Daten-Struktur, Elemente werden neu angeordnet (verschoben, Plätze getauscht, Zeiger "verbogen")

Vorteil(e):

- praktisch kein zusätzlicher Speicher-Platz-Bedarf
- meist sehr schnell

Nachteil(e):

- meist deutlich komplexere Algorithmen
- schwerer zu verstehen und zu programmieren

Beispiel: Bubble-Sort

Verschieben des jeweils größten Elementes ans Ende, bis man keine Verschiebung mehr gemacht hat

arbeitet interativ

Vorteil(e):

- sehr einfach (billig)
-

Nachteil(e):

- braucht sehr lange, (vor allem, wenn die Daten vorher zufällig genau anders herum sortiert waren)
-

Beispiel: Quick-Sort

teilt die originale Liste in zwei Teile (eine linke und eine rechte Liste)
wählt Bereichs-Grenzen ???
wählt zufälliges / mittleres Element als Vergleichs-Element (Kriterium)
alle Arbeits-Elemente, die kleiner als das vergleichs-Element sind, werden der linken Liste zugeordnet, und die anderen zur rechten Liste
nun wird auf die linke Liste das gleiche Verfahren angewandt, dann auf die rechte Liste
dies macht man solange, bis nur noch 1 Element in der Teil-Liste ist

arbeitet rekursiv (ruft sich selbst bis zu einem Abbruch immer wieder selbst auf)

Vorteil(e):

-

Nachteil(e):

-

Vorteil(e):

-

Nachteil(e):

-

14.x. Implementieren von Sortier-Algorithmen

z.B. in Snap!

Top-down-Vorgehen

Bottom-up-Vorgehen

das EVA-Prinzip erläutern

map-???-over-Block
bekommt eine Liste und eine Operation als Argumente
wendet dann intern die Opration auf die Liste an und gibt die Ergebnis-Liste dann wieder zurück

(c,p) 2015 – 2020 lsp: dre

12. Suchmaschinen verstehen

Aufbau und Funktionsweise einer Suchmaschine beschreiben

Die SuS verstehen unter einer Suchmaschine ein spezielles Datenbanksystem, das

- *den Inhalt von Webseiten mit Hilfe eines Suchroboters automatisch erhebt,*
- *die gewonnenen Daten mit Hilfe eines Indexers strukturiert und in einer Datenbank speichert,*
- *mit Hilfe einer Suchmaske Anfragen entgegennimmt, automatisch aufbereitet und an die Datenbank als Abfrage weiterleitet,*
- *die Ergebnisse der Datenbankabfrage mit Hilfe eines Algorithmus nach Kriterien sortiert und ausgibt.*

den Einfluss von Faktoren auf das Ergebnis einer Suchanfrage untersuchen, beschreiben und bewerten

Mögliche Faktoren sind Formulierungsvarianten der Abfrage, die Berücksichtigung von Geo- und Metadaten sowie die Wahl der Suchmaschine.

verschiedene Herangehensweisen zum Bewerten eines Suchergebnisses nutzen

Möglichkeiten sind die Untersuchung der Authentizität und der Vertrauenswürdigkeit der Quelle sowie des Wahrheitsgehalts.

die Macht einer Suchmaschine einschätzen und persönliche Schlussfolgerungen ableiten

Dies sollte anhand aussagekräftiger Beispiele erfolgen.

Problem-Fragen für Selbstorganisiertes Lernen

Gibt es neben google noch andere Suchmaschinen?
Welche Suchmaschine ist die Beste?
Finden Suchmaschinen alles im Internet?
Warum war und ist google so erfolgreich? Waren sie als erste im Internet vertreten?
Wie geht das, dass google seine Leistungen kostenlos anbietet? Wie finanziert sich die Firma?

Wie entsteht die Reihenfolge der Ergebnisse einer Anfrage bei google und Co?

Sind Suchmaschinen neutral und gerecht? Haben Suchmaschinen eine Macht?

Kann ich mich auch selbst googeln?
Was passiert, wenn ich andere Personen googele? Wie gut ist das Bild zu einer Person im Internet?
Google und der Datenschutz – alles ok?

x.y.z. Suchen und Finden im Internet

Suchen und Finden (ohne Suchmaschinen)

Um eine Seite im Internet aufzurufen braucht man eigentlich die exakte Adresse und den exakten Namen der Seite.
Das wird schnell zu einem Problem. Zum Einen können wir uns so lange Angaben gar nicht so genau merken und dann ist da auch noch der böse Tipp-Teufel, der immer wieder kleine Fehler in unsere Eingaben einschmuggelt.
Die Pioniere des Internets haben dieses Problem schnell erkannt und einige Verabredungen gemacht, um es Nutzern möglichst einfach zu machen, an bestimmte Seiten zu kommen.
So wurde verabredet, dass es bei jeder Internet-Adresse, die frei nutzbar sein soll, eine bestimmte Einstiegs-Seite geben soll. Diese Seite wird auch als Homepage bezeichnet. Sie stellt sozusagen die Basis – den Ausgangspunkt – für die anderen Seiten des gesamten Web-Auftrittes dar. Deren Namen muss man nicht genau wissen, der Browser probiert die vorgegebenen Möglichkeiten einfach aus. Er setzt diese Namen einfach von sich aus automatisch hinter eine Internet-Adresse. Neben einigen Spezial-Namen sind es die folgenden Datei-Namen, die als erstes geprüft werden:

index.htm	index.html	index.php
start.htm	start.html	
home.htm	home.html	

Exkurs: Aufbau einer HTML-Datei

HTML-Dateien sind sogenannte "Hyper Text Markup Language"-Dateien. Das bedeutet übersetzt Hypertext-Markierungs-Sprache.
In einer HTML-Datei ist der originale Text in lesbarer Form enthalten. Die besondere Hervorhebung z.B. von fett geschrieben Text-Teilen oder Überschriften wird durch sogenannte Tag's erreicht. Tag's werden in spitze Klammern notiert. Die Browser werten diese Tags aus und stellen die Texte dann entsprechend dar. Die meisten Tag's bestehen aus einem einleiztenden Tag und einem beendenden. Beide besitzen den gleichen Innentext, nur dass der End-Tag noch einen einleitenden Schrägstrich dazu enthält
Steht in einem HTML-Text die folgende Sequenz:

```
…
Der Text wird ab hier <b> fett gedruckt: Hallo, hallo </b>.
…
```

dann bewirkt die eine fettgedruckte Ausgabe des in die Bold-Tag's eingeschlossenen Wörter. Es ergibt sich also die Ausgabe:

Der Text wird ab hier **fett gedruckt: Hallo, hallo**.

Im Folgenden zeigen wir eine sehr, sehr einfache HTML-Datei. Diese dient nur dazu, um das Prinzip zu verdeutlichen und die wesentlichen Bereiche zu besprechen. Rechts daneben sind einige kurze Erläuterungen. Die Datei kann aber mit einem Text-Editor erstellt werden. In einem beliebigen Browser wird sie dann – wie eine Webseite – angezeigt.

```
<html>                                  Start des HTML-Textes
<head>                                  Beginn des nicht-sichtbaren Kopf-Bereiches
<title> Startseite </title>             Titel für das Browser-Fenster
<meta content="Homepage" >              Themen-Aufzählung
</head>                                 Ende des Kopf-Bereiches

<body>                                  Beginn des anzeigbaren Inhaltes
<h2> meine Homepage </h2>               eine Überschrift (Gliederungsebene 2)

Hallo, willkommen!<br>                  normaler Text mit einem Zeilenumbruch
Dies ist die Seite von
<b> dein Name </b>                      fett gedruckter Text

</body>                                 Ende des Inhalts-Bereichs
</html>                                 Ende des HTML-Textes
```

Die Ausgabe der HTML-Datei ist sehr einfach gehalten. Für eine optisch aufwendigere Internetseite sind neben HTML auch noch andere Technologien notwendig.
Wer sich über – zumindestens über Teile – einer beliebten Internetseite informieren möchte, kann das praktisch in jedem Browser tun. I.A. gibt es beim Klick auf die Webseite und beim Aufruf des Kontext-Menüs einen Punkt, der den Quelltext anzeigen lässt.
Für moderne Internetseiten benutzt man spezialisierte Programme, die Text- und Layout-Vorgaben automatisch in HTML usw. umsetzen.

Suchen und Finden mit Suchmaschinen

Wenn man jemand nach einer Suchmaschine fragt, dann kann man heute sicher sein, dass der Name google fällt. Aber google ist nicht die einzige Suchmaschine. Sicher ist google die grösste und beliebteste Suchmaschine der Welt. Worin liegt der Erfolg von google?
Google ist übersichtlich, schnell und findet immer Ergebnisse. Da hat man schnell den Eindruck, dass man mit google das gesamte Internet durchforsten könnte und google auch alle Such-Ergebnisse auch exakt ausgibt.
Viele Menschen glauben, wenn man etwas nicht mit google findet, dann gibst es das auch nicht im Internet. Das ist aber ein gefährlicher Irrglauben. Google "kennt" nur rund 1 bis 5 % des Internets.

Aufgaben:

1.
2.
3.

Gibt es noch andere Suchmaschinen?

einige Suchmaschinen nutzen andere Suchmaschinen im Hintergrund, es erfolgt dann aber oft eine Nachbereitung der Daten (z.B. Filterung hinsichtlich Jugendschutz-Bestimmungen usw.)

Suchmaschine	Internet-Adresse	Ansicht (Beispiel)	Bemerkungen	
allgemein				
google	http://www.google.de http://www.google.com		einfach, übersichtlich, allgemein gute Ergebnisse Datenschutz problematisch / unklar im Allgemeinen die meisten Suchergebnisse	
bing				
yahoo				
exalead	http://www.exalead.com/search		Seiten- bzw. Dateityp-Vorschau und Gefahrenhinweis mit weiteren Clusterungs-Möglichkeiten Datenschutz wird eingehalten (Suchmaschine in der EU)	
für Kinder und Jugendliche				
blinde Kuh	http://www.blinde-kuh.de			
Ecosia	http://www.ecosia.de			
ixquick	http://www.ixquick.de		Meta-Suchmaschine speichert keine Suchergebnisse (sehr guter Datenschutz)	
speziell				

(c,p) 2015-2020 lern-soft-projekt: drews

Wie sind die Seiten einer Suchmaschine aufgebaut?

Woran erkennt man eine gute Suchmaschine?

wichtige Kriterien

- **Übersichtlichkeit** — Trennung von Such-Ergebnissen und Werbe-Einträgen
- **Seiten-Rang-Bewertung** — Welche Seite ist warum an einer vorderen Positionen?
-

unwichtige Kriterien

-
-
-

irreführende Kriterien

- **Zeit zum Erzeugen der Ergebnisse** — durch paralleles Arbeiten vieler Computer keine Aussagekraft
- **Anzahl der Ergebnisse** — nur bei Spezial-Suchen interessant
  mehr als di ersten 10 Seiten werden kaum vom Nutzer ausgewertet
-

Was finden Suchmaschinen und warum?

(c,p) 2015-2019 lern-soft-projekt: drews

Wie speichert eine Suchmaschine die Daten zu Internetseiten?

Es gab Zeiten, da musste man seine Internet-Präsenz (Web-Auftritt, Homepage) noch bei den Suchmaschinen anmelden Erst dann durchsuchte die Suchmaschinen die Internet-Präsenz nach verwertbaren Informationen. Zwischen Anmeldung und dem ersten Erscheinen in den Such-Ergebnisse verging schnell mal ein Tag.
Heute sind die Ergebnisse weitaus aktueller. Selbst hochaktuelle Ereignisse sind innerhalb von Sekunden auf google zu finden.

Die Internetseiten werden nach bestimmten Begriff durchsucht, bzw. alle gefundenen Begriffe in einer sogenannten Index-Tabellen gesammelt.
Wir zeigen dies hier mal am Beispiel einer kleinen Text-Datei.

Computer.TXT

```
Computer

Computer – auch Personalcomputer (PC) genannt, sind Informationsverarbeitungsmaschinen, die aus Hardware und Software bestehen.
```

Als erstes wird dabei die Datei – bzw. deren Name (und Quelle) in einen Datei-Index übernommen. Computer können besser mit Zahlen (- also den Index-Nummer -) umgehen, als mit langen Dateinamen.

Computer.TXT

```
Computer

Computer – auch Personalcomputer (PC) genannt, sind Informationsverarbeitungsmaschinen, die aus Hardware und Software bestehen.
```

Datei-Index	
Index-Nr.	Dateiname
1	Computer.TXT

Für ein Beispiel wollen wir nur Sachwörter (Substantive) indizieren. Das erste passende Wort wäre "Computer". Dieser Wörtchen kommt insgesamt dreimal vor. Die verschiedenen Vorsilben und Wortenden ignorieren wir hier mal. Deshalb wir auch der Wortteil "computer" in Personalcomputer gefunden.

Computer.TXT

```
Computer

Computer – auch Personalcomputer (PC) genannt, sind Informationsverarbeitungsmaschinen, die aus Hardware und Software bestehen.
```

Die Fundergebnisse werden dann in die Wort-Index-Tabelle übernommen

Wort-Index		
Begriff	Index	Anzahl Vorkommen
Computer	1	3

und mit dem nächsten passenden Wörtchen weitergemacht.
Am Ende könnte die folgende Tabelle entstanden sein.

Wort-Index		
Begriff	Index	Anzahl Vorkommen
Computer	1	3
Personalcomputer	1	1
PC	1	1
Informationsverarbeitungsmaschinen	1	1
Hardware	1	1
Software	1	1

Kommt nun ein weiterer Text dazu,

Personalcomputer.TXT

```
Personalcomputer

Personalcomputer (Abk. PC) sind Computer, die ursprünglich für Benutzung
durch einzelne Personen gedacht waren. Davor waren meist nur große Firmen-
computer verfügbar.
```

dann werden die Tabellen aktualisiert.

Datei-Index	
Index-Nr.	Dateiname
1	Computer.TXT
2	Personalcomputer.TXT

Wort-Index		
Begriff	Index	Anzahl Vorkommen
Computer	1	3
Personalcomputer	1	1
PC	1	1
Informationsverarbeitungsmaschinen	1	1
Hardware	1	1
Software	1	1
Personalcomputer	2	2
PC	2	1
Computer	2	4
Abk	2	1
Benutzung	2	1
Personen	2	1
Davor	2	1
Firmencomputer	2	1

Ab und zu werden die Index-Tabellen sortiert und auch wieder überprüft. Dadurch wird die Nutzung beschleunigt und Fehlzugriffe oder eine fehlende Aktualität verhindet.

Datei-Index	
Index-Nr.	Dateiname
1	Computer.TXT
2	Personalcomputer.TXT

Wort-Index		
Begriff	Index	Anzahl Vorkommen
Abk	2	1
Benutzung	2	1
Computer	1	3
Computer	2	4
Davor	2	1
Firmencomputer	2	1
Hardware	1	1
Informationsverarbeitungsmaschinen	1	1
PC	1	1
PC	2	1
Personalcomputer	1	1
Personalcomputer	2	2
Personen	2	1
Software	1	1

Aufgaben:

1. *Prüfe, ob die Index-Tabellen exakt sind, oder ob dem Algorithmus hier ein oder mehr Fehler unterlaufen sind!*
2. *Erweitere die Indextabellen unter Indizierung der folgenden Datei!*

Hardware.TXT

```
Hardware

Zur Hardware eines Computers zählt man alle seine materiellen (elektroni-
schen und mechanischen) Teile. Es handelt sich praktisch um das reine Ge-
rät mit eventuell notwendigen Zusatzgeräten.
```

3. *Überlege Dir, was sich verändern würde, wenn z.B. die Datei "Personalcomputer.TXT" gelöscht würde! Wie aufwendig wäre die Korrektur der Index-Tabellen?*

Obst.TXT

```
Obst

Fast jeder kann intuitiv sehr gut zwischen Obst und Gemüse unterscheiden.
Wenn man aber sagen soll, was genau Obst ist, dann wird es schon schwieri-
ger.
Einfach gesagt, gehören zum Obst alle roh genießbaren Früchte oder Samen
von Bäumen, Sträuchern oder Stauden. Wenn man den Begriff Obst mehr bota-
nisch betrachtet, dann sind es die aus den befruchteten Blüten gebildeten
Teile.
```

Gemüse.TXT

```
Gemüse

Während Obst aus befruchteten Blüten entstehen, werden alle anderen essba-
ren Teile Gemüse genannt. Je nach Pflanzenteil unterscheidet man z.B.
Blattgemüse oder Wurzelgemüse. Es werden aber auch andere Einteilungen be-
nutzt. Verwendet man die Erntezeit als Kriterium, dann unterscheidet man
Frühgemüse, Sommergemüse, Herbstgemüse und Wintergemüse.
```

Apfel.TXT

```
Apfel

Der Apfel ist in Deutschland das meist gegessene Obst. Er gehört zum Kern-
obst. Die braunen Kerne sind die eigentlichen Samen des Apfelbaums. Die
Botaniker zählen die Apfelbäume zur Pflanzenfamilie der Rosengewächse.
```

Aufgaben:

1. ***Übernimm die nachfolgenden Tabellen! Fülle nun die Tabellen so aus, als wärest Du ein Such-Roboter! Benutze für jeden Begriff eine extra Zeile. (Der Such-Roboter soll nur Substantive – also Worte, die mit einem Großbuchstaben beginnen – auswerten.)***
2. ***Beschreibe Dein Vorgehen beim Durchsuchen der Dateien und dem Eintragen der Daten in die Tabellen!***
3. ***Vergleicht Eure Tabellen!***
4. ***Wie oft kommt das Wörtchen "Obst" in den Texten vor?***
5. ***Welche Möglichkeiten gibt es, die Vorkommen eines Begriffes möglichst schnell und sicher in der Index-Tabelle zu finden?***

Index-Nr.	Dateiname

Begriff	Index	Anzahl Vorkommen

Die Index-Tabellen werden schnell sehr groß. Nicht umsonst hat google die größten Rechenzentren der Welt. Um die Vielzahl von Worten etwas einzuschränken werden verschiedene Verfahren und Tricks angewendet.
So verwendet man nicht alle möglichen oder gefundenen Wörter, sondern schränkt die Anzahl z.B. dadurch ein, dass Neben- und Bindewörter ausgeschlossen werden. Solche Stopp-Worte, wie z.B. "und", "dann", "ein" usw. usf. wird wohl niemand ersthaft im Internet suchen.
Desweiteren beschränkt man sich auf die Wortstämme. Dazu werden Präfixe (Vorsilben) und Postfixe (Endungen, Nachsilben, Deklinationen) aus den Wörtern entfernt.
In vielen Sprachen gibt es auch die Möglichkeit der unterschiedlichen Schreibung – vor allem in der Verwendung in Internet-Medien. Unterschiedliche Schreibweisen werden auf eine heruntergebrochen und z.B. auch immer nur die Einzahl (Singular) verwendet.

Aufgaben:

1. ***Erstelle aus der folgenden Wort-Liste eine möglichst kurze, aber immer noch nutzbare, Schlüsselwort-Liste! Notiere zu jedem Schlüsselwort die Anzahl des Auftretens in der Wort-Liste!***

Scheinobst, frische, Kernobst, Obstmarkt, Frischgemüse, ungesüsst, Obst, rot, Gemüse, Kartoffel, süsslich, Kartoffeln, Apfelkerne, frisch, Steinobst, rötlich, unverdaulich, Süsskartoffeln, Frischemarkt, hellrot, Sonnenschein

2. *2.*

Festlegen von Rang-Ordnungen

Für eine Suche nach relevanten Dateien müssen wir neben dem bloßen Vorhandensein eines Suchbegriffs auch irgendein Maß für die Passung haben. Schließlich sollen ja nacher in den Such-Eregebnissen die wirklich passenden Dateien zuerst erscheinen.
Eine erste Möglichkeit wäre es, eine Rangfolge über die Anzahl der Vorkommen des Schlüsselwortes aufzubauen. Dazu müsste man dann die passenden Index-Einträge heraussuchen und in eine sortierte Liste bringen. Die häufigen Vorkommen zuerst und die seltenen nach hinten.

Aufgaben:

1. ***Erstelle eine Vorkommens-Rangliste der Dateien nach dem Schlüsselwörtchen "Obst"!***
2. *2.*

Das alleinige Vorkommen eines Schlüsselwortes ist nicht immer ausreichend für effektives Suchen und Finden. Neben der reinen Häufigkeit könnte man z.B. auch das Vorkommen des Schlüsselwortes in Überschriften auswerten. Kommt das Schlüsselwort in einem reinen Text vor, dann bekommt die Datei nur einen Wertungs-Punkt. Beim Vorkommen in der Überschrift werden drei Wertungs-Punkte vergeben.
In den Dateien einer Webseite kann der Ersteller auch noch unsichtbare Listen mit von ihm festgelegten Schlüsselwörtern (Themen) festlegen. Stimmen nun das Schlüsselwort aus dem Suchindex und das aus der Datei überein, könnte es sieben Punkte geben. So bekommen Dateien mit stimmenden Themen und einer fachlich angepassten Struktur (Überschriften) eine höhere Rang-Wertung. Sie würden dann in einer nach diesem Prinzip arbeitenden

Suchmaschinen eher vorne in den Suchergebnissen stehen, als Dateien, wo das Wörtchen nur ein paarmal vorkommt.

Aufgaben:

1. *Unten sind die möglichen versteckten Themen-Begriffe von Text-Dateien angezeigt. Erarbeite Index-Tabellen, welche die Wortvorkommen nach dem gerade beschriebenen Verfahren bewerten!*
2. *Vergleiche für die nachfolgend genannten Schlüsselwörter die Positionen in abgeleiteten Suchergebnis-Listen!*

a)	Obst	b)	Apfelbaum	c)	Unterscheidung
d)	Rosengewächs	e)	Gemüse	f)	Wintergemüse

Obst.TXT

```
Themen: Obst, Gemüse, Unterscheidung

…
```

Gemüse.TXT

```
Themen: Gemüse, Obst, Unterscheidung, Blattgemüse, Wurzelgemüse, Frühgemü-
se, Sommergemüse, Herbstgemüse

…
```

Apfel.TXT

```
Themen: Apfel, Äpfel, Obst, Apfelbaum, Rosengewächs

…
```

Viele Suchmaschinen nutzen außer solchen sachlichen Informationen auch noch die Aufruf-Zahlen. Eine Seite, die häufiger aufgerufen wird, bekommt zusätzliche Punkte. Dadurch gelangen Seiten, die andere Personen als passend empfunden haben, höher in die Ergebnis-Liste und die weniger relevanten Seiten gelangen eben weiter nach hinten.

Datei-Index

Index-Nr.	Dateiname	Aufrufe (letzter Tag)	Aufrufe (letzter Monat)
1	Computer.TXT	4	25
2	Personalcomputer.TXT	1	124

Wort-Index

Begriff	Index	Anzahl Vorkommen
Abk	2	1
Benutzung	2	1
Computer	1	3
Computer	2	4
Davor	2	1
Firmencomputer	2	1
Hardware	1	1
Informationsverarbeitungsmaschinen	1	1
PC	1	1
PC	2	1
Personalcomputer	1	1

Personalcomputer	2	2
Personen	2	1
Software	1	1

Nehmen wir z.B. an, dass der Rang einer Seite / Datei nach einer bestimmten Formel berechnet wird. Diese soll z.B. so aussehen:

Rang = akt_Aufrufe * 0,5 + alte_Aufrufe * 0,1

Datei-Index				
Index-Nr.	Dateiname	Aufrufe (letzter Tag)	Aufrufe (letzter Monat)	**Rang**
1	Computer.TXT	4	25	**4,5**
2	Personalcomputer.TXT	1	124	**12,7**

Eine weitere Möglichkeit der Vor-Positionierung von Seiten ist der Verkauf von Positionen in der Liste. Häufig werden die verkauften Einträge vor den eigentlichen Such-Ergebnissen angezeigt. Irgendwo kleingeschrieben steht dann, dass es sich um Werbung handelt. Bei google heißen die kaufbaren Begriffe Adwords.
Die Kombination von vielen Bewertungs-Verfahren haben google zum großen Erfolg verholfen. Der sogenannte Page-Rang-Algorithmus und deren genaue Bewertungs-System sind nur teilweise bekannt. Das sollte man immer beachten, wenn man die Ergebnis-Listen von Suchmaschinen angezeigt bekommt.
Die Ergebnis-Listen können also manipuliert sein, man selbst kann es in keinem Fall überprüfen. Einige kleine Manipulationen sind schon öffentlich geworden. Meist handelte es sich um die Überbetonung des amerikanischen Standpunktes zu einem Sachverhalt.
Für eine faire Bewertung müsste google seine Daten-Bestände und Berechnungs-Methoden auch anderen Nutzern und Anwendungen öffnen. Das würde aber ihrem Geschäfts-Modell widersprechen und damit sicher auch weniger Umsatz und Gewinn produzieren.

Aufgaben:

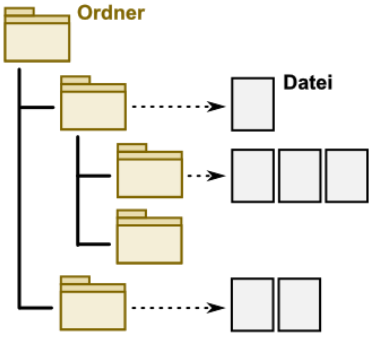
1. Übernehme die nachfolgende Tabelle in Deinen Hefter!
(Du kannst Dir weitere Hilfsspalten zur Tabelle hinzufügen!)
Berechne die Rang-Punkte für die Dateien mit Hilfe der Formel:

Rang = Aufrufe_gesamt * 0,3 + akt_Aufrufe * 0,7 + alte_Aufrufe * 0,1 + Hochstufung

Datei-Index						
Nr.	Dateiname	Aufrufe (letzter Monat)	Aufrufe (letzter Tag)	Hoch-stufung		Rang
1	Computer.TXT	35	4			
2	Personalcomputer.TXT	124	1			
3	Obst.TXT	74	3			
4	Gemüse.JPG	35	4			
5	Katze_Futter_lustig.MPG	30	38	15		
6	Gedicht.TXT	3	2			
7	geheim.TXT	18	72			
8	Werbung_Schokolade.TXT	7	38	100		
9	Hausaufgabe.PDF	1	12			
10	Hausaufgabe.ZIP	2	10			

2.

Durchsuchen von Ordner-Strukturen



Die Vielzahl der Dateien auf einem Rechner ist wegen der Übersichtlichkeit und besseren Zugreifbarkeit in einer hierrarchischen Baum-Struktur eingeordnet. In einem Ordner, der praktisch immer einen Zweig darstellt, können beliebig viele Dateien zusammengefsst werden. Die Dateien eines Odner sind in einer Liste zusammengefasst. Diese kann leicht von vorne nach hinten durchsucht werden.
Wird z.B. eine Datei mit einem bestimmten Namen gesucht, dann kann folgender Algorithmus zum Einsatz kommen:

suche_Datei:
1. Festlegen des Suchnamen's
2. **FÜR** jede Datei in der Datei-Liste **TUE**:
    2.1. **IST** Dateiname gleich dem Suchnamen **DANN**:
        2.1.1. Ausgeben "Datei gefunden"
        2.1.2. **STOPP**
3. Ausgeben "Datei nicht gefunden"

Auf die Nummerierungen wird i.A. verzichtet. Man braucht sie nur, wenn von einer Stelle zu einer anderen Stelle springen will. Der Arbeitsbefehl würde dann **GEHEZU** lauten. In manchen Algorithmen werden auch solche Arbeitsanweisungen wie **FESTLEGEN, EINGEBEN** und **AUSGABE** oder **AUSGEBEN** als Arbeitsbefehle betrachtet.

Wenn man die Anzahl der Dateien in der Datei-Liste kennt, dann kann man den Algorithmus auch etwas anders schreiben:

suche_Datei:
1. Festlegen des Suchnamen's
2. **VON** Dateinummer=1 **BIS** AnzahlDateien **TUE**:
    2.1. Nehme die Datei mit der Dateinummer
    2.2. **IST** Dateiname gleich dem Suchnamen **DANN**:
        2.2.1. Ausgeben "Datei gefunden"
        2.2.2. **STOPP**
3. Ausgeben "Datei nicht gefunden"

Aufgaben:

1. *Überlege Dir einen Algorithmus, der zählt, wieviele Dateien in einem Ordner enthalten sind!*
2. *Plane und teste einen Algorithmus, der das Vorkommen eines Schlüsselwortes (als ganzes Wort) in einer (Text-)Datei prüfen soll!*
3. *Wie könnte ein Algorithmus aussehen, der die Anzahl von Wort-Vorkommen (eines Schlüsselwortes) zählt?*

für Experten und zum Knobeln:

4. *Überlege Dir einen Algorithmus, der in einem Ordner in allen Dateien nach einem Schlüsselwort sucht und die Vorkommen zählt!*

Das Durchsuchen von Baum-Strukturen ist dagegen schon etwas komplizierter. Bei einer Liste kommt man irgendwann sicher zum Ende. Bei einer Bau-Struktur können die einzelnen Zweige aber beliebig oft weiter verzweigt sein. Das "weiss" der Such-Algorithmus auch vorher nicht. Er muss so flexibel sein, dass er mit Ordnern ohne Verzweigung – aber auch mit solchen mit beliebig tiefer Verzweigung – klar kommt. Unser nachfolgender Algorithmus soll alle Ordner im Baum durchlaufen. In einer ersten Nutzung könnte er sie ja mal alle zählen:

zähle_Ordner(Ordner):

1. **SOLANGE** ein (weiterer) Unterordner im Ordner vorhanden ist **TUE**:
    1.1. AnzahlOrdner +1
    1.2. zähle_Ordner(Unterordner)

HauptAlgorithmus:

1. Festlegen des StartOrdner's
2. Festlegen AnzahlOrdner=0
3. zähle_Ordner(StartOrdner)
4. Ausgeben AnzahlOrdner

Der hier gezeigte Algorithmus funktioniert nur in wenigen Programmiersprachen. Bei **zähle_Ordner** handelt sich mehr um eine Funktion – also einen Teil eines wirklich funktionierenden Programms. Die Funktion ruft sich selbst auf, um in einem Unterordner wieder zu prüfen, ob es dort weitere Unterordner gibt und diese zu zählen.

Solche Algorithmen sind schon recht verwirrend. Uns geht es hier auch nicht um das genaue Verstehen des Algorithmus, sondern einfach darum, zu erkennen, dass ein Computer genau so etwas (stupides) perfekt lösen kann.

Suchmaschinen nutzen nun z.B. solche – wie die oben gezeigten – Algorithmen, um eine Homepage nach Schlüsselbegriffen zu durchsuchen. Diese werden in die großen Index-Dateien aufgenommen. Ab sofort kann man die Seiten der Homepage auch über die Schlüsselbegriffe finden.

Experten-Suche in Suchmaschinen

Suchworte ausschließen: **Auto NOT Trabant** **Auto -Trabant**

Suchworte verbinden: **Bluse AND beige** **Bluse +beige**

mindestens ein Suchwort: **Rose OR Tulpe**

Entweder oder / Exklusives Oder (XOR):

Synonyme suchen: ***~Synonym*** (**~** über [Alt Gr] + [+])

Platzhalter- / Joker-Suche: **Lebensmittel***
(Stern (auch: Asterisk) steht für beliebig viele Buchstaben/Zeichen)

Begrenzung auf bestimmte Webseiten: **Tomatensuppe site:suppen.de**

Begrenzung auf ähnlichen Webseiten: **Karottenhose related:shop7.de**

Suche in der Seiten-Beschreibung (Megadaten der Webseite): **intitle:Impressum**

Suche mit Angabe von Zeiten / Zeiträumen: **Unwort 2000..·2015**
(hinter den Bereichs-Punkten muss ein Leerzeichen folgen!)

exakte Übereinstimmung / Zitat-Suche ("Lehrers-Liebling"): **"geklaut bei"**

Aufgaben:

1. *Ermittle die Anzahl gefundener Seiten, wenn man nach den Begriffen "Musik", "MP3", "Hip" und "hop" gemeinsam sucht!*
2. *Prüfe nun, wieviele Seiten zwar die Begriffe von 1. enthalten, aber nicht "Klassik"!*
3. *Finde alle Olympiasieger oder die Olympiasiegerinnen im Zehnkampf zwischen 1998 und 2003! Probiere auch mal eine andere Reihenfolge der Begriffe und Verknüpfungs-Zeichen bzw. –Wörter!*

Einige Suchmaschinen bieten auf der Start-Seite auch einen Link zu einer Spezial-Suche an. Dort findet man dann gut beschriebene Formulare zum Zusammenstellen einer detaillierten Suche.

5.x.y. Erstellen von HTML-Texten

Das Erstellen von HTML-Texten ist heute sicher nicht mehr die Wahl beim Erstellen von offiziellen Webseiten. Aber HTML eignet sich hervorragend, um das Prinzip von Formatierungs-Sprachen zu verstehen. Praktisch profesionelle Anwendungen sind dann z.B. DocBook oder Latex. Wenn man einmal das Prinzip hinter Formatierungs-Sprachen verstanden hat, dann ist es nur ein kleiner Schritt, sich in eine konkrete Anwendung einzuarbeiten.

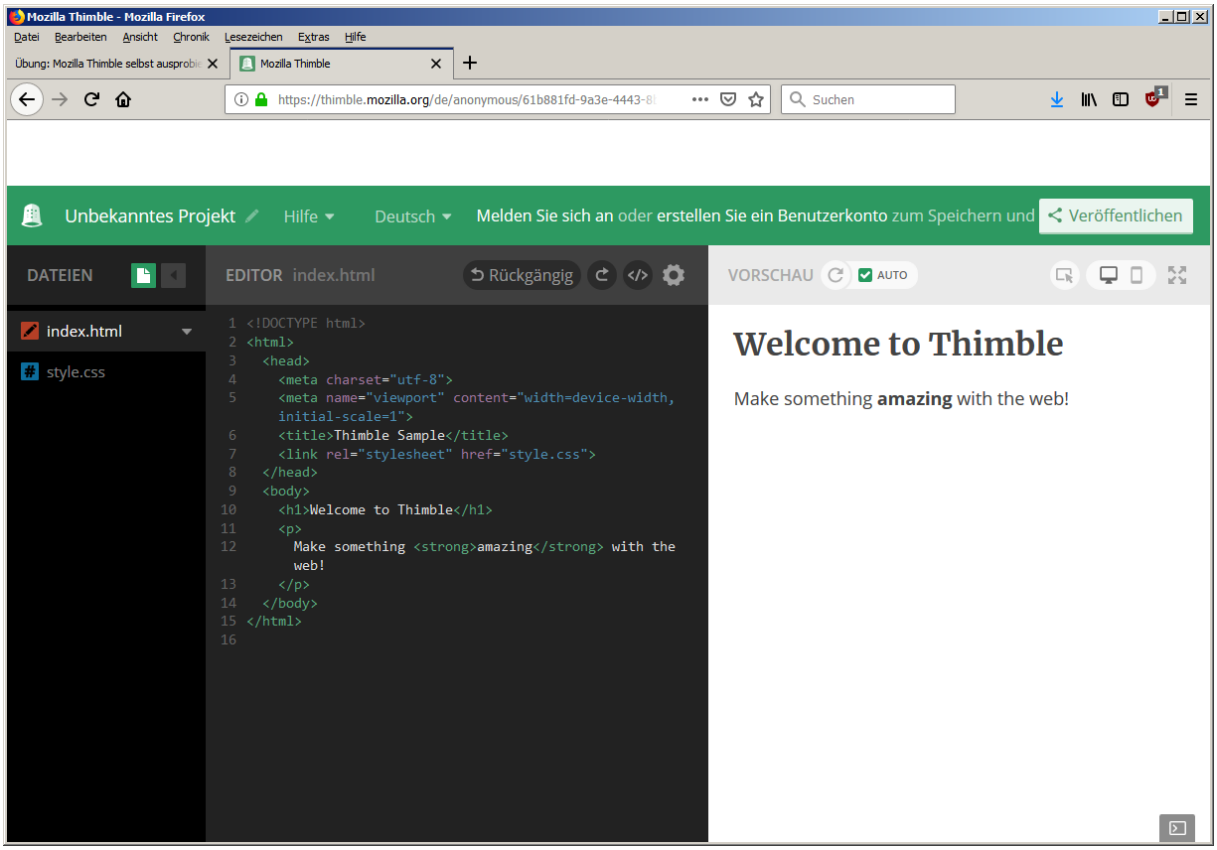
5.x.y.z. mozilla Thimble

→ https://github.com/mozilla/thimble.mozilla.org

keine Registrierung notwendig
Auto-Vervollständigung (beim Eingeben des Start-Tag's wird automatisch der End-Tag generiert

direkte Gegenüberstellung von Source-Code (Editor-) und Web-Ansicht
nachteilig ist die automatische Nutzung von CSS-basierten Formatierungen, was zu Anfang die Wirkung von Tag's etwas verkompliziert
aber in der vorgegebenen Form auch wieder gut für die Einführung von CSS geeignet ist

i.A. sollte mit einer einfachen, reinen HTML-Datei begonnen werden



automatische Neuanzeige kann durch manuelle (quasi auf Knopf-Druck) ersetzt werden, was z.B. bei komplexeren Konstrukten Sinn macht
es gibt unterschiedliche Anzeige-Rahmen (PC / Desktop; Smartphone / Tablet; Vollbild)

für die Veröffentlichung benötigt man einen Account
auch, wenn man Material für die Weiterbearbeitung bereitstellen möchte, ist ein Account notwendig

das Herunterladen der fertigen Text-Dateien ist (auch ohne Account) möglich

neben HTML-Dateien können auch CSS-, JS- und PHP–Dateien bearbeitet werden

Links:

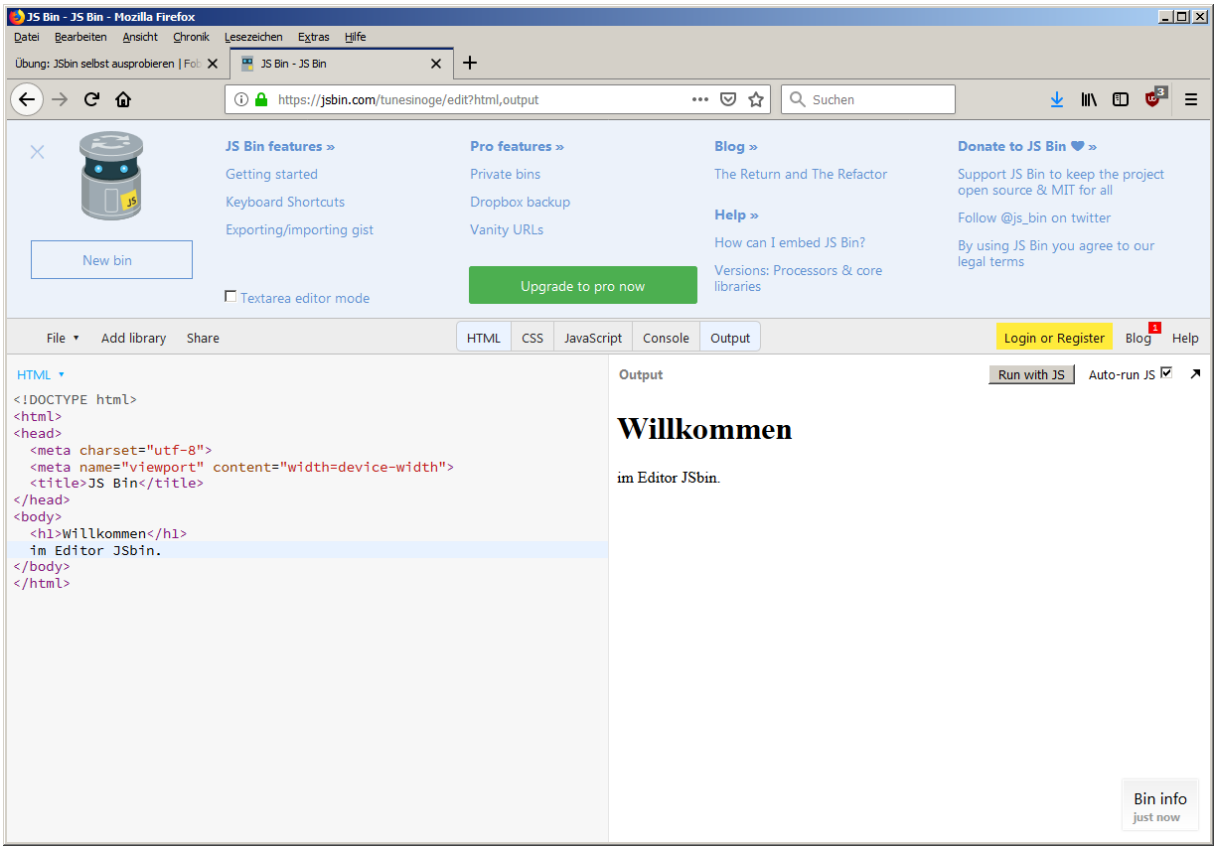
5.x.y.z. JSbin

keine Registrierung notwendig; für das Speichern benötigt man aber einen Account
keine Auto-Vervollständigung
Code muss vollständig eingetippt werden

auch geteilte Ansicht für Code und Ansicht

→ https://jsbin.com/?html,css,output (HTML- und CSS-Editor + Sofort-Anzeige)
→ https://jsbin.com/?html,output ((nur) HTML-Editor mit Sofort-Anzeige)
→ https://jsbin.com/?html (nur HTML-Editor; keine Sofort-Anzeige)

fehlende Bereiche können aber jederzeit zusätzlich eingeblendet / aufgerufen werden
auch das Ausblenden über das Reiter-Menü ist möglich



sofortige Anzeige der eingebenen Code-Abschnitte

CSS muss vollständig hinzugefügt werden

Links:
https://github.com/jsbin/jsbin (diverse Materialien / Quellcode des Editor's)

Links: (HTML-Erstellung und (Text-)Editoren)
https://www.w3schools.com/html/default.asp (verschiedene Editoren mit viel Hilfe und Zusatz-Material)

https://codepen.io/# (online-Web-Editor)
http://www.inf-schule.de/information/informationsdarstellunginternet (online-Tutorial)
https://wiki.selfhtml.org/wiki/HTML/Tutorials (Wiki zu HTML, CSS und JavaScript)
https://appcamps.de/unterrichtsmaterial/html-unterrichtsmaterial/ (Material zu HTML und CSS von AppCamps)

12.x. Internet-Suchmaschinen – gesellschaftliche Aspekte

Aufbau und Funktionsweise einer Suchmaschine beschreiben

den Einfluss von Parametern auf das Ergebnis einer Suchanfrage untersuchen, beschreiben und bewerten

verschiedene Strategien zum Prüfen des Ergebnisses einer Suchanfrage nutzen

den Einfluss der Anzeige der Suchergebnisse auf das eigene Verhalten reflektieren

den Einfluss der maschinellen Auswertung von Daten und Metadaten auf Individuum und Gesellschaft erkennen und bewerten

G. Netzwerke und kooperatives Arbeiten

Dateien unter Verwendung eines Ordnungssystems in vernetzten Systemen verwalten

Vernetzte Systeme zum kollaborativen Arbeiten nutzen

Informationssysteme problemgerecht auswählen und verwenden

H. kritischer Umgang mit Medien

Problem-Fragen für Selbstorganisiertes Lernen
Welche Medien gibt es? Wie kann man Medien einteilen? Wer nutzt welches Medium in welchem Umfang für welche Zwecke? Ist diese Kombination optimal?

kritisch-reflektiver Umgang mit Informationssystemen

Hinterfragen der herkunft und und der beabsichtigten Wirkung von Informationen und Daten

exakte und vollständige Angaben zu Quellen

H.x. Medien in unserer täglichen Welt

H.x. Medien in der Schule

Medien als Motivation

Medien als Einstieg in Problemstellungen

Medien zur Vermittlung von Inhalten

Medien als Mittel der Veranschaulichung

H.x. kritische Analyse von Medien

elementare und methodenbezogene Werkzeuge zur Bearbeitung von digitalen Bild- oder Tonmedien auswählen und benutzen

Information und digitale Darstellung unterscheiden

eine Technik zur Versionsverwaltung für Dateien verwenden

Strategien zur Datensicherung und –aufbewahrung beschreiben und vergleichen

Medienfälschungen erkennen

Fakten prüfen im Netz

(Informationen aus dem Internet einordnen und bewerten.)
nach Q: https://faktencheck.zlb.de (Zentral- und Landesbibliothek Berlin (in Zusammenarbeit mit Wikimedia Deutschland und dem eBildungslabor)), erweitert

Schritt / Tipp 1: Stopp

Wenn die Informationen / Sachverhalte Dich nicht betreffen / für Dich nicht wichtig sind, dann ist es das Beste, sich erst gar nicht damit auseinanderzusetzen! → Sage STOPP!

Schritt / Tipp 2: Quellen überprüfen

Wenn die Informationen / Sachverhalte Dich betreffen / für Dich wichtig sind, dann solltest Du die Informationen einordnen können! → Überprüfe die Quellen!

Schritt / Tipp 3: weitere Informationen / Quellen recherchieren

Wenn Du mehr eine bestimmte Aussage / Information wissen willst und Dir eine eigene Meinung bilden möchtest, dann hilft die Recherche nach weiteren Informationen, die das Thema möglichst umfassend von unterschiedlichen Seiten betrachten

Wird die gleiche Information einfach nur wiederholt? → Wahrscheinlich wurde die Information vor der Wiederholung / vor dem Liken nicht geprüft! → Prüfung verschärfen!

Wurde die grundlegende Information mit weiteren Informationen versehen, die ev. auch gegenteilige Positionen usw. wiederspiegeln, dann hat sich jemand mit dem Thema beschäftigt! → Argumente nachvollziehen und prüfen!
Wurde die grundlegende Information völlig anders dargestellt, dann ist die Prüfung beider Seiten notwendig! -- Prüfe alle Standpunkte!

Schritt / Tipp 4: Zurück zum Original

Da Nachrichten / Informationen aus dem Internet meist nur Ausschnitte / Teile von Situationen oder Gegebenheiten sind, werden diese auch nicht umfassend wiedergegeben. Es ist jetzt hilfreich nach der Original-Quelle / Erst-Information zu suchen! → Wo ist der Ursprung der Information?

Gibt es keine Quellen-Angaben oder sind diese nicht erreichbar / überprüfbar, dann ist eine gesunde Skepsis angebracht! → Glaube anonymen Quellen nur bedingt!
Gibt die ursprüngliche Quelle nur die gleichen Informationen wieder, dann handelt es sich vielleicht nur um eine Weiterleitung / Wiederholung (ohne Prüfung?) oder um ein Liken. → Weiterrechechieren?
Wenn die grundlegenden Informationen / Sachverhalte in der originallen Quelle anders dargestellt wurden, dann ist die Nachricht auf dem Weg zu Dir wahrscheinlich manipuliert worden! → Es könnte sich um Fakenews handeln!

Techniken der Bildmanipulation

• **Löschen und Einfügen von Bild-Elementen**	
• **Auswahl eines Bild-Teils / Beschneiden des Bilds Bild-Framing**	Entfernen oder Undeutlichmachen von Umgebungs- und / oder Situations-Informationen
• **Bild-Verwertung aus anderen (e.v. gegensätzlichen) Kontexten**	
• **Fotokombination**	mehrere Bilder werden übereinander gelegt und kombiniert
• **Bild-Beschriftung / Bild-Unterschrift**	
• **Verändern von Datum / Uhrzeit**	
• **strategische Wahl des Aufnahmepunktes**	von unten aufgenommen wirkt etwas größer / mächtiger von oben betrachtet wirkt etwas kleiner / unbedeutender
• **gestellte Aufnahmen / Inszenierungen** • **Ästhetisierung (bzw. aktive Subjekt-Inszenierung)**	

"Optimierung"

- **Honeyspotting** — vorrangig im "Social Media"-Bereich
  angenehme und interessante Bilder und Texte werden verbreitet und geteilt
  später werden die Bilder im Hintergrund ausgetauscht (es entsteht der Eindruck, dass das Zweit-Bild häufig geteilt wurde)
-

I. Textverarbeitung trifft Tabellenkalkulation

bedingte Funktionen zweckmäßig einsetzen

Tabellen in Textdokumenten erstellen und gestalten

Objekte in Textdokumenten beschriften

Formatvorlagen für Überschriften nutzen und modifizieren

Verzeichnisse automatisiert erstellen und aktualisieren

Seiten mit Kopf- und Fußzeilen gestalten

Dokumente in geeigneten Formaten austauschen

vernetzte Systeme zum kollaborativen Arbeiten nutzen

Jahrgangsstufe 9

13. Sensordaten erfassen und auswerten

Sensoren verwenden

- *Sensordaten analysieren und bewerten*

Die Analyse kann z.B. im Programm oder nachträglich in einer Tabellenkalkulation durchgeführt werden.

- *Zuverlässigkeiz von Sensordaten beurteilen*

Die SuS leiten Schlussfolgerungen zu Fehleranfälligkeit, Sicherheit und Konstruktionsprinzipien von Informatiksystemen ab.

- *Zusammenhang zwischen gemessener physikalischer Größe und Sensorwert erläutern*

Daten verarbeiten

- *Listen zur systemeatischen Speicherung und Verarbeitung von Sensorwerten verwenden*

Operationen zum Hinzufügen und Abfragen von Werten sind zu thematisieren.

Algorithmen strukturieren

- *Funktionen definieren und verwenden*

das EVAS-Prinzip erläutern

Die SuS erkennen die Allgemeingültigkeit des EVAS-Prinzips für Informatiksysteme. Anhand der verwendeten Hardware identifizieren sie Sensoren als Eingabegeräte, den Prozessor als Verarbeitungseinheit, Aktoren als Ausgabegeräte sowie den Speicher.

Problem-Fragen für Selbstorganisiertes Lernen

14. Publikationen und Präsentationen getalten

Inhalte in Dokumente einfügen

- *Objekte einfügen*

Je nach Kontext und Dokumenttyp fügen die SuS Formeln, Fußnoten, Querverweise, Quellen usw. ein. Sie unterscheiden eingefügte Objekte und Verweise und vermeiden Probleme, die sich aus dem Verweis auf externe Objekte ergeben.

- *Objekte beschriften*

Sie erstellen Tabellenüberschriften und Abbildungsunterschriften mit Hilfe der im Textverarbeitungssystem zur Verfügung stehenden Funktionalität.

- *Quellen exakt angeben*

Dazu gehören das Auflisten im Quellenverzeichnis und das automatische Referenzieren im Dokument.

das Prinzip der Trennung von Inhalt und Design anwenden

- *Dokumente mithilfe von Formatvorlagen gestalten*

In Präsentationssoftware kann der Zugang zu Formatvorlagen über Folienmaster erfolgen.
Dies schließ die Nutzung und Anpassung vorhandener sowie die Erzeugung neuer Formatvorlagen ein. Die SuS beachten Aspekte der Typografie.

- *Seiten mit Kopf- und Fußzeilen, Ausrichtungen und Rändern gestalten*

Bei der Gestaltung von Kopf- und Fußzeilen sollten automatische Seitennummerierungen sowie die unterschiedliche Formatierung der ersten Seite thematisiert werden.

- *Verzeichnisse automatisiert erstellen und aktualisieren*

Das Inhaltsverzeichnis wird anhand der verwendeten Formatvorlagen für Überschriften, das Abbildungs- und das Tabellenverzeichnis anhand der Beschriftungen generiert. Die SuS erkennen, dass die automatisch erstellenten Verzeichnisse Referenzen auf die jeweiligen Objekte erhalten.

Dokumente in geeigneten Formaten austauschen

Die SuS wählen begründet für den Austausch geeignete Dateiformate aus. Sie unterscheiden zwischen bearbeitbaren und nicht bearbeitbaren Formaten.

Problem-Fragen für Selbstorganisiertes Lernen

Was soll eine Trennung von Design und Inhalt?

Ist die Trennung von Design und Inhalt nicht unnötiger Mehraufwand?

15. Prinzipien der Datenübertragung verstehen

grundlegende Prinzipien der Datenübertragung im Internet beschreiben

Die Prinzipien sind anschaulich und enaktiv zu vermitteln

- *Prinzip der Adressierung*
- *Prinzip der Namensauflösung*
- *Prinzip der Zerlegung in Datenpakete*
- *Prinzip der Protokolle*
- *Prinzip des Routings*
- *Prinzip der asymmetrischen Verschlüsselung*

zur Sicherung der Vertraulichkeit, Authentizität und Integrität

Codierung verstehen

- *Prinzip der Codierung erläutern*

Ausgehend von der Übertragung einer Nachricht sollten ASCII und Unicode thematisiert werden. Die SuS stellen einen Transfer zu Codierungsverfahren aus verschiedenen Lebensbereichen her und leiten daraus allgemeine Merkmale der Codierung ab.

- *einfache Prüfsummenverfahren erläutern*
  - *für binäre Daten*
  - *für numerische Daten*

Die Erläuterung sollte anschaulich erfolgen und den Nutzen des Verfahrens anhand alltäglicher Anwendungsbeispiele aufzeigen.

die Geschwindigkeit von Datenübertragungen ermitteln und bewerten

Die SuS verwenden die Einheiten Bit und Byte für die Angabe der Datenmenge.

die Konsequenzen der Übermittlung von Daten und Metadaten einschätzen und Schlussfolgerungen ableiten

Die SuS werden sich ihrer Rolle als Lieferant von Daten bewusst. Sie erkennen das Potential und die Risiken der maschinellen Auswertung personenbezogener Daten.

Problem-Fragen für Selbstorganisiertes Lernen

15.x. Daten zuverlässig und korrekt übertragen

15.x. Wie funktionieren Netzwerke?

die Darstellung und Aufgaben von IP-Adressen und Domain-Namen beschreiben und die Aufgabe des Domain-Name-Systems erläutern

das Prinzip des Routings beschreiben

das Prinzip der paketvermittelten Datenübertragung erläutern

die Geschwindigkeit von Datenübertragungen ermitteln und bewerten

das Prinzip der Codierung von Daten erläutern

die Quelle einer Netz-Ressource exakt angeben und Quellenangaben prüfen

Kommunikationsprotokolle anhand eines Sequenzdiagramms darstellen und interpretieren

Darknet – die andere Seite des Internets

das Tor-Netzwerk

Pro:
relativ große Freiheit im Internet
vor staatlicher oder andersartiger Kontrolle / Beobachtung geschützte Kommunikation
Umgehen von Beschränkungen
sicherer Browser lässt sich von einem USB-Stick aus starten (portable App)
auch für Adroid (App heißt "Orfox") und iOS (hier heißt die App "Onion-Browser")

Contra:
verleitet ev. zum Rechtsbruch
bietet Kriminellen / Terroristen / … die Möglichkeit einer weitgehend unbeobachteten Kommunikation
ist deutlich langsamer, da der Weg der Daten stark verlängert wird und eine extra Verschlüselung und Entschlüsselung der Daten erfolgt
extra Programm (Tor-Browser) notwendig

Prinzip: Daten-Pakete werden über mehrere z.T. zufällig ausgewählte Server weitergeleitet und auch neu verschlüsselt

Neben den beiden Clients (Nutzer-Rechner) gibt es in einem Tor-Netzwerk mehrere spezielle Server. Der Eintritts-Knoten ist der erste Tor-Server. Er bekommt eine verschlüsselte Nachricht und kennt die IP-Adresse des Absenders. Das verschlüsselte Paket wird an einen weiteren Tor-Server weitergereicht. Dieser nimmt die verschlüsselten Daten entgegen und leitet sie an einen anderen Knoten weiter. Jeder Nutzer kann seinen Rechner auch als Tor-Knoten zur Verfügung stellen.
Er kennt nur die Absende-Adresse (IP) des vorherigen Knotens. Das kann sich mehrfach wiederholen. Der letzte Knoten (Austritts-Knoten) entschlüsselt die Daten und gibt die Anfrage an den zweiten Nutzer unter seiner IP weiter. Ab hier ist der Daten-Verkehr wieder beobachtbar / belauschbar. Die ursprüngliche Absender-Adresse ist nicht erkennbar. Der letzte Tor-Knoten gilt als der Urheber des Daten-Paketes.
Wenn der zweite Nutzer antwortet läuft der Vorgang umgekehrt zurück.
Optimalerweise befinden sich die Server / Knoten immer in verschiedenen Ländern. Da immer andere Gesetze gelten und die direkte Zusammenarbeit der staatlichen Organe der verschiedenen Länder fast nie funktioniert, ist ein Rückverfolgung sehr schwierig und aufwändig nur bei größeren Straftaten lohnt sich der Aufwand.
Der Staat kann aber durch fingierte Tor-Server an die Daten (Inhalte und Adressen) herankommen.
Kein Element (Knoten) des Tor-Netzwerkes hat alle notwendigen Informationen, um die Kommunikation zu kompromitieren.
Wenn man seinen Rechner als Tor-Server einrichtet ist auch ein anonymer – praktisch nicht rückvollziehbarer – Daten-Austausch möglich. Unter einer speziellen Domain (.onion) können nun Inhalte verfügbar gemacht werden. Die subdomain ist eine zufällige Zahl, eine nur Tor-intern nutzbare Adresse.

Aufgaben:

1. *Gebe zuerst www.google.com in Deinem Browser ein und beobachte auf welche Länderseite von google Du geleitet wurdest! Wähle Dich über den Tor-Browser in das Tor-Netzwerk ein! Probiere im Tor-Browser ebenfalls die Adresse www.google.com! Notiere Dir das Land, unter dessen "Identität" Du jetzt surfst! Melde Dich aus dem Tor-Netzwerk ab! Wiederhole den Test mehrfach (mindesten 5x)! Vergleiche Deine Beobachtungen mit denen anderer Kursteilnehmer!*
2. *2.*
3. *3.*

Links:
http://torproject.org (Download-Möglichkeit für Tor-Browser)
https://ricochet.im/ (abhörsichere Ende-zu_ende-KOmminikt)

Praktisch kann sich jeder einen eigenen Internet-Service programmieren / installieren. Die beiden End-Geräte (Kommunikations-Partner) benötigen ein spezielles / eigens Programm dass die Kommunikation ermöglicht. Das Internet selbst und die verschiedenen Server kön-

nen die transportierten Daten nicht erkannt werden. Dazu ist das spezielle Client-Programm notwendig.
Jeder Nutzer könnte sich so ein "eigenes", aber auch eingeschränktes "eMail-System" erschaffen. Die Daten werden innerhalb des Internets unabhängig vom anderen Netz (→ Netz-Neutralität) ganz normal weitergeleitet.

15.x. Datenübertragung mit Sicherheit

je ein einfaches Prüfsummenverfahren für binäre und für numerische Daten erläutern und dessen Nutzung begründen

das Prinzip der asymmetrischen Verschlüsselung beschreiben

(c,p) 2015 – 2019 lsp: dre

J. Medien analysieren und produzieren

J.x. Medien-Komplex: Internet

J.x. Medien-Komplex: Fernsehen

J.x. Medien-Komplex: Werbung

J.x. Medien-Komplex: Information und Nachrichten

J.x. Medien-Komplex: Kommunikation und Globalisierung

J.x. Medien-Komplex: Unterhaltung

J.x. Medien-Thema: Gewalt-Darstellung

J.x. Medien-Thema: Simulation von Medien

J.x. Nutzungs-Verhalten

die reine Medien-Nutzung nicht das wirkliche Problem

begleitet von immer kürzeren Ruhephase
suchtartiges Gefühl etwas zu verpassen / es zu spät zu erfahren
Offenlegung immer größerer Bereiche des Persönlichen – eigentlich sehr Privatem
Abschottung von der realen Umwelt; Flucht in die begrenzte Welt des Mediums

lt. PINTA-Studie (2011) rund 1% der 14- bis 64-Jährigen Internet-abhängig
somit in Deutschland rund 500'000 Personen
in der Gruppe der 14- bis 24-Jährigen größte Verbreitung

Internet-Abhängigkeit ist charakterisiert durch
zunehmende Fixierung auf die Internet-Nutzung (z.B. Chatten, Spielen, …)
bei fehlender Nutzungs-Möglichkeit trten psychische Entzugs-Erscheinungen (Unruhe, Gereiztheit, Kommunikations-Probleme, …) auf

Kontrolle über Zeit-Verbrauch / Nutzungs-Zeit geht verloren
soziale Beziehungen (in der Familie, in der Schule, bei der Arbeit, …) sind beeinträchtigt / gestört (bis hin zum Abbruch der Beziehungen; vorzeitiges Beenden der Schule; Arbeitslosigkeit, …)
Vernachlässigung anderer Interessen / Hobby's / …

Problem der klar definierten Diagnose / Erkennung

Suche nach einer Alters-gerechten Offline/online-Balance
Inhalt-liche Kontrolle
Angebot an Kommunikation über Medien / Vertrautmachen mit den Medien (Kinder ← → Eltern)
Suche nach geeigneten Behandlungs-Methoden
moderne / -gerechte Betreuung durch OASIS-Projekt (Online-Ambulanz-Service für Internet-Süchtige) Holt die Betroffenen dort ab wo sie sind (am Bildschirm)

J.x.1. Online-Spiel-Sucht

Sucht-fördendes Potential durch ständige Belohnungs- und Bestrafungs-Mechanismen (auch für die reine Anwesenheit im Spiel)

K.x. Textverarbeitung mit großen Dokumenten

Hier geht es jetzt weniger um die reine Textverarbeitung, sondern um solche Arbeitstechniken, die das Arbeiten mit größeren Dokumenten effektivieren. Dazu zählen wir die konsequente Nutzung von Format-Vorlagen oder z.B. automatische Idexe und Verzeichnisse.
In Dokumenten mit vielleicht 30 oder mehr Seiten möchte niemand Änderungen einer bestimmten Überschrift-Ebene mit der Hand vornehmen. Da ist die Gefahr einfach zu groß, dass eine Überschrift übersehen wird, oder man eine Formatierung nicht exakt reproduziert.
Änderungen an einem Kapitel auf z.B. Seite 3 können sich ev. auf das gesamte nachfolgende Dokument auswirken. Vielleicht muss dann auch noch jeder Verzeichnis oder ein Index aktualisiert werden. Da ist u.U. extrem viel Arbeit. Noch schlimmer wird es, wenn man damit fertig ist und dann merkt, dass auf Seite 6 auch noch eine Änderung erfolgen muss, die sich auch wieder auf das ganze nachfolgende Dokument auswirkt.
Alle modernen Textverarbeitungen bieten hier sehr gute automatische Systeme, die bei richtiger Anwendung die Beherrschung großer Dokumente zum Kinderspiel macht.

K.x.y. große Dokumente mit Libre Office WRITER

K.x.y. große Dokumente mit microsoft WORD

K.x.y. große Dokumente mit Libre Office TEXTMAKER (FREE)

Jahrgangsstufe 10

16. Sprachen und Sprachkonzepte

Eigenschaften natürlicher und künstlicher Sprachen analysieren und vergleichen

Die SuS beschreiben Eigenschaften von Sprachen unter Verwendung der Begriffe Zeichen, Zeichenfolge, Wort, Satz, Alphabet, Grammatik, Sprache, Syntax und Semantik. Sie untersuchen die Sprachen auf Mehrdeutigkeit, Fehlerfreundlichkeit, universalität und Flexibilität.

ein visuelles, akustisches oder haptisches Hilfssystem zur Darstellung natürlicher Sprachen untersuchen

Hilfssysteme dienen der Codierung von Sprachen für einen alternativen Übertragungsweg.

Sachverhalte und Zusammenhänge in einer logischen Programmiersprache darstellen und interpretieren

Geeignete Sachverhalte und Zu ergeben sich z.B. durch die betrachtung von Verwandtschaftsbeziehungen oder logischen Rätseln.

Problem-Fragen für Selbstorganisiertes Lernen

Können Computer Sprachen verstehen? Können sie alle Sprachen verstehen?
Was ist das besondere an Programmiersprachen?
Wie lassen sich sprachliche Einheiten zwischen Kommunikanten austauschen? Gibt es immer auch alternative Wege?

Gibt es Programmiersprachen mit denen man logische Rätsel, mathematische Beweise oder Verwandtschafts-Verhältnisse lösen kann?

16.x. natürliche und künstliche Sprachen

Eigenschaften natürlicher und künstlicher Sprachen analysieren und vergleichen

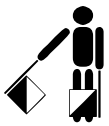
ein Hilfesystem zur Darstellung natürlicher Sprachen untersuchen

die Notwendigkeit der Formalisierung von Sprache als Voraussetzung für die Verarbeitung mit Informationssystemen begründen

den Aufbau und die grundlegende Funktionsweise eines Informationssystems nach dem von-Neumann-Modell beschreiben

die Aufgaben eines Compilers oder Interpreters beschreiben

Flaggen-Alphabet
Schriftart: Semaphore Pramuka



16.x. Programmier-Sprachen

17. Problemlösen durch Programmieren

Probleme spezifizieren

Die Probleme müssen über eine hinreichende Komplexität verfügen, um geschachtelte Strukuren, verknüpfte Bedingungen sowie Funktionen verwenden zu können.

Lösungsideen in einer programmiersprachen-unabhängigen Form interpretieren und darstellen

Mögliche Darstellungsformen sind Struktogramme oder Pseudocode.

die erforderlichen Algorithmenstrukturen und Daten interpretieren und schrittweise implementieren

Die SuS verwenden die Strukturen Sequenz, kopfgesteuerte Schleife, Verzweigung. Die Interpretation der Daten beinhaltet die Bestimmung des Datentyps. Es sind die Datentypen Zeichenkette, Wahrheitswert, Ganzzahl und Gleitkommazahl und deren typische Operationen zu thematisieren. Durch die verwendung von Variablen entwickeln die SuS allgemeine Lösungen für eine Klasse von Problemen.

die erforderlichen Elemente der Programmiersprache interpretieren und schrittweise implementieren

Auf die Spezifik der Programmiersprache ist nur im erforderlichen Maße einzugehen.
Für die Implementierung sind Hilfen zur Programmiersprache zur Verfügung zu stellen.

Funktionen verwenden, interpretieren und implementieren

Funktionen werden allgemein als Unterprogramme verstanden, bei denen Rückgabewerte und Parameter optional sind.

Lesbarkeit des Quelltextes sicherstellen

- *strukturierte Darstellung*
- *Verwendung zweckmäßiger Variablen- und Funktionsbezeichner*
- *aussagekräftige Kommentare*

vom Compiler bzw. Interpreter festgestellte Syntaxfehler interpretieren und korrigieren

Problemlösungen systematisch testen und Fehler im Programmablauf beseitigen

Problem-Fragen für Selbstorganisiertes Lernen

17.x. Projekt: Informationssystem

ein Vorgehensmodell für die Entwicklung eines Projekts kennen und anwenden

Projektauftrag und Meilensteine festlegen

Arbeitsprozesse gemeinsam planen und reflektieren

ein Produkt systematisch testen

ein Produkt mit Ergebnis und Verlauf präsentieren

eine Methode der Projektauswertung anwenden

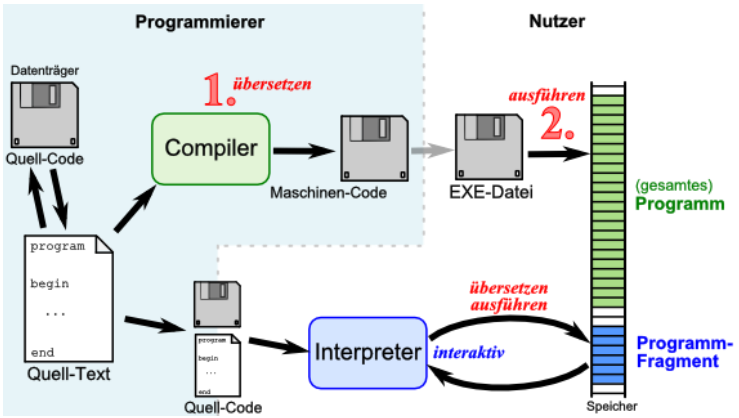
17.x. Übersetzungs-Konzepte für Programmier-Sprachen in Maschinen-Sprache

Die Mikroprozessoren, aber auch alle anderen technischen Strukturen in Computern, verstehen nur die Maschinen-Sprache. Diese besteht lediglich aus den Buchstaben 0 (Null) und 1 (Eins). Genaugenommen verstehen die Computer nur Spannung AN und AUS. Eine anliegende Spannung wir von uns Menschen als 1 (Eins) betrachtet und eine fehlende Spannung aben als 0 (Null).
Jedem wird sicher sofort klar, dass es uns Menschen sehr schwer fallen würde Computer nur mit 0 und 1 alle notwendigen Arbeiten, Berechnungen usw. usf. mitzuteilen. Wir benutzen lieber klare – für uns verständliche Befehle, wie: "addiere 3 und 4".
Diese Berechnung kann man noch ganz anders notieren: "3 + 4" oder "addiere(3; 4)" oder Summe(3; 4)". Man kann diese unterschiedlichen Formulierungen für die gleiche Aufgabe als unterschiedliche Programmiersprachen (Befehlssprachen) verstehen. Für uns Menschen wahrscheinlich gleich gut verständlich.
Da Computer die Befehle als Nullen und Einsen brauchen, müssen unsere verständlichen Anweisungen in die Computersprache übersetzt werden. Wir nennen diese Sprache Maschinensprache.
Da unsere Rechner 8-, 16-, 32- oder 64-bit-Prozessoren enthalten, werden die Befehle immer in diesen Portionen – also als Byte's, Word's oder Doubleword's vom Prozessor erwartet.

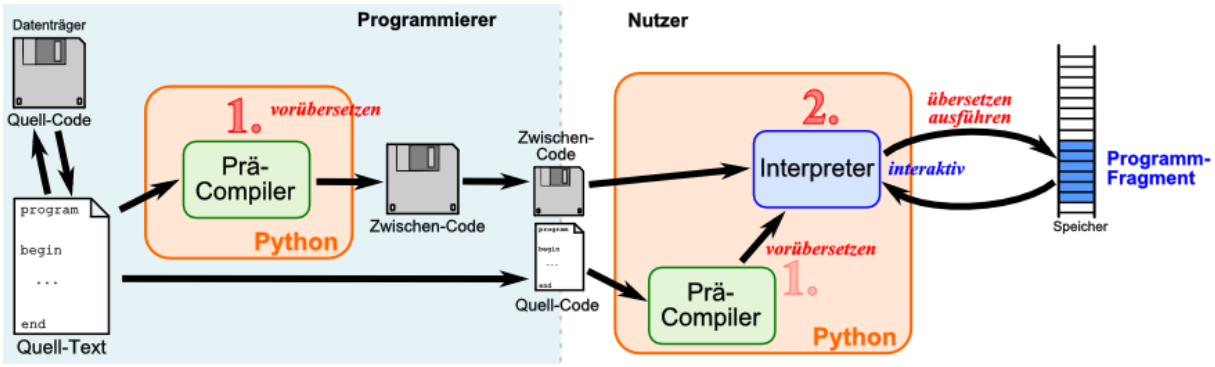
An irgendeiner Stelle muss die Übersetzung der Menschen-verständlichen Anweisungen in Maschinen-verständliche erfolgen. Früher hat ein Programmierer diese Übersetzung per Kopf gemacht. Er notierte die Sequenzen von Nullen und Einsen auf Lochbändern oder Lochkarten. Dazu waren sehr genaue Kenntnisse über den inneren Aufbau des Prozessors und des Computers notwendig.
Als erste Verbesserung wurd die Assembler-Programmierung erfunden. Statt Bit-Folgen erhielten die Maschinen-Befehle jetzt verständlichere Bezeichnungen. Trotzdem handelte es sich immer noch um ein fast direkte Maschinen-Programmierung. Die Programmierer blieben immer noch die unangefochtenen Guru's. Bald folgten Programmiersprachen, die mit verständlichen Befehlen arbeiteten. Ein "print 3 + 4" konnte dann schon fast jeder Computer-Nutzer programmieren. Solche Programme werden als Quell-Text bezeichnet. Der Nachteil

einer solchen Menschen-verständlichen Formulierung ist aber, dass die Anweisungen in die Maschinen-Sprache übertragen werden müssen.
Die eine Möglichkeit ist die vollständige Übersetzung des Quell-Textes in eine Programm-Datei. Programm-Dateien sind dabei solche Dateien, die ein Computer abarbeiten kann. Typische Dateiendungen sind .EXE, .COM oder .CMD (auf Mac-Rechnern). Das Übersetzungs-Programm für eine solche Umschreibung des Quell-Textes wird Compiler genannt. Diese Art der Programm-Übersetzung hat diverse Vorteile. Die fertigen Maschinen-Codes sind selbstständig lauffähig. Man kann sie als solches beliebig oft kopieren und auf beliebig vielen Rechnern benutzen. Die sogenannten Setup-Programme übernehmen genau diese Aufgabe, sie kopieren eine ausführbare Datei (und vieles mehr) auf den ausgewählten Computer. Compilierte Programme sind auch sehr schnell. Sie brauchen sich nur ihre Daten aus dem Speicher oder von Datenträgern holen und schon können sie die Leistungen erbringen, die Computer heute zu den vielbenutzten Werkzeugen machen. Da nur die ausführbaren Dateien weitergegeben werden müssen, bleibt der eigentliche Quell-Text das Geheim der Programm-Entwickler.
Aber die compilierten Programme haben auch Nachteile. Zum Einen lassen sich sich nur auf bestimmten Klassen von Computern nutzen. Ein für Windows compilertes Programm läuft nicht auf einem Mac oder unter Linux. Mac-Programme laufen wiederum nicht unter Windows und auch nicht auf einem Linux-Rechner. Für einen Mac oder ein Linux-System muss das Programm jeweils extra compiliert werden.
Besonders problematisch ist bei comilierten Programmen der Umgang mit internen Fehlern. Meist kam es beim Auftreten von Fehlern zu Total-Abstürzen. Das konnte soweit gehen, dass der ganze Rechner neu gestartet werden musste. Oft war dann auch die gesamte Arbeit (also die bearbeiteten Daten) weg. An welcher Stelle das Programm genau einen Fehler produziert hat, war nur aufwendig herauszubekommen.
Typische Compiler-Sprachen sind FORTRAN, PASCAL, C und Assembler.
Bei der zweiten Übersetzungs-Art geht man anders heran. Hier werden immer nur einzelne Quell-Text-Zeilen oder Anweisungen in Maschinen-Code übertragen. Dieser Code wird dann sofort ausgeführt und dann mit der nächsten Zeile / Anweisung weitergemacht. Man nennt solche Übersetzungs-Programme Interpreter. Durch die Schritt-für-Schritt-Übersetzung wird eine Fehler-Quelle schneller erkannt. Der Interpreter bleibt in der Zeile stehen und kann diese als Fehlerstelle anzeigen. Die Quell-Texte können jetzt auf jedem beliebigen Betriebssystem (Windows, Mac oder Linux) genutzt werden. Hier kommt aber einer der entscheidenden Nachteile zutage. Man benötigt für jedes Betriebssystem einen Interpreter. Und dieser Interpreter wird auch auf jedem Rechner gebraucht. Nur dann kann ja der Quell-Text wieder in Maschinen-Code übersetzt werden. Durch das schrittweise Übersetzen vergeht aber auch viel Zeit. Interpetierte Programme sind deshalb deutlich langsamer als compilierte.
Die Nutzer brauchen auch immer den Quell-Text. Das ist aber etwas, was Programmierer nicht unbedingt herausgeben wollen. Darin steckt ja ihre geistige Arbeit und die soll sich nicht jeder unbedingt immer frei kopieren dürfen.
JavaScript, PHP oder Ruby sind klassische Vertreter der Sprach-Familie von Interpreter-Sprachen.

Für viele Programmier-Sprachen existieren sowohl Compiler- und auch Interpreter-Versionen. Dazu gehört z.B. BASIC. Eigentlich ist BASIC eine ursprüngliche Interpreter-Sprache, aber es gibt eben auch Compiler.



Heute gibt es auch viele Programmier-Sprachen, die kombinierte Compiler-Interpreter-Übersetzungen verwenden. Bekannte Beispiele sind Python und JAVA. Bei ihnen wird der Quell-Text zuerst in einen Zwischen-Code comiliert. Dieser Zwischen-Code kann dann auf den verschiedenen Systemen für die Verwendung interpretiert werden.



17.x.y. Entwicklung einfacher Programme mittels Text-Anweisungen

Die Block-orientierte Programmierung ist uns schon mehrfach begegnet. Wir brauchten nur Bausteine nehmen und an geeigneten Kontaktstellen anbinden. Programme waren so schnell und intuitiv erstellt. Leider eignet sich dieses Konzept nur für kleinere Programme. Bei größeren Programmen würde es einfach zu unübersichtlich und bunt werden. Moderne oder größere werden eher über Text-Anweisungen erstellt.
Hinter den Blöcken von Programmier-Sprachen, wie Scratch, Snap! usw. stecken eigentlich auch Text-Anweisungen. In einigen Programmier-Systemen kann mach sich die Programm-Texte (Quell-Texte) auch ansehen.

Die folgenden Programmiersprachen sollen nur kurz und Beispiel-haft vorgestellt werden.
Für Python und Ruby gibt es in dieser Reihe passende Scripte, um sich intensiver mit der Sprache zu beschäftigen. Wir zeigen hier im Wesentlichen einige typische EVA-Programme.
Man sollte sich zum Ausprobieren eine Sprache auswählen. Das Prinzip ist bei allen Sprachen identisch.
Wir sehen hier auch sehr schön, dass gerade Interpreter-Sprachen für einen ersten Einstieg am Besten geeignet sind.

17.x.y.z. Programme mit Python entwickeln

Python ist mein persönlicher Favorit. Es ist klein und Leistungs-fähig. Von einfache Script-Programmen bis zu komplexen Software-Produkten kann man alles mit Python erstellen

17.x.y.z. Programme mit BASIC entwickeln

Früher war BASIC die Sprache, die als der Beste Einstieg in die Welt der Programmierung galt. BASIC ist sehr einfach und leicht verständlich. Leider ist in BASIC ein Befehl integriert, der viel Unheil stiften kann – der GOTO-Befehl. Er ermöglicht sogenannten Spagetti-Code, bei dem im Quell-Text wild hin- und her gesprungen werden kann. Selbst eingefleischte BASIC-Programmierer haben bei größeren Programmen da schnell die Übersicht verloren. Aber als Erste-Einstieg oder zum schnellen Programmieren eines einfachen Programm's ist BASIC immer noch gut nutzbar.

17.x.y.z. Programme mit JavaScript entwickeln

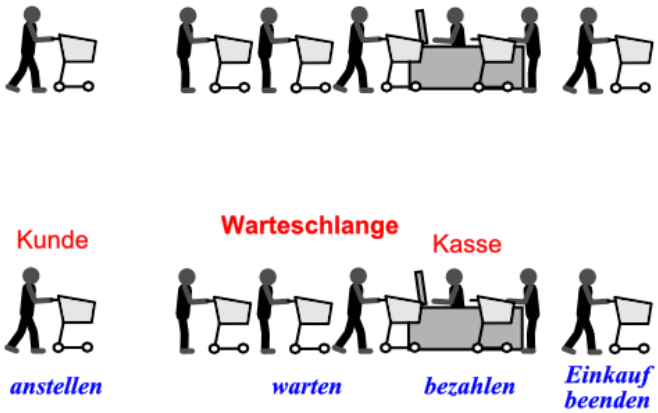
17.x.y.z. Programme mit Ruby entwickeln

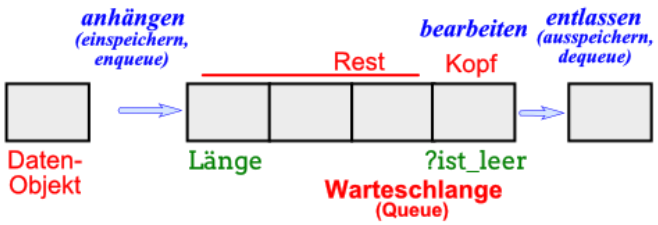
L. Daten-Strukturen

Anordnung und Hierrarchie von Daten-Objekten

die meisten Daten-Strukturen lassen sich durch andere Daten-Strukturen simulieren / darstellen
meist für uns Menschen schwerer zu durchschauen, weil nicht so assoziativ
oder für Computer aufwändiger zu bearbeiten
einige Programmiersprachen bieten oft nur einen reduzierten Umfang an Strukturen, meist solche, die sehr allgemein sind oder sich besonders einfach mit Algorithmen benutzen lassen

(Warte-)Schlange FIFO





Puffer

einfache Liste

komplexe Liste

Dictonary Wörterbücher

doppelt verkettete Liste

Ring-Puffer / Daten-Ring

Feld Array Vector

Keller Stapel Stack FILO

Baum Tree ???

Binär-Baum

ausgewogene Bäume

Netz Graph

18. Grundlagen der Digitalisierung

den Aufbau und die grundlegende Funktionsweise eines Informatiksystems nach dem von-Neumann-Modell beschreiben

Die SuS nutzen ein Werkzeug zur Simulation maschinennaher Programmierung mit Hilfe eines vereinfachten von-Neumann-Modells. Sie erkennen die Zweckmäßigkeit der verwendung höherer Programmiersprachen im Vergleich zur Maschinensprache.

Eignung binärer Signale für die maschinelle Verarbeitung erläutern

Die Prinzipien sind anschaulich mit einem Verweis für Möglichkeiten einer technischen Umsetzung zu vermitteln.

- *Bits logisch verknüpfen*
- *binäre Addition*

eine Grenze der binären Zahlendarstellung erklären

Die Grenzen können z.B. anhand der Wertebereiche ganzer Zahlen oder der eingeschränkten Abbildbarkeit von Dezimalbrüchen im Binärsystem verdeutlicht werden.

Auswirkungen der Digitalisierung erkennen

- *Chancen und Risiken der Nutzung von konkreten alltäglichen Informatiksystemen erkennen und bewerten*
- *aus den Folgen der Digitalisierung Rückschlüsse für das eigene Verhalten ziehen*
- *gesellschaftliche Folgen der Digitalisierung beurteilen und bewerten*

Die SuS diskutieren aktuelle Fragestellungen zur Digitalisierung.
Möglich Themenfelder sind

- *Ausfall- und Datensicherheit im privaten, beruflichen und wirtschaftlich-technischen Umfeld,*
- *digitale Identität und digitaler Nachlass,*
- *Nachhaltigkeit digitaler Daten,*
- *Grenzen komplexer Informatiksysteme.*

Problem-Fragen für Selbstorganisiertes Lernen

Gibt es allgemeine Aufbau-Prinzipien / -Strukturen / -Modelle von Datenverarbeitungssystemen?

Was ist das Geniale am VON-NEUMANN-Modell informations-verarbeitender Systeme?
Hat das VON-NEUMANN-System auch Nachteile? Kann man sie beheben? Gibt es Modelle ohne diese Probleme / Nachteile?

Zerstört die Digitalisierung Arbeitsplätze? Wird die Arbeitslosigkeit steigen?
Schaft die Digitalisierung neue Arbeitsplätze? Welche Qualifikationen werden für die Arbeitswelt der Digitalisierung gebraucht?

18.x. Digitalisierung in unserem Leben

Verfahren zur Sicherung von Vertraulichkeit. Authentizität und Integrität von Daten verwenden und beschreiben

Risiken der Nutzung von Informationssystemen erkennen und bewerten

aus den Folgen der Digitalisierung Rückschlüsse für das eigene Verhalten ziehen

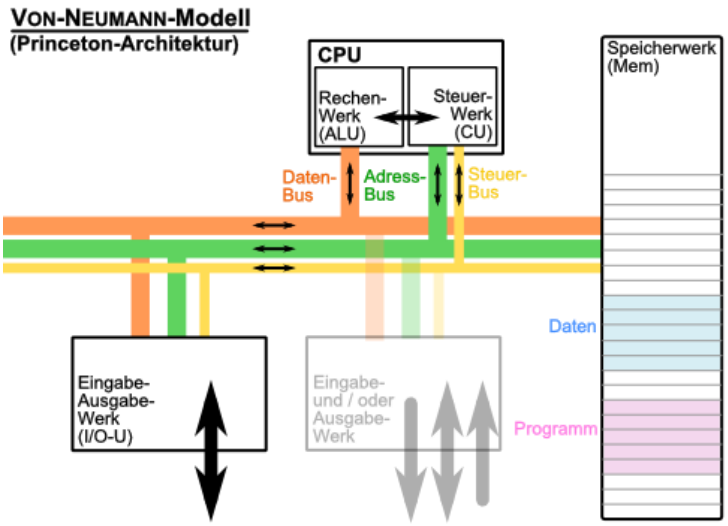
gesellschaftliche Folgen der Digitalisierung beurteilen und bewerten

Eine Brain-Untersuchung von mehr als 400 Unternehmen ergab, dass diejenigen, die frühzeitig die interne vernetzung vorantreiben, der analogen Konkurrenz in der Regel um Längen voraus sind. Der Studie zufolge ist die Wahrscheinlichkeit, zu den bestverdienenden Anbietern der Branche zu gehören, bei digitalisierten Firmen doppelt so hoch wie bei analog arbeitenden Konkurrenten. Die Chance, wichtige Entscheidungen schneller zu treffen als der Mitbewerber, ist sogar fünfmal größer. Und die Wahrscheinlichkeit, dass eine Entscheidung auch zum gewünschten Ergebnis – also zu einem Wettbewerbsvorteil – führt, ist dreimal so hoch wie bei Rivalen, die noch nicht auf Industrie 4.0 umgestellt haben.
Q: IN: manager magazin; Heft 12/2016; S. 104

18.x. Aufbau eines VON-NEUMANN-Rechner's

VON NEUMANN entwickelte diese Architektur an der Princton Univercity. Besonders, wenn man nach einer Alternative zum VON-NEUMANN-Rechner sucht, dann wird die Örtlichkeit wichtig. Die etwas andere Harvard-Architektur wurde an der Harvard Univercity entwickelt.

Praktisch hängen alle Werke, wie Lampion's am zentralen Bus.



Allgemeiner Ablauf in einem VON-NEUMANN-Rechner

System-Ablauf:

1. Programm (Befehle) befinden sich im Speicher
2. der Befehlzähler zeigt auf nächsten Befehl
3. Befehl (auf den (/ dessen Adresse) der Befehlszeiger zeigt) wird gelesen
   a) ev. werden Daten eingelesen (Speicher → Register)
   b) Befehl wird ausgeführt (Verknüpfung von Daten, Fällen von Entscheidungen, Berechnungen von Sprüngen), ev. werden Daten (z.B. Befehlsergebnisse) zurückgespeichert (Register → Speicher)
   c) Befehlszähler verändert (normal um eins erhöht; bei Sprungbefehlen auch in größeren Schritten)
4. weiter bei 3.

Befehls-Zyklus eines Mikroprozessor's

Schritt	Bezeichnung	Inhalt / Arbeits-Leistung
1	**FETCH**	Befehls-Aufruf nächster Befehl wird aus der Speicher-Adresse geladen, die in Befehls-Register (Befehls-Zähler) steht
2	**DECODE**	Dekodierung das Steuerwerk schaltet die zum Befehl gehörenden Logik-Schaltungen ein (bzw. gibt Daten-Leitung dahin frei)
3	**FETCH OPERANDS**	Operanden-Aufruf aus dem Speicher oder anderen Registern werden zusätzliche Operanden (Daten) geladen (auf die Logik-Schaltungen geleitet)
4	**EXECUTE**	Befehls-Ausführung Rechen-Logik arbeitet; Befehlszähler wird erhöht bzw. bei Schleifen / Sprüngen auf anderen Wert

		(neue Befehls-Adresse) gesetzt
5	**WRITE BACK**	Zurückschreiben (des Ergebnisses) ev. werden die Ergebnisse aus der Logik-Schaltung in Register oder Speicherzellen geschrieben

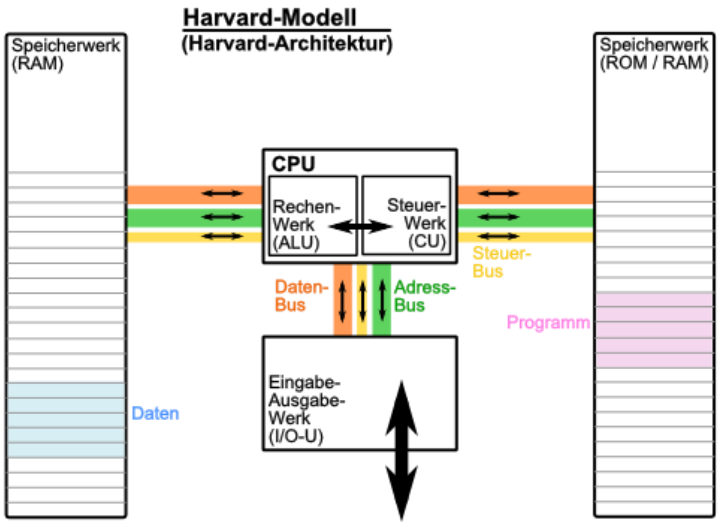
beobachtet man die Aktivitäten der CPU genauer, dann können die folgenden feiner gegliederten Abfolgen beobachtet werden

Daten (Anweisung) in Befehlsspeicher einlesen	1. Adresse auf dem Adressbus einstellen 2. Eingabeleitung auf Steuerbus aktivieren 3. Daten vom Datenbus (in Befehlsspeicher) übernehmen
Anweisung ausführen	1. Befehlslogik einstellen (Anweisung decodieren) 2. Befehlslogik aktivieren 3. Befehls(adress)zähler um 1 erhöhen
Daten eingeben	1. Eingabeadresse auf dem Adressbus einstellen 2. Eingabeleitung auf Steuerbus aktivieren 3. Daten vom Datenbus übernehmen
Daten in Speicher schreiben	1. Adresse auf dem Adressbus einstellen 2. Daten auf den Datenbus legen 3. Schreibleitung auf Steuerbus aktivieren
Daten in Register einlesen	1. Adresse auf dem Adressbus einstellen 2. Eingabeleitung auf Steuerbus aktivieren 3. Daten vom Datenbus (in Register) übernehmen
Registern verändern	1. Rechenlogik aktivieren 2. Ergebnis in Register speichern 3. Befehls(adress)zähler erhöhen
2 Registern verknüpfen	1. Rechenlogik aktivieren 2. Ergebnis in ein Register speichern 3. Befehls(adress)zähler erhöhen
Registern auswerten	1. Rechenlogik aktivieren 2. je nach Ergebnis den Befehls(adress)zähler verändern
Befehlszähler verändern	1. Befehlszähler einstellen
Daten aus Register speichern	1. Adresse auf Adressbus einstellen 2. Daten auf den Datenbus legen 3. Schreibleitung auf Steuerbus aktivieren
Daten aus Speicher lesen	1. Adresse auf Adressbus einstellen 2. Leseleitung auf Steuerbus aktivieren 3. Daten vom Datenbus übernehmen
Daten ausgeben	1. Adresse auf Adressbus einstellen 2. Daten auf den Datenbus legen 3. Schreibleitung auf Steuerbus aktivieren

18.x.y. Alternativ-Konzept: die Harvard-Architektur

Das VON-NEUMANN-Konzept ist aber nicht die einzige Lösung. An der Harvard Univercity wurde eine andere System-Architektur entwickelt und in Zusammenarbeit mit IBM im legänderen Rechner Mark I umgesetzt.
Diese enthält die gleichen Grund-Elemente Rechenwerk (CPU), Ein- und Ausgabe-Werke sowie Speicher-Werke. Aber genau beim letzteren liegt der wesentliche Unterschied. Es gibt nämlich zwei unabhängige Speicher-Werke, die über separate Bus-Systeme mit der CPU verbunden sind. Das eine Speicherwerk enthält das Programm und kann als RAM oder ROM ausgeführt sein. Das zweite Speicher-Werk, was die Daten enthält, ist grundsätzlich RAM.
Auch das Eingabe-Ausgabe-Werk ist über einen extra Bus an die CPU gekoppelt.
Alle drei Buss können mit unterschiedlichen Takt-Frequenzen (also langsamer oder schneller) arbeiten.

Insgesamt ergibt sich eine Sternförmige System-Architektur mit dem Rechenwerk im Zentrum.



Die Harvard-Architektur hat sich vor allem in kritischen Bereichen behauptet. Viele Systeme in der Raumfahrt orientieren sich an diesem System-Konzept.
Moderne Signal-Prozessoren beinhalten bis zu 8 CPU's, die auf gemeinsame Speicher zugreifen können. In der obigen Abbildung würden diese mittig zwischen den Speichern übereinander liegen.
In der Praxis kommen aber auch immer mehr Symbiosen aus beiden Konzepten zum Einsatz. Diese Verknüpfungen leben von den jeweiligen Vorteilen.
Besonders um die sogenannten RISC-Prozessoren sammelt sich die Praxis der Harvard-Rechner-Systeme.
Selbst innerhalb von VON-NEUMANN-Rechner finden sie ihren Platz in Leistungs-stärkeren Graphikkarten. Die darauf enthaltenen GPU's (Graphics Processing Unit) sind vorrangig nach der Harvard-Struktur aufgebaut.

Exkurs: der Rechner Mark I von IBM

elektromechanisch
1944
Harvard-Konzept für die Anordnung der Teilwerke

Aufgaben:

1. *Ermitteln Sie die Bezeichnung Ihrer Graphikkarte und die verfügbare Graphik- bzw. Video-Speichergröße!*
2. *Recherchieren Sie die Struktur und wichtige Merkmale der Graphikkarte! Ist Sie Harvard-orientiert?*
3. *Erklären Sie, warum auf einer – besonders für Spiele-Anwendungen geeigneten – Graphikkarte so große Speicher verbaut sind!*
4. *Lisa möchte auf ihrem Rechner (Desktop-Gehäuse mit Power- und Reset-Knopf, 7x USB) mit Farben arbeiten, die jeweils für Rot, Grün und Blau 256 Abstufungen zulassen. Der Desktop ist auf eine Auflösung von 1920 x 1080 Pixel (HD-Auflösung) eingestellt. Die Tastatur hat 102 Tasten und sie verwendet eine optische 3-Tasten-Scroll-Maus. Welches Farbsystem wird auf dem Rechner verwendet? Wie groß muss der Graphik-Speicher für Lisa's Rechner minimal sein.*

x.

für die gehobene Anspruchsebene:

x. Welche Graphik-Speichergröße müsste Lisa's Rechner mindestens haben, wenn Sie sich einen hypermodernen 4K-Bildschirm an ihre HD-Graphik mit je 32 bit pro Farbe anschliesst?

Vergleich und Gegenüberstellung von VON-NEUMANN- und Harvard-Architektur

	VON-NEUMANN-Architektur	Harvard-Architektur
Gemeinsamkeiten	• Bauelemente: Busse, CPU, Speicher, I/O-Systeme •	
Unterschiede	• gemeinsamer Haupt-Bus (geteilt in Daten-, Adress- und Steuer-Bus) • zentraler Bus • periphere CPU	• getrennte Busse für Daten und Befehle • Busse peripher angelegt • zentrale CPU
Vorteile	• nur ein Bus, der universell genutzt werden kann • Speicher ist flexibel zwischen Daten und Programmen aufteilbar	• Daten und Befehle können gleichzeitig, aber auch unabhängig voneinander geladen werden • da Trennung von Daten und Programm in untterschiedlichen Speichern, kann fehlerhafter od. manipulierter Code keine Programme überschreiben (nur Daten)
Nachteile	• Bus mit seiner Geschwindigkeit ist der Flaschenhals des Systems • Daten und Programme nicht unterscheibar	• freier Speicher lässt sich nicht für die jeweils anderen Daten bzw. Programme nutzen

18.x. Simulation eines VON-NEUMANN-Rechner mit MOPS

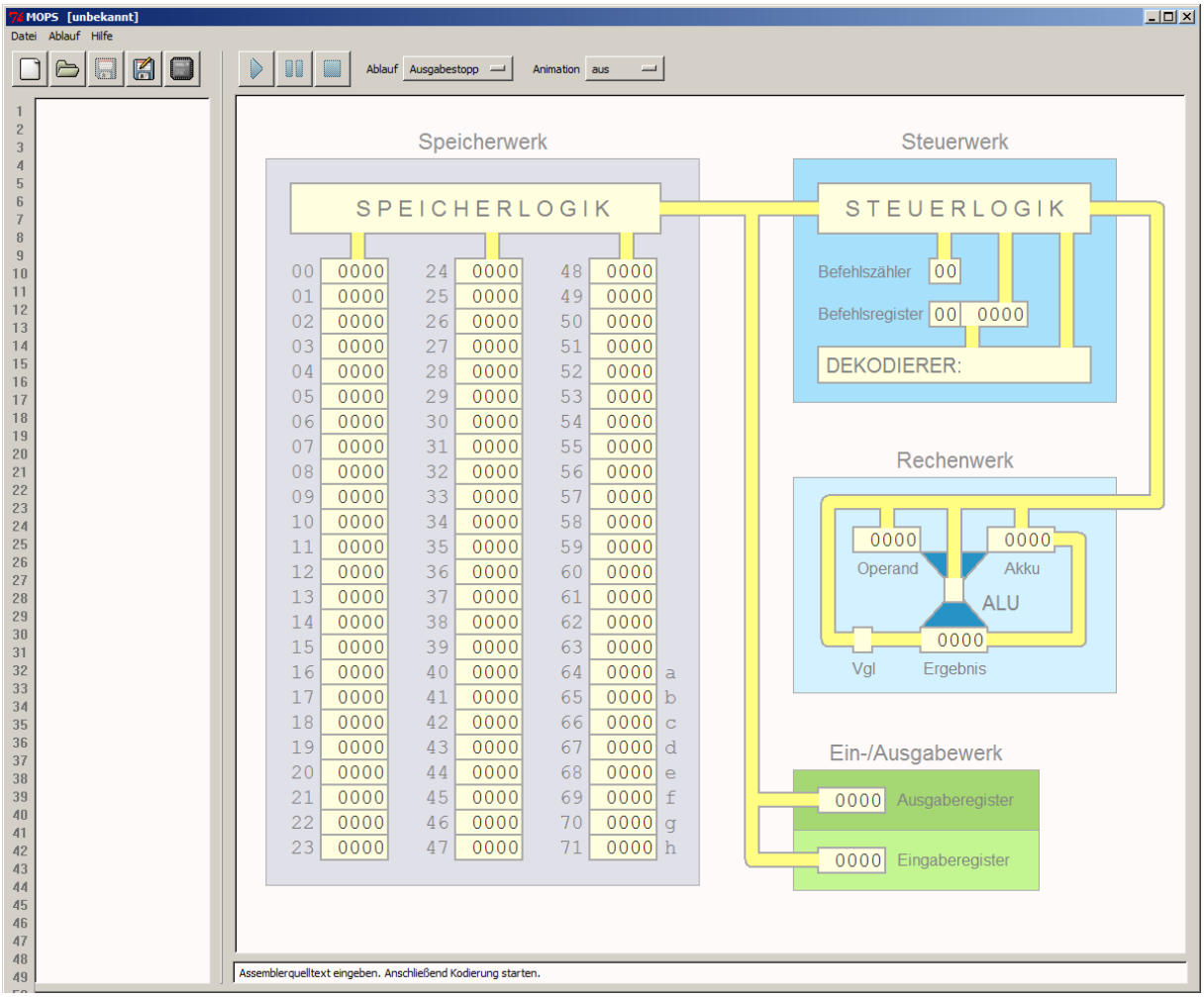
18.x.y. Weg zu MOPS / Installation

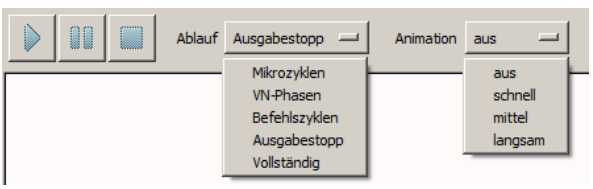
integriert in IoStick von Timo HEMPEL

als Download von der Hersteller-Seite

zum aktiven Arbeiten benötigt das Programm einen installierten Python-Interpreter ab Version 3.2.x (für die Umsetzung der Assembler-Befehle in Maschinen-Code)

18.x.y. Kurzvorstellung MOPS





Aufgaben:

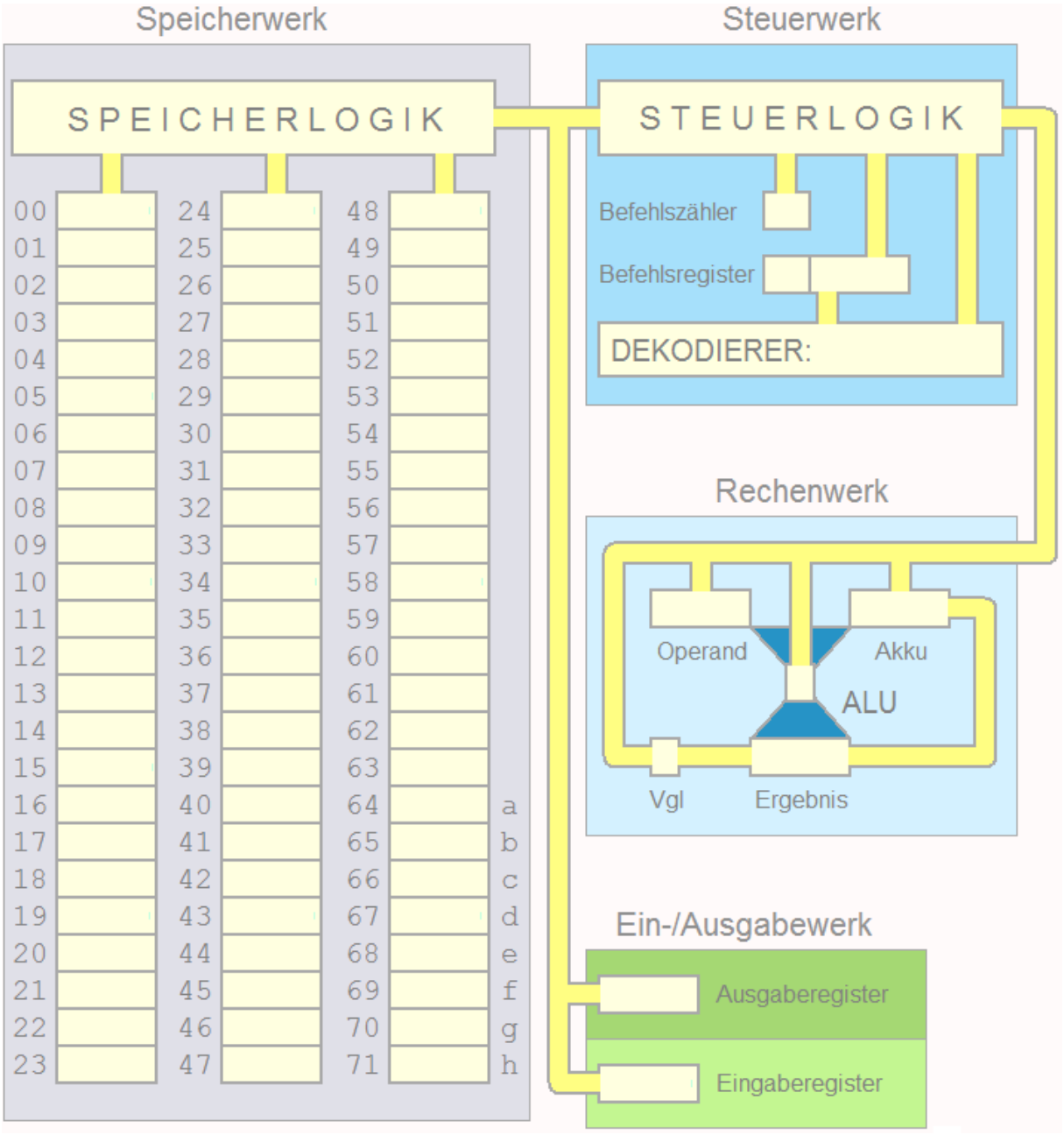
1. ***Drucken oder kopieren Sie die nachfolgende Seite (Arbeitsblatt MOPS)!***
2. ***Laminieren Sie sich die Seite! Alternativ kann die Seite auch in einer Klarsicht-Folie benutzt werden.***
   Ein Beschreiben des folierten Arbeitsblattes ist mit Non-permanent-Stiften möglich! Mit Küchenpapier oder einem einfachen Putzlappen kann gelöscht und korrigiert werden.
3.

Arbeitsblatt für MOPS-Spiel und die Dokumentation von Programmier-Aufgaben und Simulationen

Nr.	Assembler-Befehl
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

Nr.	Assembler-Befehl
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	

Nr.	Assembler-Befehl
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	



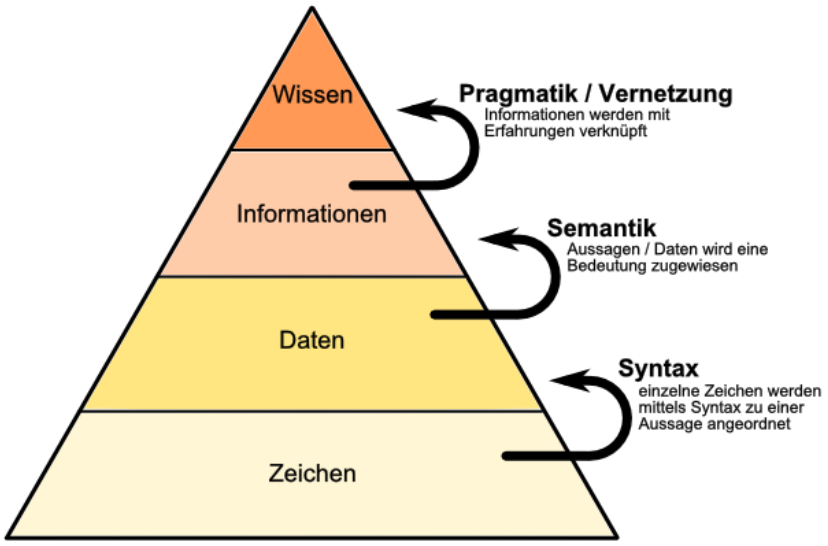
Q: MOPS-Programm; leicht verändert: dre

(c,p) 2015 – 2019 lsp: dre

ohne Jahrgangs-stufe

27. Alles, was noch keinen Platz gefunden hat

Wissens-Pyramide



Q: nach HERMANN 2012 IN: Log IN 187/188 (2017), S. 39 (leicht geänd.: dre)

x.y. Digitales Storytelling

x.y.z. digitales Storytelling mit Adobe Spark

online App → https://spark.adobe.com oder: → https://spark.adobe.com/sp/

für fast alle Gestaltungs-Aufgaben geeignet

Erstellen eines eMail-Accounts für ein Projekt oder eine Arbeitsgruppe, dadurch keine erzwungene Angabe von Personen-Daten

mit einem Schul-Accout lassen sich Schüler-Login's an der Schule mit denen von Adobe verknüpfen, dadurch erhält man auch die Premium-Funktionen (! Datenschutz beachten!!!)

3 Lizenz-Stufen

- **freie Version**

  permanentes Adobe Spark Logo

- **Individual-Version**

- **Team-Version**

Basis-Start: Kreieren …

- einer Grafik (create a graphic)
- einer Webseite (create a web page)
- eines Video (create a video)

mögliche Medien-Arten (alphabetisch)

- **Announcement**
- **Blog Graphic**
- **Brochure**
- **Business Card**
- **Campaign Video**
- **Card**
- **Case Study**
- **Collage**
- **Cover**
- **Etsys Cover**
- **Eventbrite Cover**
- **Explainer Video**
- **Facebook Ad**
- **Facebook Cover**
- **Facebook Post**
- **Flyer**
- **Infographic**
- **Info Page**
- **Instagram Ad**
- **Instagram Post**
- **Instagram Story**
- **Intro Video**
- **Invitation**
- **Lesson Plan video**
- **Menu**
- **Newsletter**
- **Photo Collage**
- **Photo Journal**
- **Pinterest Post**
- **Playlist Cover**
- **Portfolio**
- **Poster**
- **Presentation**
- **Promo Video**
- **Proposal Cover**
- **Recap Video**
- **Report**
- **Report Cover**
- **Slide Show**
- **Social Graphic**
- **Statistic Graphic**
- **Teaser Video**
- **Twitter Cover**
- **Twitter Post**
- **YouTube Thumbnail**

lt. Startseite Adobe Spark

Erstellen eines Video's mit Adobe Spark Video

1. Erstellen eines handschriftlichen Story Boards

enthält Reihenfolge und Zeitplan
enthält notwendige Medien (Foto's, Grafiken, Pictogramme, Titel von Musikstücken, Video's, …)
enthält einzusprechende Texte

2. Auswahl des passenden Projektes; Namens-Vergabe; Auswahl einer Vorlage

3. Hinzufügen von Medien-Material

Video's müssen im MP4-Format vorliegen
über Bearbeiten () lassen sich Positionen usw. anpassen
Fremdmaterial wird automatisch im "" mit den Lizenz- bzw. Quellen-Daten aufgeführt

4. Vertonung / Aufsprechung von Texten

über das Mikrophon-Symbol
jederzeit wiederholbar, wenn die Aufnahme verbessert / verändert werden soll

5. Hinzufügen einer Seite / Ändern der Seiten-Reihenfolge

6. Probelauf

Preview
Kontrolle durch Fremdpersonen

7. Exportieren / Veröffentlichen

share
Kategorie muss gewählt werden (optimal: Education)
get noticed auf off setzen, wenn Projekt für andere Nutzer unsichtbar bleiben soll
Link-Erzeugung (Link kann dann an potentielle Nutzer weitergegeben werden)

anschließend klassischer Download (zum internen Verteilen und zur Datensicherung (Zensierung))

Erstellen einer Info-Grafik mit Adobe Spark Post

Verwenden der Vorlagen (Beispiele) oder der leeren Vorlage

Hinweis: hier gibt es eine "Zurück" und "Vor"-Schaltfläche (Undo" und "Redo")

1. Auswählen eines Basis-Formates

Vorlagen für viele typische Verwendungen

2. Einfügen eines Bildes

quasi Basis-Bild
auch Piktogramme (besser mit engl. Suchbegriffen!)

3. Gestalten des Bildes

Veränderbar sind Größe, Position, Orientierung / Drehwinkel
Text-Box kann nach Doppel-Klick mit eigenen Texten gefüllt werden; Größe, Schriftgröße usw. veränderlich)
setzen von Color-Kombinationen (einstellbar)
einstellen von Layout-Vorlagen

es lassen sich weitere Objekte (Bild, Text, Logo, Piktogramm) hinzufügen

4. Exportieren / Veröffentlichen

Teilen

oder Download
z.B. auch zum Ausdrucken (Flyer, Einladungen, …)

Erstellen einer Info-Grafik mit Adobe Spark Page

1. Betitelung der Seite

2. Hinzufügen weiterer Elemente

möglich sind Texte, Video, Links,

3. Vorschau

4. Veröffentlichung

nur teilen möglich

Link kann z.B. per eMail verteilt werden oder in eine andere Webseite eingebunden werden

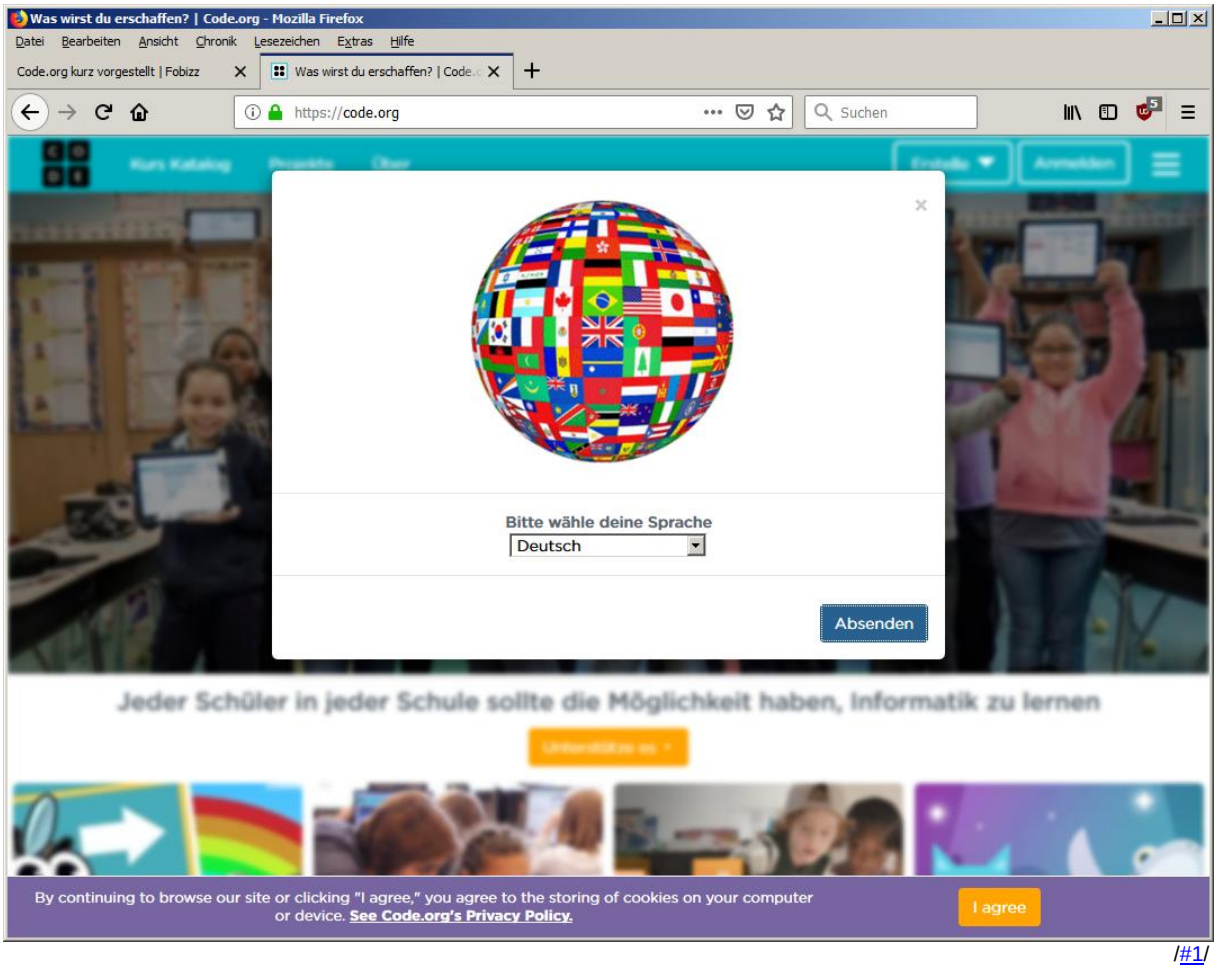
x.y. diverse Programmier-Tools

x.y.z. Code.org

nach Q und Info's: Code.org; FoBizz-online Weiterbildung ""
alle Browser-Bildschirm-Foto's: Q: code.org → /#1/

gemeinnützig; Sitz in USA; gegründet 2013

Sprach-Auswahl beim ersten Start oder sonst unten links



/#1/

Übersetzungen unvollständig und z.T. kryptisch; i.A. aber erkennbar
Video's sind englisch mit Untertitel (wobei die gewählte Sprache nicht immer auch angezeigt wird!)

zusätzliche Materialien weitesgehend englisch, einige auch deutsch (s.a. Links am Ende dieses Abschnitts)

für eine optimalere Nutzung am immer gleichen Computer sollten die Browser-Cookies akzeptiert werden ("I agree" unten rechts)

nach Anmeldung auch Kurse belegbar

gedacht als durchgehende Kurse von Unterstufe bis Sek.I und II
lt. Angabe von der Seite aufeinander aufbauend und den vorgelagerten Kurs voraussetzend

innerhalb der Kurse verschiedene Aktivitäten mit und ohne Computer

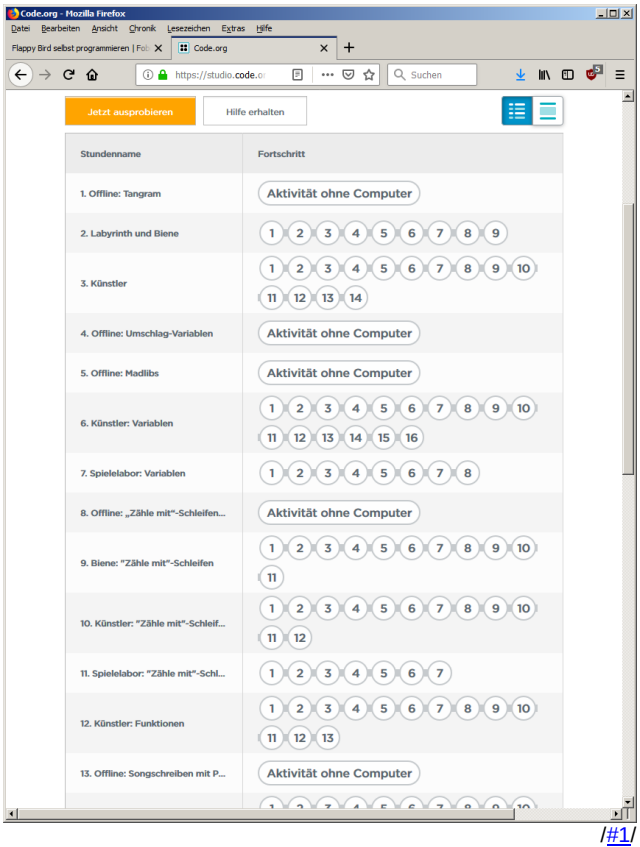
die Aufgaben am Computer sind meist spielerische Programmier-Aufgaben

Anzeige / Bühne ist links zu sehen

Objekte müssen bestimmte Aufgaben erfüllen, dazu stehen bestimmte Fähigkeiten in Form von Blöcken bereit
die Blöcke

Blöcke sind mittig angezeigt, rechts ist die Editier-Fläche

Fortschritt wird in zwei verschiedenen Formen angezeigt
mehr für den Computer / PC optimiert eine Kurs-Abschnitt-Übersicht



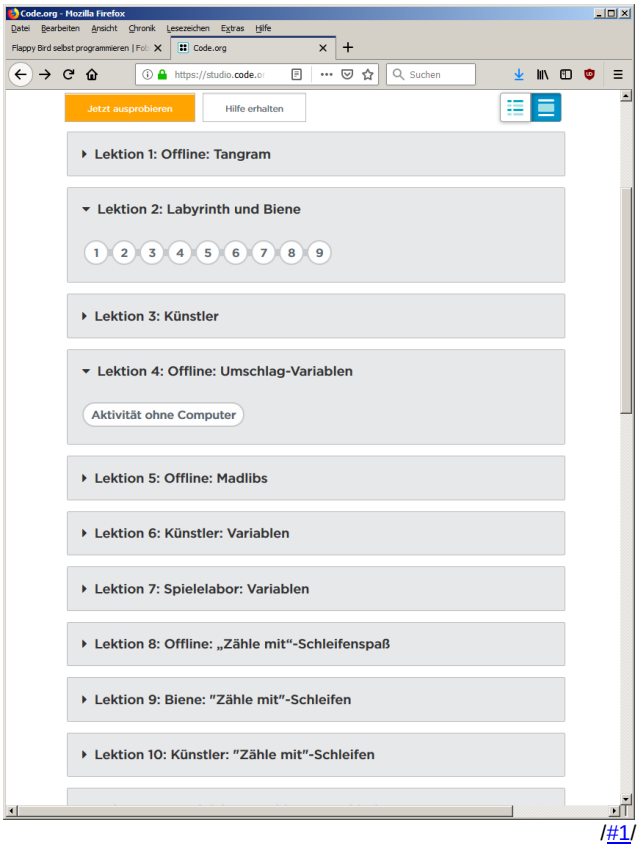
/#1/

und vorrangig für die Nutzung auf Tablet's gedacht die Ansicht mit Ausklapp-Menü's (Hamburger-Menü)

Abgeschlossene Aktivitäten oder Kurs-Bestandteile werden angezeigt

auch offline-Aufgaben

Kurse durch "Stundenpläne" begleitet
auch als (engl.) PDF download-bar



/#1/

x.y.z.1. Flappy Bird

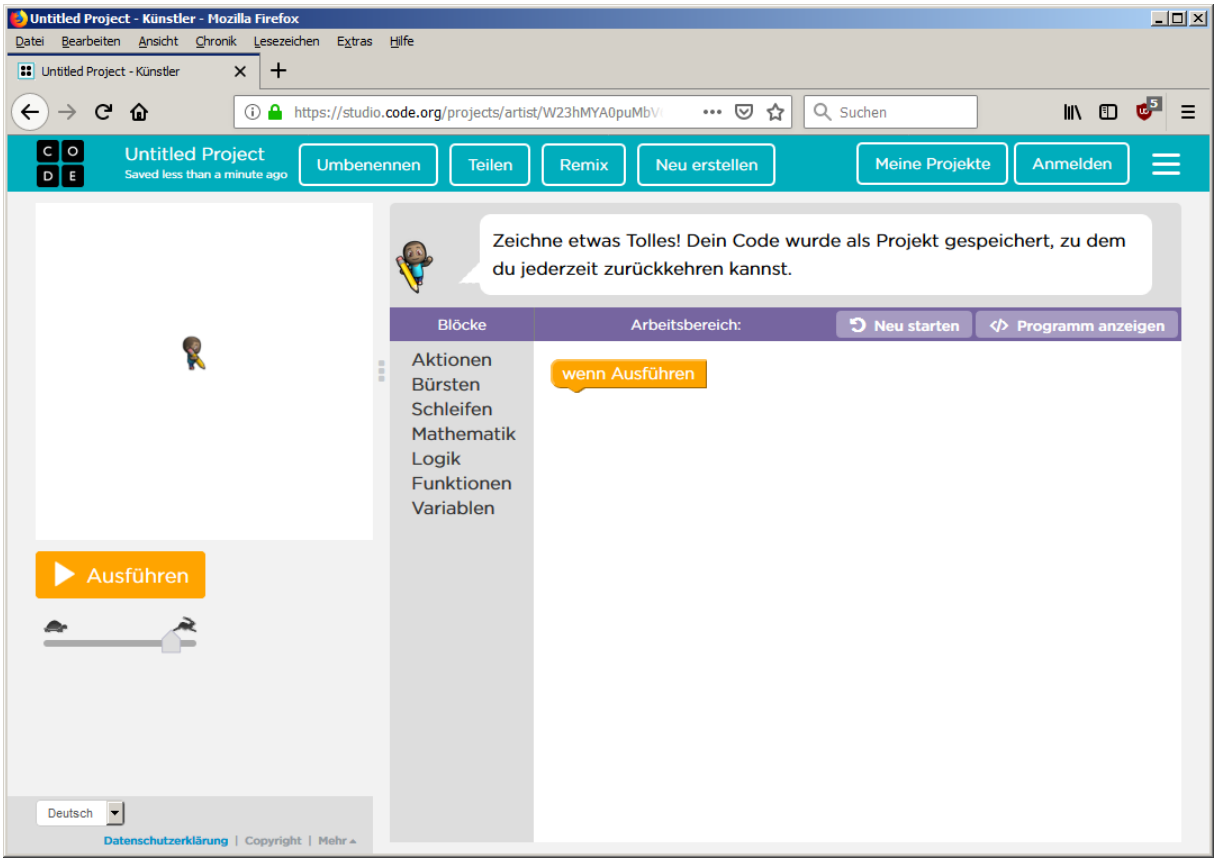
Direkt-Start → https://studio.code.org/flappy/1 mit Einführungs-Video
Sequenz mehrerer Programmier-Aufgaben mit immer mehr Bausteinen und höherer Schwierigkeit
Nutzung ohne Registrierung möglich
Niveau Grundschule ab Klasse 2 – also in unserem Skript-Umfeld eher als eine spaßige Variante für eine Veretungsstunde; gut als Einstieg, um die Arbeitstechniken zu üben und die Programmier-Oberfläche kennen zu lernen
englisches Einführungs-Video mit deutschen Untertiteln
Block-orientiert
Programmier-Spiel
Ziel ist es, den Vogel so zu steuern, dass er in das Ziel gelangt

Links (zu Programmier-Spielen):
https://code.org/learn (ähnliche Programmier-Spiele)
http://compute-it.toxicode.fr/ (Programmier-Spiele mit Schleifen (für etwas ältere Schüler))

x.y.z.2. Sprite Lab

x.y.z.3. Artist

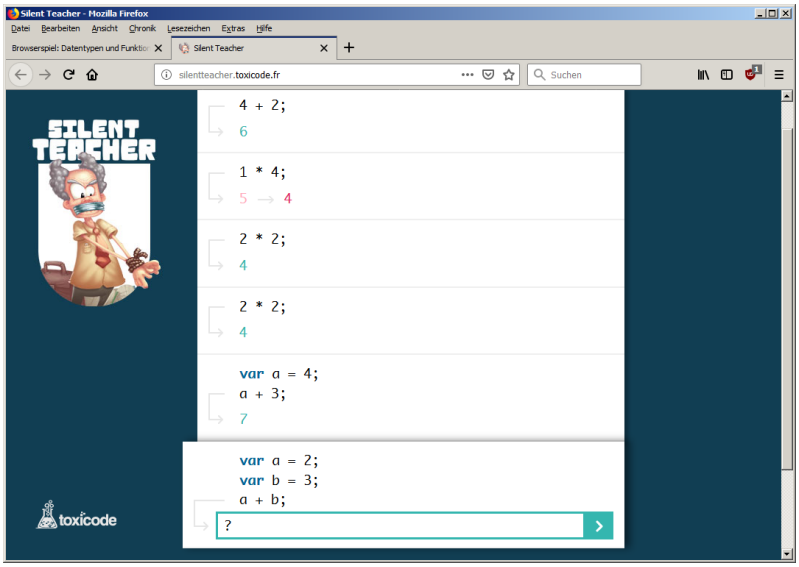
weniger verspielt
praktisch Scratch-entsprechend
modernes Design
gut mit PC und dem Tablet zu bedienen
Nutzung auf Smartphone eher problematisch



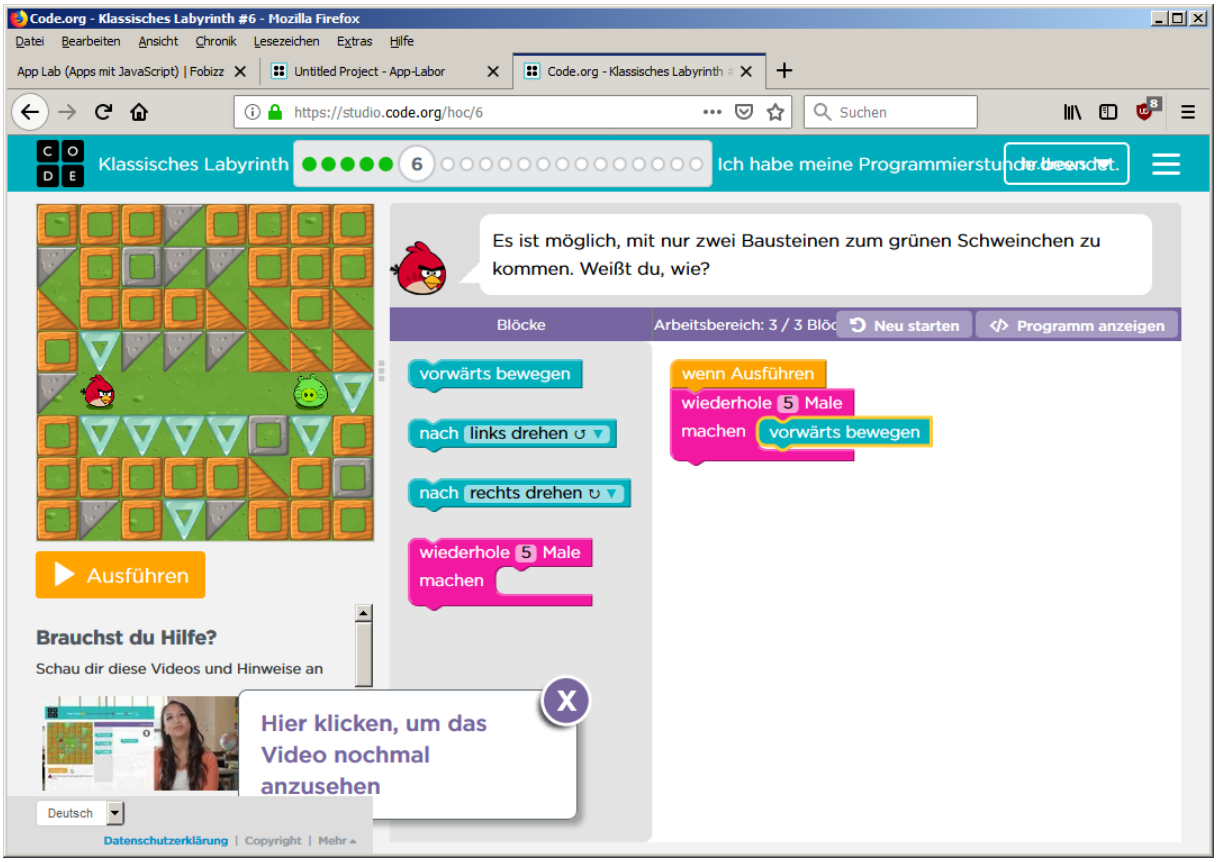
Links:
https://code.org/ (Startseite)

spielerische Einführung des Variablen-Konzepts und Daten-Typen

mit SilentTeacher von toxicode.fr (→ http://silentteacher.toxicode.fr/)
für weiterführende Schule

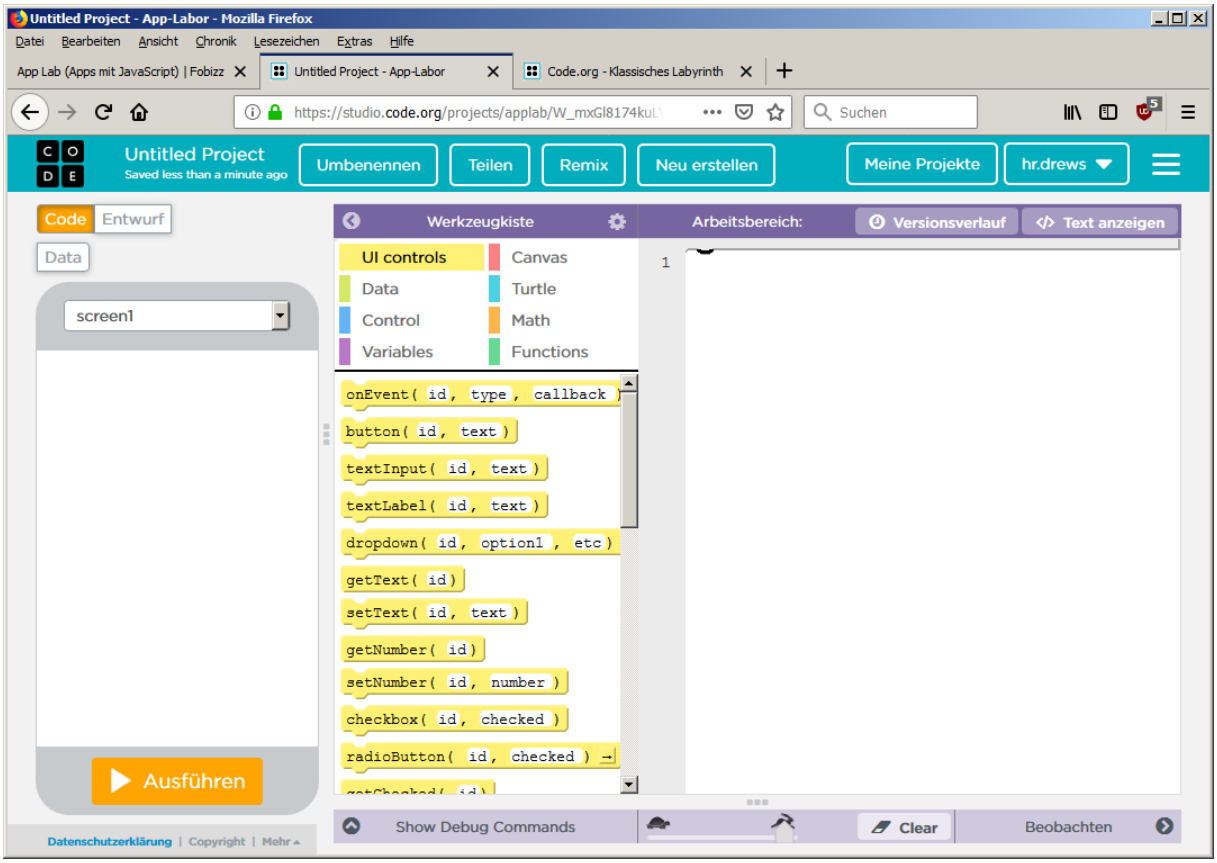

Silent Teacher - Mozilla Firefox
SILENT TEACHER
4 + 2;
6
1 * 4;
5 → 4
2 * 2;
4
2 * 2;
4
var a = 4;
a + 3;
7
var a = 2;
var b = 3;
a + b;
?
toxicode

Links:


Code.org - Klassisches Labyrinth #6 - Mozilla Firefox
https://studio.code.org/hoc/6
Klassisches Labyrinth
6
Es ist möglich, mit nur zwei Bausteinen zum grünen Schweinchen zu kommen. Weißt du, wie?
Blöcke
Arbeitsbereich: 3 / 3 Blöc
Neu starten
Programm anzeigen
vorwärts bewegen
nach links drehen
nach rechts drehen
wiederhole 5 Male machen
wenn Ausführen
wiederhole 5 Male
machen
vorwärts bewegen
Ausführen
Brauchst du Hilfe?
Schau dir diese Videos und Hinweise an
Hier klicken, um das Video nochmal anzusehen
Deutsch
Datenschutzerklärung | Copyright | Mehr

x.y.z.4. App-Lab -- App-Programmierung mit JavaScript

AppLab; Registrierung ist notwendig
(→ https://code.org/educate/applab)
es gibt auch einige Aufgaben / Spiele ohne Registrierung
kurzes Einführungs-Video (engl.) mit deutschen Untertiteln
ab 6./7. Klasse
Ziel ist das Erstellen einer eigenen (web-)App für's Smartphone



Code- und Design- bzw. Entwurf-Bereich
extra Daten-Bereich für eigene Daten

Umschaltung zwischen Block- und Code-Ansicht
Programmiersprache ist JavaScript

Ein Verteilen der App's geht nur über Kopieren oder Teilen per eMail oder Messanger.

x.y.z. in vielen Sprachen unterwegs - Programmieren mit repl.it

→ https://repl.it
praktisch ein Text-Editor für mehrere Programmier- und Formatier-Sprachen

scheinbar seit Neuem nur noch mit Registrierung zu Nutzen

der Bildschirm ist in drei Bereiche gegliedert
links Werkzeug-Leiste und Datei-Auswahl
mittig das Editier-Fenster
rechts eine (linux-)Konsole (entspricht: Eingabeaufforderung, Shell, …)

verfügbare Sprachen

- **Clojuge** Lisp-Dialekt in einer virtuellen Java-Maschine
- **Haskell** funktionelle Programmierung
- **Kottin** Programmiersprache mit statischen Datentypen, die mit JAVA und android zusammenarbeitet
- **QBasic** einfache strukturierte Programmiersprache für Anfänger
- **Forth** Stack-orientierte Programmiersprache
- **LOLCODE**
- **BrainF** Programmierung von TURING-Maschinen
- **Emoticon**
- **Bloop** Programmierung gebundener Schleifen
- **Unlambda**
- **Lua** Programmiersprache mit verschiedenen Programmier-Paradigmen
- **JavaScript**
- **CoffeeScript**
- **Scheme** dynamischer LISP-Dialekt
- **APL** Feld-orientierte Programmierung
- **Python 2.x** ältere Python-Versions-Reihe
- **Ruby** dynamische, Objekt-orientierte Programmierung
- **Roy**
- **PHP** Server-basierte Script-Sprache
- **Python (3.x)** aktuelle und empfohlene Python-Version
- **Node.js**

verfügbare Sprachen (Fortsetzung)

- **Enzyme** JavaScript-Testwerkzeug für React
- **Go**
- **Java** Objekt-orientierte, Plattform-übergreifende Programmierung
- **C++** universelle Programmiersprache
- **C++11** universelle Programmiersprache
- **C** imperative, Maschinen-nahe Programmierung
- **C#** universelle Programmierung für .Net-Framework (microsoft)
- **F#** funktionelle Programmierung mit .Net-Framework (microsoft)
- **HTML, CSS, JS** Web-Seiten-Entwicklung
- **Rust** schnelle und sichere System-Programmierung
- **Swift** moderne Programmiersprache für Apple-Systeme
- **Python (with Turtle)** Python mit Turtle-Graphik
- **Jest** schmerzloses Testen von JavaScript
- **Django** Python-Framework
- **Express** JavaScript-Framework für Web-Applikationen
- **Sinatra** einfache Erstellung von Web-Applukationen mit Ruby
- **Ruby on Rails**
- **R** Entwicklungs-Umgebung für statistische Berechungen und Graphik
- **Next.js** einfaches Framework für statische und Server-basierte React-Applikationen

verfügbare Sprachen (Fortsetzung)

- **GatsbyJS**
- **React** JavaScript-Bibliothek zum Erstellen von Nutzer-Oberflächen
- **React Typescript** JavaScript-Bibliothek zum Erstellen von Nutzer-Oberflächen
- **React Reason**
- **bash** UNIX-Shell mit möglicher Programmierung
- **Quil**

x.y.z.1. Programmierung in QBasis

```
1 # Hallo Welt Programm
2
3 print(""Hallo Welt!")
4
```

x.y.z.2. Programmierung in Python

```
1 # Hallo Welt Programm          Kommentar
2
3 print("Hallo Welt!")          Ausdruck eines Textes
4
```

Blöcke – also Gruppen von Anweisungen (z.B. in Schleifen oder Verzweigungen werden um 4 Positionen eingerückt
Prinzip setzt sich für weitere Unter-Blöcke entsprechend fort
Variablen-Namen müssen mit einem Buchstaben beginnen, der restliche Teil darf Buchstaben, Ziffern und den Unterstrich enthalten. Es wird Groß- und Klein-Schreibung unterschieden. Variablen haben keinen Daten-Typ.
Eingaben erfolgen mit input(). Normalerweise werden die Eingaben als Text aufgefasst. Sie müssen in einen Zahlen-Typ umgewandelt werden.

```
1 # Eingabe und Ausgabe ei-     Kommentar
2 ner Zahl
3                               Ausdruck eines Textes
4 print("Hallo Welt!")
```

Verzweigungen erfolgen mit if gefolgt von einer Bedingung. Die Bedingung muss bei Ber berechnung einen logischen Wert (- also True oder False -) ergeben. In Python haben aber alle Ausdrücke und Werte auch einen Wahrheits-Wert. Hinter einem Doppelpunkt folgen dann eingerückt eine oder mehrere Anweisungen (als Block).
Für die Alternative steht ein else: zur Verfügung.

```
1  # Entscheidung ob Zahl ge-   Kommentar
2  rade ist
3                               Ausdruck eines Textes
4  print("Hallo Welt!")
```

Es gibt auch die Möglichkeit der Mehrfach-Verzweigung mit elif Bedingung:.
Wiederholungen / Schleifen sind in Python immer Kopf-gesteuert. Sie können abzählbar sein. Dann beginnen sie mit for Zählung:. Schleifen ohne klare Wiederholungs-Anzahl werden mit while Bedingung: erstellt.

```
1  # Wiederholung / Reihe       Kommentar
2
3  print("Hallo Welt!")        Ausdruck eines Textes
4
```

x.y.z.3. Programmierung in JavaScript

x.y.z.4. Programmierung in PHP

```
1  // Hallo Welt Programm
2
3  echo "Hallo Welt!"
4
```

x.y.z.5. Programmierung in Ruby

```
1  # Hallo Welt Programm
2
3  puts "Hallo Welt!"
```

```
4
```

x.y.z.4. Programmierung in Java

Java ist wohl die Objekt-orientierte Programmier-Sprache.

größerer Schreibaufwand, der für Anfänger kaum verständlich ist

```
1  class HalloWelt {
2  # Hallo Welt Programm
3
4  public static void main(String[] args) {
5    System.out.println(""Hallo Welt!");
6    }
7  }
8
```

x.y.z.1. Programmierung in

```
1  # Hallo Welt Programm
2
3  print(""Hallo Welt!")
4
```

x.y.z. thematische Link-Listen

x.y.z.a. Material / Material-Listen

http://schuelerlabor.informatik.rwth-aachen.de/materialsammlung-fuer-den-informatikunterricht (umfangreiche Liste mit tollen Projekten usw. für den Informatik-Unterricht (mit Zielgruppe und Kurz-Beschreibung))

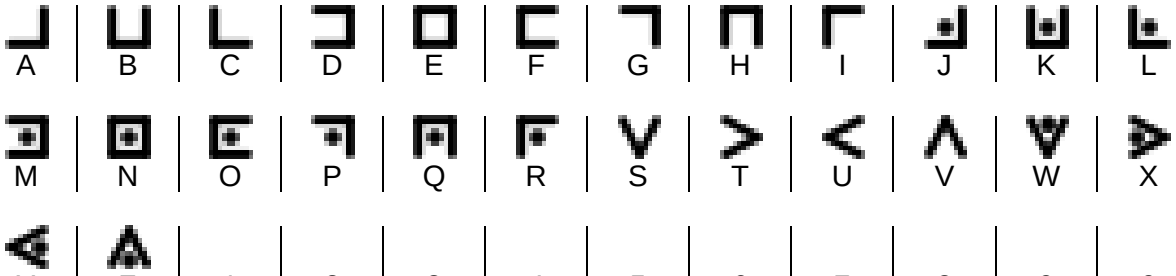
x.y.z.b. Tools

http://schuelerlabor.informatik.rwth-aachen.de/toolsammlung-fuer-den-informatikunterricht (umfangreiche Liste mit den klassischen App's und Programmen für den Informatik-Unterricht (mit Zielgruppe und Kurz-Beschreibung))

Anhang: Code- und Chiffre-Tabellen

Freimaurer-Code

A	B	C	D	E	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V	W	X
Y	Z	1	2	3	4	5	6	7	8	9	0



-Code

A	B	C	D	E	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V	W	X
Y	Z	1	2	3	4	5	6	7	8	9	0

Vorlage

A	B	C	D	E	F	G	H	I	J	K	L
M	N	O	P	Q	R	S	T	U	V	W	X
Y	Z	1	2	3	4	5	6	7	8	9	0

Literatur und Quellen:

/1/
ISBN

/*/ Curriculares Konzeptpapier für die Einführung des Faches Infromatik und Medienkunde in den Jahrgangsstufen 5 bis 10 an Regionalen Schulen, Gesamtschulen und Gymnasien im Modellvorhaben "Integrierte Berufsbildung" (Juli 2017)
Ministerium für Bildung, Wissenschaft und Kultur Mecklenburg-Vorpommern

/#/ (KMK-Papier: "digitale Kompetenzen" (2017))

/A/ Wikipedia
http://de.wikipedia.org

Die originalen sowie detailliertere bibliographische Angaben zu den meisten Literaturquellen sind im Internet unter http://dnb.ddb.de zu finden.

weitere Links:
http://www.bitsbook.com/excerpts/ (Buch (engl.): ABELSON, LEDEEN, LEWIS: Blown to Bits) (cc by nc sa)

Abbildungen und Skizzen entstammen den folgende ClipArt-Sammlungen:

/A/ 29.000 Mega ClipArts; NBG EDV Handels- und Verlags AG; 1997

/B/

andere Quellen sind direkt angegeben.

Alle anderen Abbildungen sind geistiges Eigentum:

/I/ lern-soft-projekt: drews (c,p) 1997-2020 lsp: dre

verwendete freie Software:

Programm	**Hersteller:**
Inkscape	inkscape.org (www.inkscape.org)
CmapTools	Institute for Human and Maschine Cognition (www.ihmc.us)

⊞- (c,p)2015 - 2020 lern-soft-projekt: drews -⊟
⊞- 18069 Rostock; Luise-Otto-Peters-Ring 25 -⊟
⊞- Tel/AB (0381) 760 12 18 FAX 760 12 11 -⊟