

IT2School

Gemeinsam IT entdecken



Aufbau-, Erweiterungs- und Methodenmodule zu IT2School

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Impressum IT2School

Herausgeber:

Wissensfabrik – Unternehmen für Deutschland e.V.
Ruthenstr. 23
67063 Ludwigshafen
www.wissensfabrik.de

Konzepterstellung und Umsetzung:

OFFIS e.V. – Institut für Informatik
Carl von Ossietzky Universität Oldenburg
Prof. Dr. Ira Diethelm
Melanie Schaumburg
Anatolij Fandrich

Nils Pancratz
Mirko Janssen
Mareike Daeglau
Annette Diruf
Estherk Nabo

Für inhaltliche Anregungen danken wir dem Arbeitskreis Bildung der Wissensfabrik – Unternehmen für Deutschland e.V. im Speziellen der Projektgruppe sowie allen Lehrkräften und Unternehmensvertreter*innen, die aktiv an der Pilotphase beteiligt waren oder uns Feedback gegeben haben:

Projektgruppe:

Leitung: Dr. Franziska Hutzler (Wissensfabrik)
Christiane Bauer (SAP)
Siegfried Czock (Bosch)
Michael Detmer (Wissensfabrik)
Matthias Dietel (IBM) Christian Greger (Trumpf)
Stefan Hüppe (Böhringer Ingelheim)
Axel Jentzsch (Wissensfabrik)
Peter Kusterer (IBM)
Reinhard Pittschellis (Festo Didactic)
Markus Riefling (BASF)
Ingmar Sassmann (BASF)
Thomas Schmitt (Deutsche Telekom Stiftung)
Birgit Schmitz (Deutsche Telekom Stiftung)
Andreas Schneider (Trumpf)
Peter Schubert (Softwarekontor)

Steffi Feldhaus (Berufskolleg Kohlstraße, Wuppertal)
Daniel Jungblut (SAP)
Martin Kempa (Gesamtschule Melsungen)
Markus Knak (Graf-Anton-Günther Gymnasium Oldenburg)
Bernadette Krüger (Oberschule Lemwerder)
Torsten Klaus (Trumpf)
Hannes Koderisch (Privatgymnasium Schwetzingen)
Harald Rothkirch (Gymnasium Neue Oberschule, Braunschweig)
Eva Nickel (Softwarekontor)
Gerburg Lubor (Softwarekontor)
Klaus-Dieter Neff (Leonardo da Vinci Gymnasium Neckargemünd)
Frank Röhr (Erich-Kästner-Gesamtschule Bochum)
Carsten Rohe (Gymnasium Damme)
Christiane Schicke (Inselschule Langeoog)
Tobias Stuckenberg (Paulusschule Oldenburg)
Armin Tischler (Gymnasium Damme)
Holger de Vries (KGS Rastede)
Prof. Dr. Carsten Schulte (Uni Paderborn)
Benjamin Piétza (FU Berlin)

Pilotierung:

Torsten Barth (Gemeinschaftsschule Lauenburgische Seen)
Nadine Bergner (RWTH Aachen - Schülerlabor Infosphere)
Eric Böhmfeld (Dräger)
Miriam Böhne (SAP)
Jens Eschen (Realschule Rhauderfehn)

Urheber- und Nutzungsrechte:

Dieses Werk bzw. Inhalt steht unter einer Creative Commons Lizenz (**Namensnennung, nichtkommerziell, Weitergabe unter gleichen Bedingungen**).



Darüber hinaus ist die Nutzung an Privatschulen und Bildungseinrichtungen freier Träger gestattet, auch wenn dafür Gebühren erhoben werden, solange die Materialien im Unterricht oder in Betreuungsangeboten für Schulkinder verwendet werden. Gleichermaßen gilt für

Fortbildungsangebote für Lehrkräfte. Alle Teile dieses Werkes sind vom Herausgebenden und von der für die Erstellung verantwortlichen Redaktion sorgfältig erwogen und geprüft worden. Eine Haftung des Herausgebenden bzw. der für die Redaktion verantwortlichen Institutionen für etwaige Personen-, Sach- oder Vermögensschäden, die sich aus dem Gebrauch dieses Werkes ergeben oder ergeben könnten, ist ausgeschlossen. In diesem Handbuch werden geschlechtsumfassende Formulierungen und der Genderstern verwendet. Sollte dies aus Gründen der Lesbarkeit an manchen Stellen nicht möglich sein, gelten die dort verwendeten Personenbezeichnung gleichermaßen für alle Geschlechter.

Bildnachweise:

Die Bildnachweise sind jeweils neben dem Bild angegeben. Ist dies nicht der Fall, stammen die Bilder von www.pixabay.com (Creative Commons - CC0) oder die Rechte liegen bei den Entwicklern des Konzeptes und der Wissensfabrik. Grafische Gestaltung: www.active-screen.de Illustration: Christoph J. Kellner, Animation / Illustration / Graphic Recording, studio animanova

Online verfügbare Aufbau-, Erweiterungs- und Methodenmodule

Modul A1 – Mobilfunk

Vom Mobilfunk zu Big Data

Modul A2 – Kryptologie

Kryptologie

Modul A3 – Programmieren II

Objektorientierte Programmierung mit Python

Modul E1 – IT Kinderleicht

IT und Informatik spielend entdecken

Modul E2 – Wearable

Smarte Kleidung selbst gestalten

Modul E3 – Robotik

BB8 selber bauen

Modul E4 – Webseiten

Erstellung von Webseiten

Modul M1 – Design Thinking

Modul M2 – Projektmethode

Die hier aufgeführten (klassischen) Module von IT2School werden durch die (neuen) Module zum Thema künstliche Intelligenz ergänzt. Diese sind im Ordner „IT2School Module zur künstlichen Intelligenz“ und online verfügbar.

IT2School

Gemeinsam IT entdecken



Modul A1 – Mobilfunk

Vom Mobilfunk zu Big Data

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	Vom Mobilfunk zu Big Data	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	5
4	Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters.....	5
5	Inhalte des Moduls.....	5
5.1	Die Funktionsweise von Mobilfunk.....	5
5.2	Visualisierung von Mobilfunkdaten mit Processing	7
5.3	Big Data.....	8
6	Unterrichtliche Umsetzung.....	9
6.1	Grober Unterrichtsplan.....	9
6.2	Stundenverlaufsskizzen	10
6.2.1	Wie funktioniert das Mobilfunknetz?	10
6.2.2	Das Smartphone und ortsbezogene Daten.....	11
6.2.3	Das Smartphone als Datenschleuder	12
6.2.4	Big Data	13
7	Einbettung in verschiedene Fächer und Themen	15
8	Anschlussthemen.....	16
9	Literatur und Links	16
10	Arbeitsmaterialien	16
11	Glossar.....	17

1 Vom Mobilfunk zu Big Data

Handy und Smartphone sind heute auch alltäglicher Begleiter von Kindern und Jugendlichen. Obwohl das Gerät für viele Kinder und Jugendliche nicht mehr wegzudenken ist, wissen nur die wenigsten, wie es technisch funktioniert und wie Daten übertragen werden.



Im Rahmen dieses Moduls befassen sich die Schülerinnen und Schüler mit der Funktionsweise des Mobilfunks. Sie erfahren dabei, wie Daten von einem Mobiltelefon zum nächsten übertragen werden. Im Anschluss werten die Schülerinnen und Schüler Mobilfunkdaten aus. Aus Zeit- und Ortsangaben versuchen sie so viele Informationen wie möglich über eine Person zusammenzutragen. Am Ende setzen sich die Schülerinnen und Schüler mit dem Datenschutz, sowie den Chancen und Risiken von Big Data auseinander.¹

Lernfeld/Cluster:	Daten erforschen
Zielgruppe/Klassenstufe:	4. bis 5. Klasse
	6. bis 7. Klasse
	X 8. bis 10. Klasse
	X 11. bis 12. Klasse
Geschätzter Zeitaufwand:	6 – 7 Einzelstunden
Lernziele:	<ul style="list-style-type: none">• Funktionsweise von Mobilfunk verstehen• Fachbegriffe verstehen und anwenden• Sich eine eigene Meinung zu Big Data bilden
Vorkenntnisse der Schülerinnen und Schüler:	Erforderlich: <ul style="list-style-type: none">• Kenntnisse über die Funktionsweise des Internets (ggf. sollte das Modul B2 bereits durchgeführt worden sein)• Kenntnisse im Programmieren im Idealfall sogar textuelle Programmierung (ggf. sollte das Modul B5 oder B7 bereits durchgeführt worden sein)
Vorkenntnisse der/des Lehrenden:	Erforderlich: <ul style="list-style-type: none">• Auswertung der Mobilfunkdaten sollte im Vorfeld von der Lehrkraft unbedingt selbst einmal durchgeführt werden.• Durcharbeiten der Materialien V2.3-4 und V2.5_L Empfohlen: <ul style="list-style-type: none">• Kenntnisse über die Funktionsweise des Internets (ggf. sollte das Modul B2 bereits durchgeführt worden sein)• Kenntnisse im Programmieren im Idealfall sogar textuelle Programmierung (ggf. sollte das Modul B5 oder B7 bereits durchgeführt worden sein)

¹ Wir bedanken uns herzlich bei Prof. Dr. Carsten Schulte (Uni Paderborn) sowie Benjamin Piéta und Christoph van Heteren-Frese vom Institut der Informatik der FU Berlin, die uns Teile ihres Konzepts zum Thema Mobilfunk zur Verfügung gestellt haben. Einzelne Teile wurden übernommen und für dieses Konzept angepasst.

Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters	<p>Empfohlen:</p> <ul style="list-style-type: none"> • Kenntnisse über die Funktionsweise des Internets und des Mobilfunks • Ggf. Kenntnisse über Datenschutzbestimmungen im eigenen Unternehmen
Sonstige Voraussetzungen:	<p>Erforderlich:</p> <ul style="list-style-type: none"> • Die Schülerinnen und Schüler benötigen sowohl in der Schule als auch Zuhause eine Internetverbindung. • Ausreichend Computer zur Durchführung von Recherchen sowie der Auswertung der Bewegungsdaten.

2 Warum gibt es das Modul?

Neue Posts bei Facebook lesen, das aktuelle Video des Lieblings-Youtubers ansehen oder das neueste Selfie bei Instagram veröffentlichen: Für viele Jugendliche ist die Nutzung von Handys und Smartphones alltäglich. Dies zeigt auch die aktuelle Ausstattung der Kinder und Jugendlichen. Laut der JIM-Studie 2015² besitzen 98% der Kinder und Jugendlichen von 12 bis 19 Jahren ein eigenes Mobiltelefon, die Mehrheit (92%) davon ein Smartphone. Das Smartphone wird in dieser Altersgruppe auch am häufigsten genutzt, um ins Internet zu gehen und hat damit den Computer/Laptop abgelöst (JIM, 2015).

Obwohl das Gerät für viele Kinder und Jugendliche nicht mehr wegzudenken ist, wissen nur die wenigsten, wie es technisch funktioniert und wie Daten übertragen werden.

Neben der technischen Unwissenheit gibt es auch eine große Wissenslücke im Bereich Tracking und Datenschutz. Egal ob wir mobil online sind, bestimmte Apps verwenden oder am heimischen PC sitzen - unser Surfverhalten, die verwendeten Geräte sowie der Zugriff auf beispielsweise Ortungsdaten erlauben heute einen tiefen Einblick in unser Privatleben.

Daher gehört das Wissen über die Funktionsweise von Mobilfunk bis hin zu Big Data zu den Grundlagen, um als mündige Bürgerin und mündiger Bürger die vernetzte Welt zu verstehen und darin selbstbestimmt und reflektiert agieren zu können.

In diesem Modul erfahren die Schülerinnen und Schüler etwas über die Struktur und Funktionsweise von Mobilfunk. Daneben befassen sie sich aber auch damit, wie unter anderem Mobilfunkdaten zur Informationsgewinnung herangezogen werden und welche Potentiale aber auch Risiken in Big Data stecken.

² Medienpädagogischer Forschungsverbund 2015: JIM-Studie - <http://www.mpfs.de/?id=676>

3 Ziele des Moduls

- Die Technik und Funktionsweise von Mobilfunkübertragung kennenlernen und verstehen.
- Wichtige Fachbegriffe verstehen (UMTS, Funkzelle, Big Data,...).
- Gesellschaftliche Veränderungen durch Mobilfunk und mobiles Internet reflektieren.
- Fundierte Auseinandersetzung mit den Chancen und Risiken von Big Data in ihrer persönlichen und gesellschaftlichen Relevanz.

4 Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters

Im *Modul A1 – Mobilfunk* hat die Unternehmensvertreterin/der Unternehmensvertreter mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

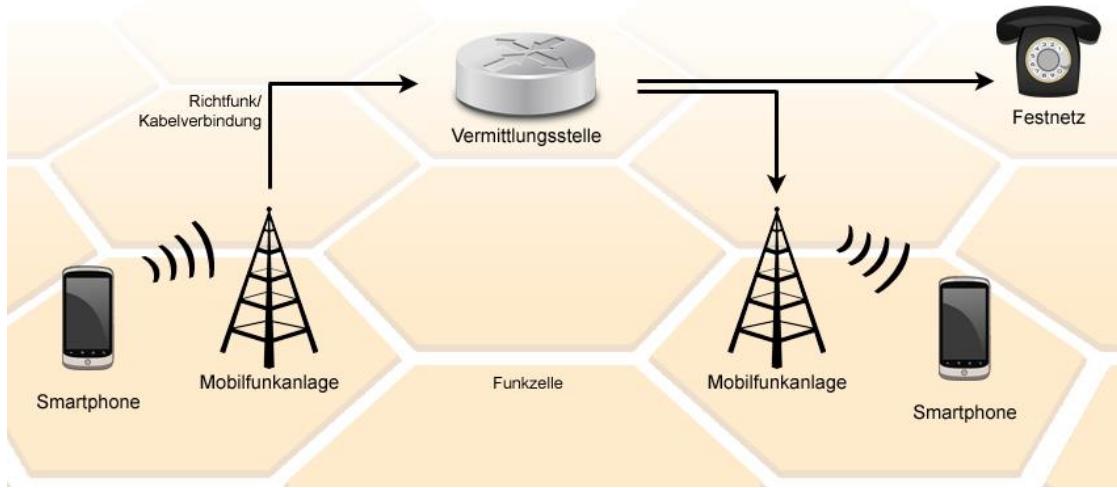
- Unterstützung der Lehrkraft, z.B. bei der Umsetzung der Experimente mit dem Handy
- Der/die UnternehmensvertreterIn kann berichten, wie in seinem/ihrem Unternehmen mit Daten umgegangen wird. Wie sichert sich das Unternehmen selbst ab, beispielsweise in Bezug auf Hacking und Datenmissbrauch (darf man das Smartphone auch privat verwenden?), oder sammelt das Unternehmen selbst Daten und wertet diese aus? Wenn ja, für welchen Zweck wird dies getan?
- Der/die UnternehmensvertreterIn kann zur Präsentation der Erklärvideos oder zur Debatte als Special-Guest eingeladen werden.

5 Inhalte des Moduls

5.1 Die Funktionsweise von Mobilfunk

Mit Hilfe eines Handys lassen sich Daten wie beispielsweise Gespräche, Texte oder Bilder als Funksignal übertragen. Dazu ist das gesamte Mobilfunknetz in kleinere Funkzellen unterteilt. In jeder Funkzelle steht eine so genannte Mobilfunkbasisstation oder Mobilfunkanlage, die sowohl senden als auch empfangen kann.

Schaltet man sein Handy ein und entsperrt die SIM-Karte mit der Eingabe der persönlichen PIN-Nummer, meldet sich das Handy automatisch mit einem Funksignal im Mobilfunknetz an. Mit dieser Anmeldung wird auch registriert, in welcher Funkzelle sich das Handy aktuell befindet.



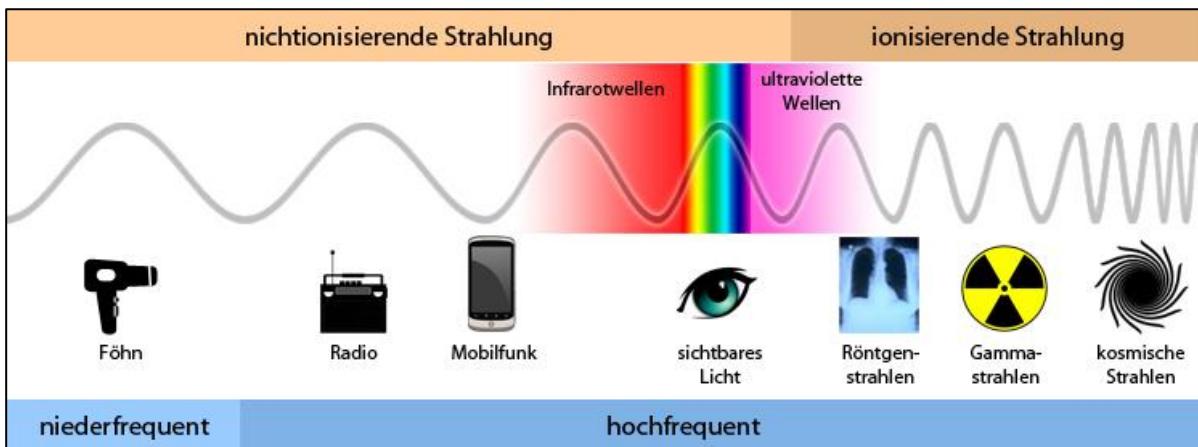
Will man nun telefonieren, wird der Anruf als Anfrage mit Hilfe von elektromagnetischen Wellen an die nächste Mobilfunkanlage gesendet. Von dort werden die Daten mit Richtfunk oder einer Kabelverbindung an einen Zentralcomputer übermittelt, der als Vermittlungsstelle dient. In diesem Zentralcomputer werden alle relevanten Mobilfunkdaten gespeichert wie z.B. welchen Netzanbieter man benutzt, in welcher Funkzelle man sich gerade aufhält und ob eine Rufumleitung aktiviert ist. Mit Hilfe dieser gespeicherten Daten übernimmt der Zentralcomputer die Vermittlung zur Mobilfunkanlage innerhalb der richtigen Funkzelle. Die Mobilfunkanlage übermittelt die Anfrage nun mit elektromagnetischen Wellen zum gewünschten Gesprächspartner. Nimmt der gewünschte Gesprächspartner ab, so werden die Daten in Sekundenschnelle hin und her übertragen.

Ist man nun mit seinem Handy unterwegs, meldet es sich immer automatisch mit einem Signal innerhalb der aktiven Funkzelle an. Die Ortungsdaten werden jedesmal an den zentralen Computer weitergeleitet und gespeichert. Auf diese Weise kann man überall erreichbar sein.

Die einzelnen Funkzellen sind unterschiedlich groß. Auf dem Land können sie mehrere Kilometer umfassen, in der Stadt nur wenige Meter. Dies liegt daran, dass eine einzelne Mobilfunkanlage nur eine begrenzte Menge an Daten verarbeiten kann. Daher werden in Gegenden mit einem hohen Datenaufkommen, wie in der Stadt, mehr Funkzellen benötigt. Trotzdem kann auch hier das Netz bei großen Menschenansammlungen überlastet sein, wie viele sicher schon einmal an Silvester oder anderen Gelegenheiten festgestellt haben.

Elektromagnetische Übertragung

Die Voraussetzung für Mobilfunk sind elektromagnetische Wellen. Es gibt viele verschiedene elektromagnetische Wellen, die sich in ihrer Wellenlänge unterscheiden. Je kürzer die Wellenlänge, desto höher wird die Frequenz, deren Einheit in Hertz (Hz) gemessen wird. Ein Hertz bedeutet eine Schwingung pro Sekunde.



Man kann zwischen niederfrequenter und hochfrequenter Strahlung unterscheiden.

Niederfrequente Strahlung tritt bei Stromfluss auf, beispielsweise wenn man seinen Fön anschaltet oder andere Haushaltsgeräte. Diese Frequenz liegt im Bereich von 0 – 10 KHz. Hochfrequente Strahlung wird für die Funkübertragung genutzt, wie beim Radio oder Mobilfunk. Die Strahlung liegt im Bereich von 10 – 300 Gigahertz.

Die verschiedenen Mobilfunkstandards beruhen auf unterschiedlichen Frequenzbereichen. Der GSM-Standard (Global System for Mobile Communication) gehört zur so genannten 2. Generation (2G) und bewegt sich im Frequenzbereich von ca. 900 Megahertz. Der Standard UMTS (Universal Mobile Telecommunications System) liegt zwischen 199 und 2100 Megahertz und die neueste 4. Generation - LTE (Long Term Evolution) nutzt die Frequenzbereiche um 800 Megahertz, 1,8 Gigahertz und 2,6 Gigahertz.

5.2 Visualisierung von Mobilfunkdaten mit Processing



Abbildung: BÜNDNIS 90/DIE GRÜNEN - Foto von Malte Spitz

Quelle:
https://commons.wikimedia.org/wiki/File:20110808_MalteSpitz.png

In einer Arbeitsphase dieses Moduls visualisieren die Schülerinnen und Schüler vorhandene Mobilfunkdaten. Dafür werden das Programm *Processing*, ein für *Processing* geschriebenes Script und die Mobilfunkdaten benötigt. (Alle notwendigen Daten und Unterlagen wurden bereits zusammengetragen und sind in diesem Modul enthalten.)

Die abgebildeten Mobilfunkdaten stammen von Malte Spitz, einem aktiven Politiker³. Die Daten wurden von T-Mobile gespeichert und wurden nach einer Klage durch Malte Spitz herausgegeben. Er wollte damit aufzeigen, welche Informationen aus gesammelten Daten (besonders im Zusammenhang mit der Vorratsdatenspeicherung) ermittelt werden können.

Wichtig ist, dass es sich bei Processing um eine Programmierumgebung handelt, in der mit Java gearbeitet wird. Programmierkenntnisse sind jedoch nur in geringem Maße notwendig, da das enthaltene Processing-Skript die Visualisierung und Filterung der Daten sehr vereinfacht.

³ Partei Bündnis 90/Die Grünen, von 2006 bis 2013 Beisitzer im Bundesvorstand

```

P Visualisierung_von_Mobilfunkdaten | Processing 1.5.1
File Edit Sketch Tools Help
STANDARD
Visualisierung_von_Mobilfunkdaten Amountbubble Button OpenStreetMapProvider Trackpoint
import java.text.*;
import java.util.*;

/*
*****
Visualisierung der Handydaten von Malte Spitz
*****
***** Bedienungsmöglichkeiten *****
Pfeiltasten    -> Bewegen der Karte
+/-Mausrad     -> Ein/Auszoomen
Leertaste       -> Karte wird auf den letzten Bewegungspunkt zentriert
b               -> eigene Markierung auf Karte setzen
m               -> Menü ein/ausschalten

```

5.3 Big Data

Das Smartphone ist heute für viele Menschen ständiger Begleiter, auch schon für Kinder und Jugendliche. Ohne die Verwendung von Ortsangaben würde Mobilfunk nicht funktionieren. Nur aufgrund der Speicherung der aktuellen Funkzelle in einem Zentralcomputer können Anrufe zum gewünschten Gesprächspartner weitergeleitet werden.

Das Smartphone ist heute aber vielmehr als nur ein mobiles Telefon, bei dem Ortungsdaten anfallen. Der kleine Taschencomputer ist mit dem Internet verbunden und bietet zahlreiche Programme bzw. Applikationen (sogenannte Apps) zum Spielen, Arbeiten sowie Kommunizieren und egal, was wir gerade verwenden, wir hinterlassen immer eine Spur von Daten.

Diese riesigen Datenmengen, die wir durch die Nutzung des Internets, sozialer Netzwerke, Ortungsdaten, Fitnesstracker oder auch durch Sensoren an Autos etc. produzieren sind mittlerweile zu groß, um sie mit klassischen Datenbanken oder Analysewerkzeugen zu verarbeiten. Der Begriff Big Data steht für diese riesigen Datenmengen, die mit Hilfe massiver Rechenleistung ausgewertet werden und in vielen Bereichen Anwendung findet.

Ein großer und bekannter Bereich ist die Werbung. Firmen wie Google oder Facebook nutzen unserer Daten wie beispielsweise personenbezogene Daten (Alter, Geschlecht, Wohnort), unsere Suchanfragen und Kontakte, um ein genaues Profil von uns zu erstellen. Mit dieser Analyse ist es möglich personalisierte Werbung zu schalten.

Im Verkehrsbereich können uns Dank dieser Auswertungen z.B. Staus vorausgesagt und alternative Routen vorgeschlagen werden. Im Medizinsektor hilft Big Data u.a. den Verlauf von Epidemien zu prognostizieren.

Spätestens seit Edward Snowden wissen wir, dass auch Staaten, wie beispielsweise die USA, elektronische Kommunikation erfassen und auswerten, darunter auch die Daten zahlreicher Regierungschefs wie beispielsweise Angela Merkel.

Diese Beispiele zeigen welche Vor- und Nachteile Big Data mit sich bringt. Eine besondere Herausforderung spielt dabei der Datenschutz, denn bei vielen Daten die erhoben werden, geht es um Personenbezogene Daten. Der Datenschutz wird daher in Zukunft eine der größten Herausforderungen sein, die mit Big Data in Verbindung stehen.

6 Unterrichtliche Umsetzung

Dieses Modul ist gekennzeichnet durch handlungsorientierten und schüleraktivierenden Unterricht. Zu Beginn steht die Produktion eines Erklärvideos im Vordergrund. Die Schülerinnen und Schüler recherchieren eigenständig wie Mobilfunk funktioniert und stellen die gewonnenen Erkenntnisse in Form eines Videos dar.

Im mittleren Teil des Moduls befassen sich die Schülerinnen und Schüler mit der Auswertung von Mobilfunkdaten. In der Auseinandersetzung mit diesen Daten erkennen die Schülerinnen und Schüler, dass man aus vermeintlich harmlosen Metadaten schon eine ganze Reihe an Informationen über eine Person bekommen kann. Darüber hinaus reflektieren sie ihren eigenen Umgang mit persönlichen Daten.

Im letzten Teil steht das Thema Big Data im Mittelpunkt. Die Schülerinnen und Schüler recherchieren selbstständig zu den Vor- und Nachteilen von Big Data und befassen sich mit verschiedenen Einsatzgebieten. Die Ergebnisse werden schließlich dem Plenum vorgestellt und die Lehrkraft hat die Möglichkeit eine Diskussion anzuregen.

6.1 Grober Unterrichtsplan

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Experimente mit dem Smartphone/ Handy
Erarbeitung	Wie funktioniert Mobilfunk? Produktion eines Erklärvideos
Vertiefung	Mobilfunkdaten erforschen, Was kann man aus Ortungsdaten herauslesen, Analyse von Mobilfunkdaten.
Vertiefung	Reflexion über den eigenen Umgang mit Daten und Bedeutung von Datenschutz
Vertiefung	Vom Mobilfunk zu Big Data, die Schülerinnen und Schüler befassen sich mit den Chancen und Risiken von Big Data, sowie mit den Möglichkeiten des Datenschutzes.

6.2 Stundenverlaufsskizzen

Abkürzungen/Legende

AB = Arbeitsblatt/Arbeitsblätter; L = Lehrkraft; MuM = Mitschülerinnen und Mitschüler; SuS = Schülerinnen und Schüler;

UV = Unternehmensvertreterin/Unternehmensvertreter

6.2.1 Wie funktioniert das Mobilfunknetz?

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
15 Min.	Einstieg	Lehrerpräsentation eines technischen Phänomens	Begrüßung der SuS; Im Anschluss Experimente mit dem Handy/ Smartphone Frage: Ist das Handy erreichbar, wenn man es anruft? 1. Experiment: Das Smartphone wird in eine vollständig verschlossene Plastikhülle bzw. Gefrierbeutel getan und dann in ein Wasserbad gelegt. Anschließend ruft L das Handy/Smartphone an. (Vorsicht: Der Verschluss der Plastikhülle bzw. Gefrierbeutel sollte vorher ohne Handy/Smartphone getestet werden.) - <i>Das Handy/Smartphone sollte erreichbar sein.</i> 2. Experiment: Das Handy/Smartphone wird in eine Aluminiumdose gelegt und dann von L angerufen. - <i>Das Handy/Smartphone sollte erreichbar sein.</i> 3. Experiment: Das Handy/Smartphone wird vollständig und dick mit Aluminiumfolie umwickelt und anschließend in die Aluminiumdose gelegt. (Es sollte darauf geachtet werden, dass keine freien „Löcher“ entstehen.) - <i>Das Handy/Smartphone sollte nicht erreichbar sein.</i>	Zwei Handys oder Smartphones (Anrufer und Angerufener), wasserdichte Plastikhülle bzw. Gefrierbeutel, Gefäß mit Wasser, Aluminiumfolie, Aluminiumdose



15 Min.	Einstieg	Plenum	Besprechung Arbeitsauftrag: Wie funktioniert Mobilfunk? Es wird erklärt, wie das Erklärvideo umgesetzt werden soll, wie viel Zeit den SuS gegeben wird, etc.	Computer mit Internet, A1.1 und A1.2
20 min.	Einstieg	Lehrerpräsentation, Plenum	Einführung in die Produktion eines Erklärvideos ggf. Programme erklären (PowerpointMix, Powtoon, ExplainEverything) Erklärvideos können auch mit Schere, Papier, Stift und Smartphone produziert werden.	
30 Min.	Erarbeitung	Partner- oder Gruppenarbeit (2 – 3 SuS)	Die SuS recherchieren selbstständig zum Thema Mobilfunk und erstellen daraus ein Storyboard	A1.2
60 - 80	Sicherung	Partner- oder Gruppenarbeit (2 – 3 SuS)	Die SuS produzieren ein Erklärvideo zum Thema Mobilfunk	Computer mit Internet, ggf. Schere, Papier, Stifte, Kamera/ Smartphone
20	Sicherung und Auswertung	Präsentation im Plenum	Die Ergebnisse werden im Plenum präsentiert und besprochen, es wird Feedback zu den einzelnen Filmen gegeben.	

6.2.2 Das Smartphone und ortsbezogene Daten

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
	Vorbereitung		Im Vorfeld der Unterrichtsstunde sollten das Programm Processing sowie alle notwendigen Daten für die Analyse der ortsbezogenen Daten den SuS zugänglich gemacht werden.	A1.3 L, A1.4
10 Min.	Einstieg	Plenum	<p>Wiederholung: kurze Zusammenfassung, der letzten Stunde; Hinweis durch L, dass Mobiltelefonieren ohne die Speicherung von Ortsdaten nicht funktionieren würde.</p> <p>Einstiegsfrage: Was kann man über eine Person allein durch ortsbezogene Daten erfahren? Ideen werden an der Tafel gesammelt</p>	



15 Min.	Einstieg	Lehrerpräsentation	L nennt den SuS die Ziele der nachfolgenden Aufgabenstellung und weist diese im Gebrauch von Processing und dem Auswertungsprogramm an.	Computer A1.3 SuS, A1.4
45 Min.	Erarbeitung	Partner-/Gruppenarbeit	Arbeitsauftrag: Den SuS werden die ortsbezogenen Daten von Malte Spitz übergeben. Sie sollen die Daten entsprechend auswerten, um gezielte Informationen zu erhalten (siehe Fragestellungen auf AB V2.2_SuS).	Computer A1.3 SuS, A1.4
20 Min.	Sicherung	Plenum	Vergleich der Ergebnisse, Diskussion: Was sagen die Daten tatsächlich aus? Wie vertrauenswürdig sind die Ergebnisse?	
25 Min.	Hausaufgabe		Die SuS sehen sich zur Vorbereitung auf die nachfolgende Stunde die vierte interaktive Episode „Der Spion in der Tasche“ der Serie DoNotTrack an (https://donottrack-doc.com/de/episodes/), Flipped Classroom-Methode ⁴	Computer mit Internet, A1.5

6.2.3 Das Smartphone als Datenschleuder

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 - 10 Min.	Einstieg	Plenum	Begrüßung der SuS, Die SuS haben zur Vorbereitung den Film „Der Spion in der Tasche“ von Donottrack gesehen. Klärung von Fragen zum Film	
10 min.	Einstieg	Einzelarbeit	Methode: Think – Pair – Share ⁵ Think: Die SuS reflektieren ihren eigenen Umgang mit dem Smartphone	A1.6
10 min.	Vertiefung	Partnerarbeit	Pair: Die SuS tauschen sich mit einem Partner einer Partnerin darüber aus	
15 – 20 min.	Vertiefung	Plenum	Share: offene Diskussion	Tafel oder Whiteboard

⁴ Unterrichtsmethode, bei der die Hausaufgaben und die Stoffvermittlung im klassischen Unterricht vertauscht werden. Die SuS bereiten sich inhaltlich auf ein Thema zu Hause vor. Im Unterricht kann direkt eingestiegen oder das Thema vertieft werden.

⁵ Denken – Austauschen – Teilen: Unterrichtsmethode, um sich in kurzen Schritten mit einer Frage auseinanderzusetzen. Erst jeder für sich, dann mit seinem Nachbarn und am Ende gemeinsam im Plenum.



		<ul style="list-style-type: none"> • Wer hat Interesse an unseren Daten? • Verlieren oder gewinnen wir etwas durch die Speicherung und Auswertung unserer Daten? • Welche Bedeutung hat der Datenschutz? • Was können wir selbst tun, um uns zu schützen? Gibt es Verhaltensregeln zum Schutz der Privatsphäre? Und kann man technische Vorkehrungen treffen, um die Privatsphäre zu schützen? <i>Ideen werden an der Tafel/ Whiteboard gesammelt</i> 	
--	--	---	--

6.2.4 Big Data

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
10 Min.	Einführung	Plenum	Begrüßung und Wiederholung – Was haben wir das letzte Mal gemacht? (Thema: Auswertung unserer Daten über das Handy), Übergang zur heutigen Stunde: Daten werden aber nicht nur über das Handy erhoben, sondern auch über das Tablet, FitnessTracker, Sensoren im Auto, etc. Film zum Einstieg: Big Data – ein Film der Landesmedienanstalt NRW https://www.youtube.com/watch?v=otWN5o1C2Bc	Computer, Internet
60 Min.	Erarbeitung	Gruppenarbeit	Arbeitsauftrag: Aufteilung in 5 Gruppen Recherche und Vertiefung zu einem Schwerpunktthema in Bezug auf Big Data: <ul style="list-style-type: none"> • Big Data und Werbung (einfach, Wiederholung) • Big Data im Verkehrswesen (einfach) • Big Data und Schule/ Bildung (mittel) • Big Data im Gesundheitswesen/ Medizin (mittel) • Big Data in der Politik/ Demokratie (schwer) • BigData und SpiegelMining (schwer) Die Arbeitsaufträge sind unterschiedlich anspruchsvoll, wodurch eine Differenzierung möglich ist.	Computer, Internet, A1.7, Flipchartpapier, Stifte (verschiedene Farben)

			Erstellung eines Wandplakats	
20 Min.	Sicherung	Plenum	Vorstellung der Ergebnisse, abschließende Diskussion über die Vor- und Nachteile von Big Data	Plakat

7 Einbettung in verschiedene Fächer und Themen

Im Rahmen dieses Moduls lassen sich verschiedenen Zusammenhänge zu Inhalten in anderen Fächern ausmachen. Durch die Bezüge zu unterschiedlichen Fächern kann das gesamte Modul oder auch nur einzelne Teile in verschiedenen Fächern durchgeführt werden.

Darüber hinaus haben wir im Folgenden aufgeführt, welche Kompetenzen aus den Bildungsstandards der Kultusministerkonferenz oder der einzelnen Rahmenlehrplänen der Länder durch das Modul A1 von IT2School unterstützt werden:

Physik

Schülerinnen und Schüler...

- beschreiben Phänomene und führen sie auf bekannte physikalische Zusammenhänge zurück.
- führen einfache Experimente nach Anleitung durch und werten sie aus.
- lernen Eigenschaften elektromagnetischer Wellen kennen.
- lernen Techniken zur Übertragung von Informationen kennen.

Politik/Sozialkunde

Schülerinnen und Schüler...

- erläutern die Bedeutung des Datenschutzes für Individuum und Gesellschaft.
- bewerten Chancen und Risiken unterschiedlicher Mediennutzung für den Einzelnen (u.a. Informationsbeschaffung, Sicherung der Privatsphäre, Datenschutz, Suchtproblematik).
- erörtern die gesellschaftspolitische Relevanz dieser Entwicklungen im Hinblick auf die informelle Selbstbestimmung, das Recht auf Privatsphäre und den Datenschutz.
- gewinnen einen Überblick über die Inhalte, Ziele und Anwendungsmöglichkeiten von Innovationen in der Informationstechnologie.
- vertreten die eigene Position im unterrichtlichen Zusammenhang und begründen diese – auch in Konfrontation mit anderen Sichtweisen – sachlich.
- nehmen fremde Positionen im Rahmen von Rollenspiel ein und bilden diese simulativ ab.

Informatik

Die Schülerinnen und Schüler ...

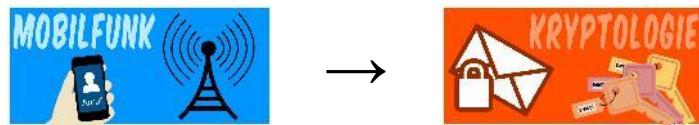
- unterscheiden beim Sammeln und Verarbeiten zwischen personenbezogenen und anderen Daten.
- kennen Grundzüge des Datenschutzes und des Rechts auf informationelle Selbstbestimmung.



8 Anschlussthemen

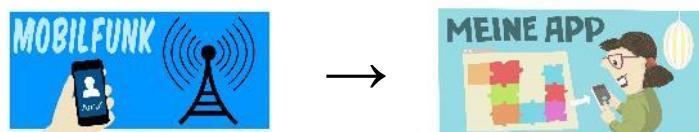
Als Anschlussthemen im Zusammenhang mit IT2School bietet sich folgendes Modul an:

Beispiel: Vom Mobilfunk zur Datensicherheit



Beispiel: IT spielend entdecken

Möchten Sie danach ein handlungsorientiertes Projekt umsetzen, können Sie zur App-Programmierung mit dem App Inventor.



9 Literatur und Links

- UPIAN et al: **Do Not Track**. Webserie über das Thema Tracking im Internet und Mobilfunk sowie Big Data. Online: <https://donottrack-doc.com/de/intro/>
- Sendung mit der Maus. **Sachgeschichte zur Funktionsweise eines Handys**. Online: https://www.youtube.com/watch?v=kno6nx_1ID8
- **PRISM-Rollenspiel zum Datenschutz**: <http://www.lehrerfreund.de/schule/1s/datenschutz-prism-spiel/4407>
- Bundeszentrale für politische Bildung - **Daten-Comix**: <http://www.bpb.de/mediathek/203501/data-comix>
- Papier zum Thema Big Data des Fraunhofer Instituts:
https://www.sit.fraunhofer.de/fileadmin/dokumente/studien_und_technical_reports/Big-Data-Studie2015_FraunhoferSIT.pdf

10 Arbeitsmaterialien

Nr.	Titel	Beschreibung
😊 A1.1	Wie produziert man ein Erklärvideo?	Verschiedene Möglichkeiten und Software zur Produktion von Erklärvideos werden vorgestellt
😊 A1.2	Wie funktioniert Mobilfunk?	Arbeitsblatt mit Rechercheauftrag zur Funktionsweise von Mobilfunk und zur Erstellung eines Erklärvideos
😊 A1.3 L	Datenanalyse mit Processing	Einführung in die Funktionsweise von Processing



A1.3 SuS	Auf den Spuren von Malte Spitz	Arbeitsblatt zur Auswertung von Daten von Malte Spitz
A1.4	Processing Zip-Archiv	Das Archiv enthält alle notwendigen Programme zur Datenanalyse der Mobilfunkdaten von Malte Spitz.
A1.5	Hausaufgabe	Link zum Video, Fragen zum Video
A1.6	Der Spion in unserer Tasche	Arbeitsauftrag zur Think-Pair-Share-Methode
A1.7	Big Data	Arbeitsauftrag zur Gruppenrecherche und Erstellung eines Wandplakats zum Thema Big Data

Legende

- Material für Schülerinnen und Schüler
- Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
- Zusatzmaterial

11 Glossar

Begriff	Erläuterung
Bewegungsdaten	s. Ortungsdaten
Big Data	Big Data bezeichnet Datenmengen, die zu groß oder zu komplex sind oder sich zu schnell ändern, um sie mit händischen und klassischen Methoden der Datenverarbeitung auszuwerten. Der traditionellere Begriff im Deutschen ist Massendaten. Der Begriff „Big Data“ unterliegt als Schlagwort derzeit einem kontinuierlichen Wandel; so wird mit Big Data ergänzend auch oft der Komplex der Technologien beschrieben, die zum Sammeln und Auswerten dieser Datenmengen verwendet werden. Die gesammelten Daten können aus nahezu allen Quellen stammen: angefangen bei jeglicher elektronischer Kommunikation, über von Behörden und Firmen gesammelte Daten, bis hin zu den Aufzeichnungen verschiedenster Überwachungssysteme. Big Data können so auch Bereiche abdecken, die bisher als privat galten.
Datenschutz	Je nach Betrachtungsweise wird Datenschutz verstanden als Schutz vor missbräuchlicher Datenverarbeitung, Schutz des Rechts auf informationelle Selbstbestimmung, Schutz des Persönlichkeitsrechts bei der Datenverarbeitung oder Schutz der Privatsphäre. Datenschutz steht für die Idee, dass jeder Mensch grundsätzlich selbst entscheiden kann, wem wann welche seiner persönlichen Daten zugänglich sein sollen. Vom Datenschutz abgrenzen ist die Datensicherheit, bei dem andere Ziele im Vordergrund stehen.



Datensicherheit (oder auch Informations-sicherheit)	Als Informationssicherheit (oder auch Datensicherheit) bezeichnet man Eigenschaften von informationsverarbeitenden und -lagernden (technischen oder nicht-technischen) Systemen, die die Schutzziele Vertraulichkeit, Verfügbarkeit und Integrität sicherstellen. Informationssicherheit dient dem Schutz vor Gefahren bzw. Bedrohungen, der Vermeidung von wirtschaftlichen Schäden und der Minimierung von Risiken. Beispiele für Maßnahmen zur Datensicherheit stellen Sicherheitskopien, Verschlüsselungssysteme, Nutzung von Berechtigungen und ähnliches.
Fitnessstracker (oder auch Activity Tracker)	Ein Activity Tracker (auch Fitness-Armband, Smart Band oder Fitness Tracker) ist ein Gerät oder eine Applikation zur Aufzeichnung und Versendung Fitness-relevanter Daten wie etwa Laufstrecken, Kalorienverbrauch und in manchen Fällen auch Herzschlagfrequenz oder Schlafqualität. Die Bezeichnung wird hauptsächlich für am Körper tragbare elektronische Überwachungsgeräte verwendet, welche (in vielen Fällen drahtlos) mit einem Computer oder Smartphone für die Datenerfassung über einen längeren Zeitraum synchronisiert werden. Abgesehen von diesen tragbaren Geräten, gibt es auch vergleichbare Applikationen für Smartphones und Facebook.
Funkzelle	Eine Funkzelle ist der Bereich, in dem das von einer Sendeeinrichtung eines Mobilfunknetzes gesendete Signal empfangen und fehlerfrei decodiert werden kann. ... Die Größe einer Funkzelle ist abhängig von meteorologischen und geografischen Gegebenheiten, Aufbauhöhe und Typ der verwendeten Antennen, der Sendeleistung und dem verwendeten Mobilfunkstandard.
GSM	s. Mobilfunk-Frequenzbereiche
LTE	s. Mobilfunk-Frequenzbereiche
Mobilfunkanlage	Eine Mobilfunkanlage versorgt eine bis mehrere Funkzellen, dies ist abhängig davon, wie viele Frequenzen genutzt werden. Innerhalb einer Funkzelle dient die Mobilfunkanlage als Sende- und Empfangsstation zum Austausch von Mobilfunkdaten. Das bedeutet, dass sowohl die Standortbestimmung aber auch die Datenübertragung während des Telefonierens über diese Anlage funktioniert.
Mobilfunk-Basisstation	s. Mobilfunkanlage
Mobilfunk-Frequenzbereiche	Im Mobilfunk werden für die verschiedenen Übertragungsstandards unterschiedliche Frequenzen verwendet. Der GSM-Standard (Global System for Mobile Communication) gehört zur so genannten 2. Generation (2G) und bewegt sich im Frequenzbereich von ca. 900 Megahertz. Der Standard UMTS (Universal Mobile Telecommunications System) liegt zwischen dem 199 und 2100 Megahertz und die neueste 4. Generation - LTE (Long Term Evolution) nutzen die Frequenzbereich um 800 Megahertz, 1,8 Gigahertz und 2,6 Gigahertz.
Open Data	Open Data bedeutet die freie Verfügbar- und Nutzbarkeit von – meist öffentlichen – Daten. Sie beruht auf der Annahme, dass vorteilhafte Entwicklungen unterstützt werden wie Open Government, wenn



	adressatengerecht und benutzerfreundlich aufbereitete Informationen öffentlich zugänglich gemacht werden und damit mehr Transparenz und Zusammenarbeit ermöglichen. Dazu verwenden die Ersteller Lizenzmodelle, die auf das Urheberrecht, Patente oder andere proprietäre Rechte weitgehend verzichten. Open Data ähnelt dabei zahlreichen anderen „Open“-Bewegungen, wie zum Beispiel Open Source, Open Content, Open Access, Open Education und ist eine Voraussetzung für Open Government.
Ortungsdaten	Als Ortungsdaten bezeichnen wir Daten, welche zur Ortung einer Person genutzt werden können. Hierbei kann es sich um ermittelte GPS-Positionen handeln, aber auch um Anmeldedaten des Smartphones/Handys an einer Funkzelle sowie andere Handydaten (durch Telefonate, SMS oder Nutzung von Apps).
personenbezogene Daten	Personenbezogene Daten bezeichnen Daten, welche Rückschlüsse auf eine Person zulassen. Hierbei kann es sich beispielsweise um die IP-Adresse eines Internetnutzers, Interessen, biologische Daten, die IMEI-Nummer oder Apps eines Smartphones handeln. Bei einer ausreichenden Datenmenge lassen sich so sehr genaue Aussagen über eine Person aufstellen.
Processing	Processing ist eine objektorientierte, stark typisierte Programmiersprache mit zugehöriger integrierter Entwicklungsumgebung. Die Programmiersprache ist für die Einsatzbereiche Grafik, Simulation und Animation spezialisiert. Processing wird in einem quelloffenen Projekt entwickelt, das am Massachusetts Institute of Technology in Boston von Ben Fry (Broad Institute) und Casey Reas (UCLA Design Media Arts) initiiert wurde. Processing hat den Charakter einer stark vereinfachten Version der Programmiersprache Java, ermöglicht Interaktionen und visuelle Elemente zu programmieren und richtet sich vorwiegend an Gestalter, Künstler und Programmieranfänger.
Tracking	Tracking ... umfasst alle Bearbeitungsschritte, die der gleichzeitigen Verfolgung von (bewegten) Objekten dienen. Davon unterschieden wird das Tracing, das eine zeitlich versetzte Verfolgung anhand von Aufzeichnungen betrifft, z. B. in der Programmierung als Ablaufverfolgung. Die Abgrenzung ist aber nicht einheitlich, so spricht man z. B. von einem GPS-Tracking unabhängig davon, ob die Verfolgung (Auswertung) gleichzeitig oder nachträglich erfolgt. Web Analytics (auch Clickstream-Analyse, Datenverkehrsanalyse, Traffic-Analyse, Web-Analyse, Web-Controlling, Webtracking) ist die Sammlung und Auswertung des Verhaltens von Besuchern auf Websites. Ein Analytic-Tool untersucht typischerweise, woher die Besucher kommen, welche Bereiche auf einer Internetseite aufgesucht werden und wie oft und wie lange welche Unterseiten und Kategorien angesehen werden. In Deutschland ist der Einsatz solcher Werkzeuge aus Datenschutzgründen umstritten.
UMTS	s. Mobilfunk-Frequenzbereiche

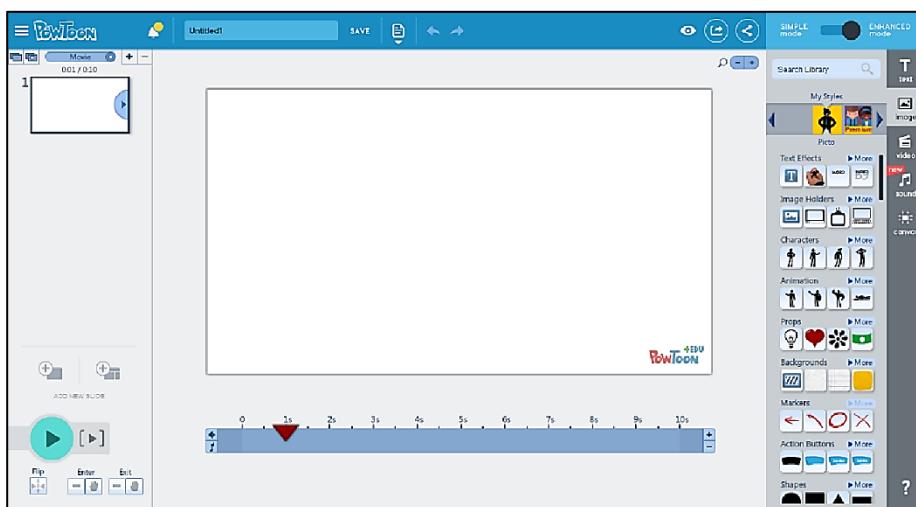


Wie produziert man ein Erklärvideo?

Die Produktion von Erklär- und Lernvideos ist heute, dank intuitiver Software und günstiger Technik viel einfacher geworden. Im Folgenden werden drei Möglichkeiten vorgestellt, wie Schülerinnen und Schüler selbst aktiv kleine Erklärvideos produzieren können.

Erklärvideos mit Powtoon

Für Schüler, die kein Office und damit auch keine Microsoft PowerPoint installiert haben, besteht die Möglichkeit online Erklärvideos mit Powtoon zu produzieren. Die Oberfläche ist übersichtlich und intuitiv zu bedienen. Ähnlich wie in PowerPoint sind auf der linken Seite die einzelnen Folien aufgelistet. Rechts findet man viele Vorlagen für Hintergründe, Bilder, Charaktere und Animationen. Auch eigene Bilder können eingefügt werden.



Links unten kann man für jedes eingefügte Objekt bestimmen, wie es auf der Folie erscheinen und auch wieder verschwinden soll. Durch einfache Pfeile kann die Richtung angegeben werden. Durch die Hände kann angegeben werden, ob die Objekte durch Hände reingeschoben werden sollen. Dadurch bekommen die Videos den typischen Charakter von Erklärvideos.



Links:

- Powtoon Tutorial (Video/englisch): <https://www.youtube.com/watch?v=bRqO5MasiFk>
- Powtoon im Unterricht: <http://carsten-reichert.de/2014/05/04/erklaervideos-im-unterricht-selbst-erstellen/>

Erklärvideo mit BYOD

Sollte in der Schule die Möglichkeit von PowerPoint oder einer guten Internetverbindung nicht gegeben sein, nutzen Sie sie die Ressourcen der Schülerinnen und Schüler. Ein Großteil der Schülerinnen und Schüler hat das eigenes Aufnahmegerät in der Hosentasche: Bring your own device (BYOD) fasst diese Möglichkeit unter einem Begriff zusammen.

Die Schülerinnen und Schüler können Figuren aufzeichnen oder ausdrucken und ausschneiden. Einer übernimmt die Kamera am Smartphone, ein weiterer Schüler legt und verschiebt die einzelnen Objekte. Ein dritter Schüler liest parallel den zuvor geschriebenen Text vor. Häufig werden mehrere Aufnahmen benötigt, bis ein zufriedenstellendes Ergebnis erreicht wird.

Für eine bessere Qualität bei der Tonaufnahme, kann das Video am Ende mit Hilfe von Windows Movie Maker noch einmal nachvertont werden. Auch ein Vor- und Abspann kann so eingefügt werden.

Links:

- Erklärvideo mit BYOD: <https://www.medienpaedagogik-praxis.de/2013/12/17/erklaervideos-mit-byod-produzieren/>
- Lernen durch Erklären – Projektbeispiel: <https://www.klett.de/alias/1067964>

Wie funktioniert Mobilfunk?

Mit Hilfe von Handys können wir heute jederzeit und an (fast) jedem Ort telefonieren, uns informieren, Videos anschauen oder auch Musik streamen.

Aber hast du dir schon einmal überlegt, wie eine Telefonverbindung zwischen zwei Handys zustande kommt? Oder welchen Weg eine SMS oder Messenger-Nachricht zurücklegt bis sie ankommt, selbst wenn du sie an Mitschülerinnen oder Mitschüler schickst, die direkt neben dir stehen?



Aufgaben

1. Teil euch in Kleingruppen auf (2-3 Schüler/ Schülerinnen pro Gruppe)
2. Recherchiert im Internet, wie Mobilfunkübertragung funktioniert.
3. Macht euch eine Skizze. Zeichnet den Weg während eines Telefonats von einem Handy zu einem anderen als Schaubild auf.
4. Produziert ein **Erklärvideo** zum Thema „Mobilfunk“
 - a. Erstellt zu Beginn ein Storyboard. In einem Storyboard werden die wichtigsten Szenen eines Films zeichnerisch festgehalten. Euer Schaubild kann dafür als Grundlage dienen.
 - b. Macht euch nun einen Arbeitsplan: Wer macht was? Welche Materialien benötigt ihr? Müssen noch lizenfreie Bilder im Internet dafür gesucht werden? Und bis wann sollen die einzelnen Schritte fertiggestellt sein?
 - c. Produziert euren Film (mit einem Videoschnittprogramm, wie z.B. Powtoon, oder mit dem Smartphone).
 - d. Präsentiert am Ende eure Ergebnisse.

Datenanalyse mit Processing

In diesem Lehrermaterial wird Ihnen gezeigt, wie Informationen aus den Mobilfunkdaten einer Person ermittelt werden können. Dafür werden das Programm *Processing*, ein für *Processing*



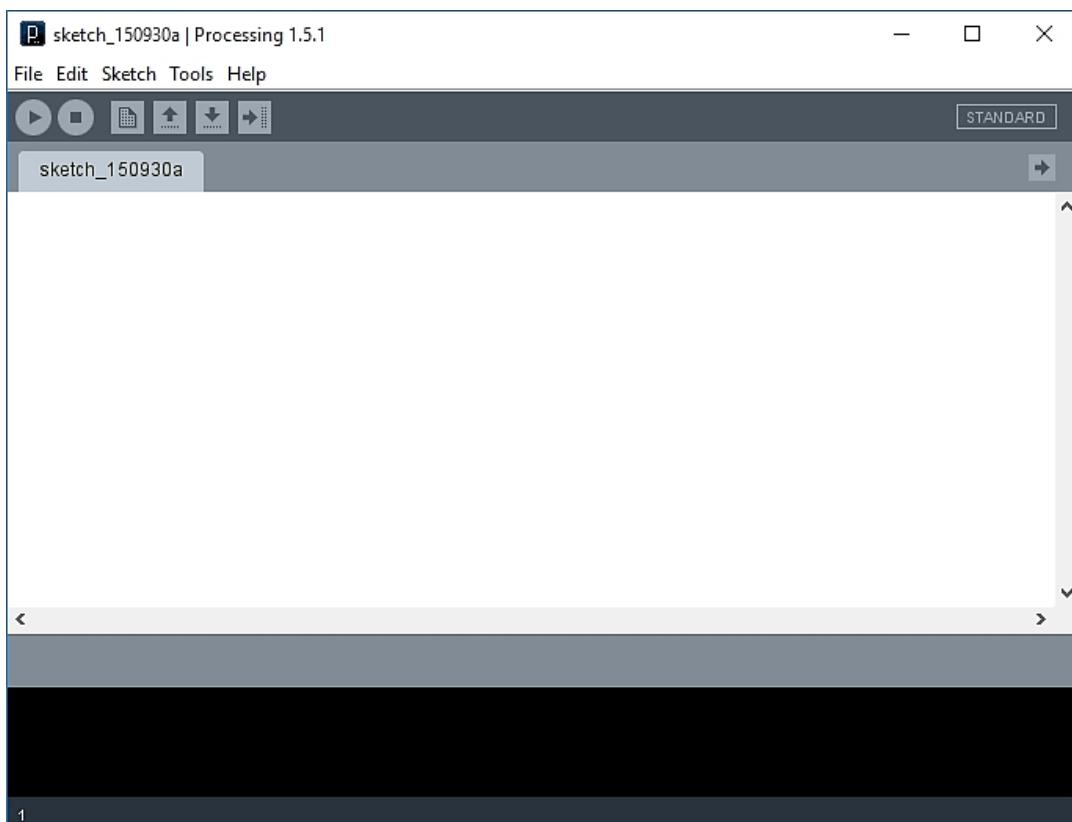
Abbildung: Malte Spitz

geschriebenes Script und die Mobilfunkdaten einer Person benötigt. Diese finden Sie alle im ZIP-Archiv V1.4. Malte Spitz, von dem die hier verwendeten Mobilfunkdaten stammen, ist ein Politiker und war von 2006 bis 2013 Beisitzer im Bundesvorstand der Partei Bündnis 90/Die Grünen. Seit Juni 2014 ist er Mitglied im Landesvorstand in Nordrhein-Westfalen. Die Daten stammen von T-Mobile und wurden nach einer Klage durch Malte Spitz herausgegeben. Er wollte damit

zeigen, welche Informationen aus den Daten (besonders im Zusammenhang mit der Vorratsdatenspeicherung) ermittelt werden können.¹

Processing

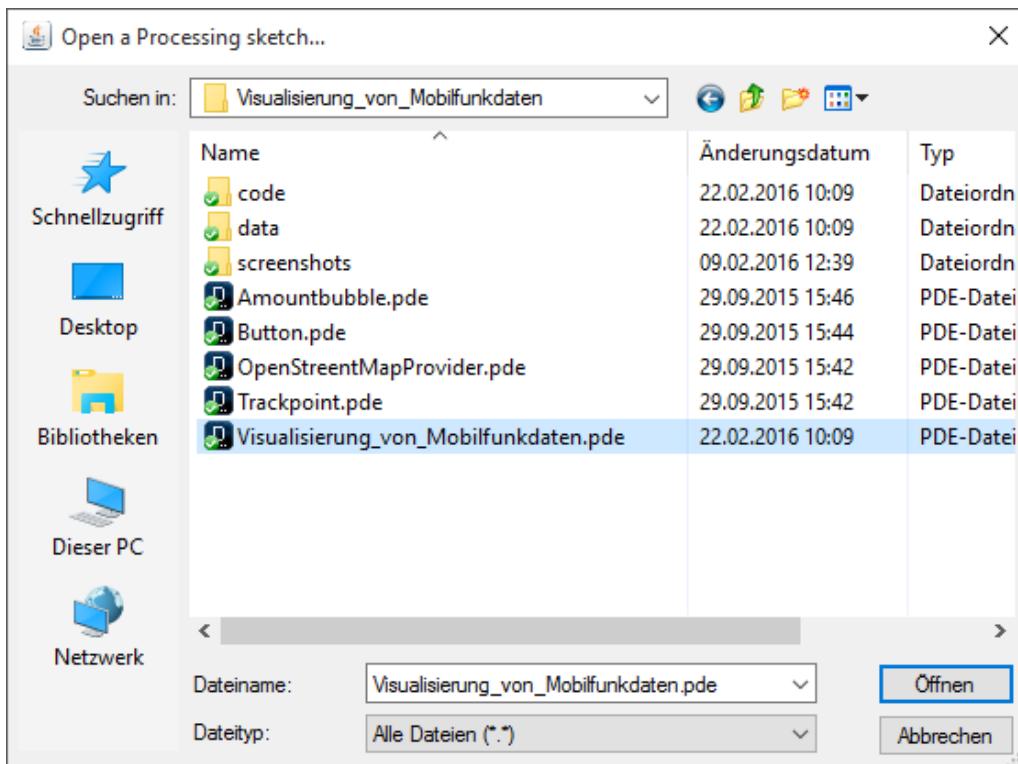
Wie bereits erwähnt benötigen Sie zur Visualisierung der Daten *Processing*, das sich in der Version 1.5 im ZIP-Archiv V1.4 befindet. Als Voraussetzung zur Ausführung von *Processing* ist JAVA nötig. Sobald Sie *Processing* geöffnet haben, sollten Sie folgendes Fenster sehen.



¹ Quelle: https://commons.wikimedia.org/wiki/File:20110808_MalteSpitz.png



Nun müssen Sie über das Register *File* und dem Unterpunkt *Open...* das Script zur Visualisierung der Mobilfunkdaten laden. Es befindet sich in der Datei: *Visualisierung_von_Mobilfunkdaten.pde*.



War dies erfolgreich, erscheint folgendes Fenster.

```

P Visualisierung_von_Mobilfunkdaten | Processing 1.5.1
File Edit Sketch Tools Help
STANDARD
Visualisierung_von_Mobilfunkdaten Amountbubble Button OpenStreetMapProvider Trackpoint
import java.text.*;
import java.util.*;

/*
*****
Visualisierung der Handydaten von Malte Spitz
*****
***** Bedienungsmöglichkeiten *****
Pfeiltasten    -> Bewegen der Karte
+/-/Mausrad   -> Ein/Auszoomen
Leertaste      -> Karte wird auf den letzten Bewegungspunkt zentriert
b              -> eigene Markierung auf Karte setzen
m              -> Menüs ein/ausschalten

```



Das Script zur Visualisierung der Mobilfunkdaten

Der in *Processing* angezeigte Quellcode gehört zu dem Skript, das die Daten von Malte Spitz aus einer Datei ausliest und anschließend auf einer Karte visualisiert. Wenn Sie den Quellcode anschauen, dann werden Ihnen ganz am Anfang einige Variablen auffallen, mit denen Sie das Visualisierungsprogramm anpassen können. Dazu gehören:

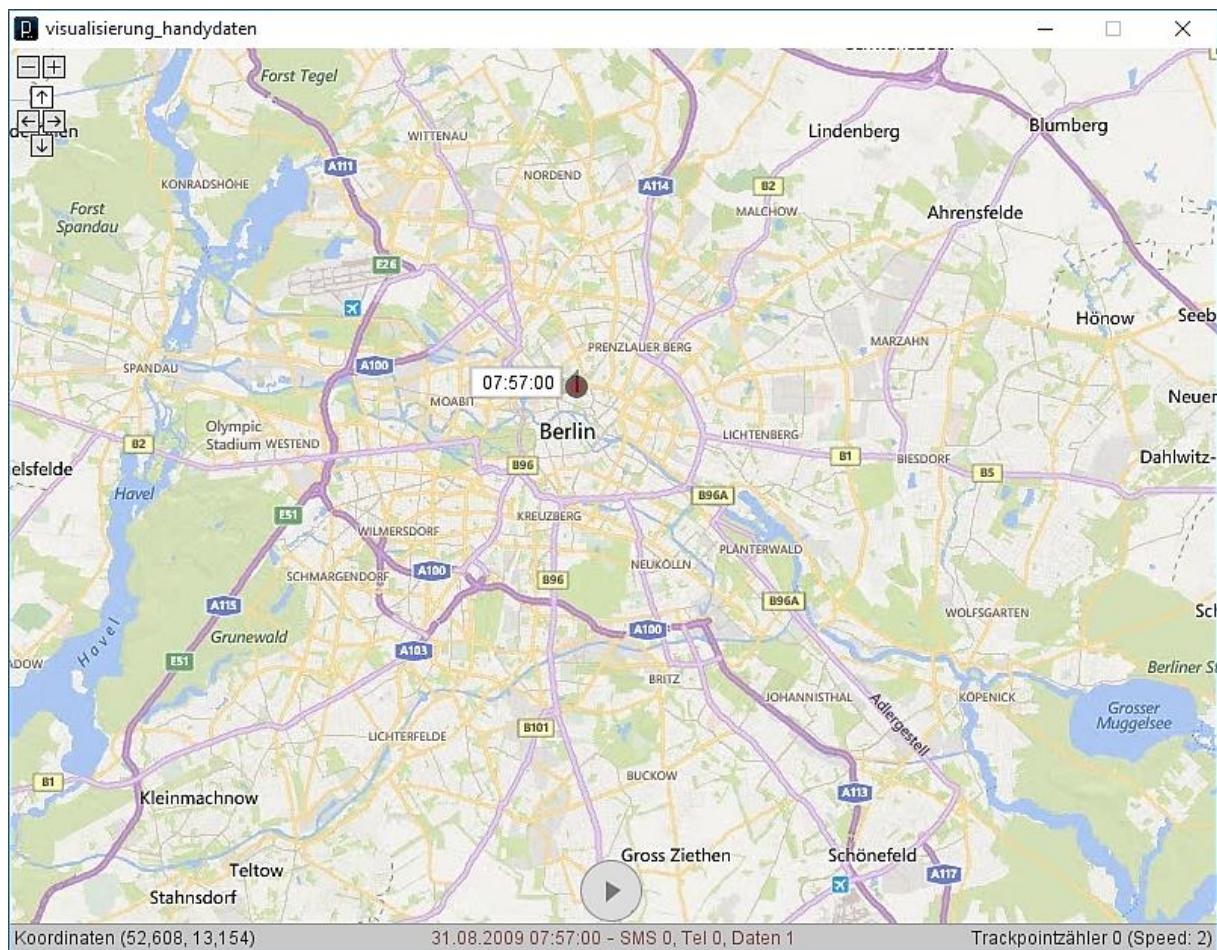
Variable	Beschreibung
filter_nach_datum	Wenn diese Variable den Wert <i>true</i> besitzt, dann werden die Daten von Malte Spitz vor der Visualisierung nach einem Anfangs- und Enddatum gefiltert. Wenn keine Filterung nach dem Datum vorgenommen werden soll, muss die Variable den Wert <i>false</i> besitzen.
datum_anfang	Wenn die Daten nach Datum gefiltert werden sollen, dann kann hier das Anfangsdatum (und die Uhrzeit) gesetzt werden. Möchte man z.B. die Daten ab dem 5. November 2009 starten lassen, dann muss der Wert der Variable “05.11.2009 00:00:00“ lauten (die Anführungsstriche müssen sein!).
datum_ende	Wenn die Daten nach Datum gefiltert werden sollen, dann kann hier das Enddatum (und die Uhrzeit) gesetzt werden. Möchte man z.B. die Daten am 28. November 2009 enden lassen, dann muss der Wert der Variable “28.11.2009 24:00:00“ lauten (die Anführungsstriche müssen sein!).
filter_nach_stunde	Wenn diese Variable den Wert <i>true</i> besitzt, dann werden die Daten von Malte Spitz vor der Visualisierung nach einer Anfangs- und Endstunde gefiltert. Wenn keine Filterung nach einer Stunde vorgenommen werden soll, muss die Variable den Wert <i>false</i> besitzen.
stunde_anfang	Wenn die Daten nach einer Stunde gefiltert werden sollen, dann kann hier die Anfangsstunde gesetzt werden. Möchte man z.B. die Daten ab der Stunde 13 beginnen lassen, dann muss der Wert der Variable 13 lauten (hier sind keine Anführungszeichen nötig!).
stunde_ende	Wenn die Daten nach einer Stunde gefiltert werden sollen, dann kann hier die Endstunde gesetzt werden. Möchte man z.B. die Daten ab Stunde 17 enden lassen, dann muss der Wert der Variable 17 lauten (hier sind keine Anführungszeichen nötig!).
filter_nach_wochentag	Wenn diese Variable den Wert <i>true</i> besitzt, dann werden die Daten von Malte Spitz vor der Visualisierung nach einem Wochentag gefiltert. Wenn keine Filterung nach einem Wochentag vorgenommen werden soll, muss die Variable den Wert <i>false</i> besitzen.
wochentag	Wenn die Daten nach einem Wochentag gefiltert werden sollen, dann kann hier der Wochentag gesetzt werden. Möchte man z.B. die Daten auf den Donnerstag eingrenzen, dann muss der Wert der Variable 5 lauten (hier sind keine Anführungszeichen

nötig!). Folgende Zahlenkodierung ist zu beachten: 1 für Sonntag, 2 für Montag, 3 für Dienstag, 4 für Mittwoch, 5 für Donnerstag, 6 für Freitag und 7 für Samstag.

Die obige Tabelle zeigt und erläutert dabei die wichtigsten Variablen. Im Quellcode lassen sich zwar noch weitere Variablen finden, jedoch sind diese von geringerer Bedeutung und werden auch an den entsprechenden Stellen erläutert.

Die Visualisierung der Daten

Möchte man sich nun die Daten visualisieren lassen, reicht es in Processing auf den Play-Button in der oberen linken Ecke zu klicken. Daraufhin sollte sich folgendes Fenster öffnen.



Drückt man nun in diesem Fenster auf den unteren Play-Button, so werden die Daten von Malte Spitz schrittweise angezeigt. Die Steuerung in diesem Fenster ist dabei ähnlich wie bei Google Maps oder anderen Diensten, aber es gibt auch einige weitere nützliche Funktionen, die den Schülerinnen und Schülern (und Ihnen) beim Lösen der späteren Aufgaben helfen. Folgende Tastenbelegungen gibt es für das obige Fenster:

Tasten	Beschreibung
Pfeiltasten	Mit den <i>Pfeiltasten</i> lässt sich die Karte in die entsprechende Richtung bewegen.

+/-/Mausrad	Um in die Karte rein und raus zu zoomen, kann das <i>Mausrad</i> oder die + bzw. – Taste verwendet werden.
Leertaste	Mit der <i>Leertaste</i> wird die Karte auf den letzten Bewegungspunkt zentriert.
b	Eigene Markierungen auf der Karte können mit der Taste <i>b</i> gemacht werden.
g	Mit der Taste <i>g</i> kann das Menü in dem Fenster ein und ausgeschaltet werden.
p/o	Die Tasten <i>p</i> und <i>o</i> erhöhen bzw. verringern die Geschwindigkeit bei der Visualisierung der Daten.
s	Um einen Screenshot der aktuellen Ansicht zu erstellen, kann die Taste <i>s</i> verwendet werden.
x	Die Visualisierung kann auch mit der Taste <i>x</i> gestartet und pausiert werden.
z	Da beim Zoomen manchmal die Schrift unleserlich wird, hilft es mit der Taste <i>z</i> die Karte neu zu skalieren.

Die folgende Grafik zeigt, wie Informationen auf der Karte abgelesen werden können.



Auf den Spuren von Malte Spitz



Abbildung: Malte Spitz¹

Auf diesem Arbeitsblatt wird dir gezeigt, wie viele Informationen man aus den Mobilfunkdaten einer Person herauslesen kann. Dafür benötigst du das Programm *Processing* und die Mobilfunkdaten einer Person. Das Programm sollte bereits für dich bereit liegen, ebenso wie die Mobilfunkdaten von Malte Spitz.

Malte Spitz ist ein deutscher Politiker. Die Daten stammen von T-Mobile und wurden nach einer Klage durch Malte Spitz herausgegeben. Er wollte damit zeigen, welche Informationen aus solchen Daten (besonders im Zusammenhang mit der Vorratsdatenspeicherung) ermittelt werden können.

Arbeiten mit Processing

Variable	Beschreibung
filter_nach_datum	Wenn diese Variable den Wert <i>true</i> besitzt, dann werden die Daten von Malte Spitz vor der Visualisierung nach einem Anfangs- und Enddatum gefiltert. Wenn keine Filterung nach dem Datum vorgenommen werden soll, muss die Variable den Wert <i>false</i> besitzen.
datum_anfang	Wenn die Daten nach Datum gefiltert werden sollen, dann kann hier das Anfangsdatum (und die Uhrzeit) gesetzt werden. Möchte man z.B. die Daten ab dem 5. November 2009 starten lassen, dann muss der Wert der Variable “05.11.2009 00:00:00“ lauten (die Anführungsstriche müssen sein!).
datum_ende	Wenn die Daten nach Datum gefiltert werden sollen, dann kann hier das Enddatum (und die Uhrzeit) gesetzt werden. Möchte man z.B. die Daten am 28. November 2009 enden lassen, dann muss der Wert der Variable “28.11.2009 24:00:00“ lauten (die Anführungsstriche müssen sein!).
filter_nach_stunde	Wenn diese Variable den Wert <i>true</i> besitzt, dann werden die Daten von Malte Spitz vor der Visualisierung nach einer Anfangs- und Endstunde gefiltert. Wenn keine Filterung nach einer Stunde vorgenommen werden soll, muss die Variable den Wert <i>false</i> besitzen.
stunde_anfang	Wenn die Daten nach einer Stunde gefiltert werden sollen, dann kann hier die Anfangsstunde gesetzt werden. Möchte man z.B. die Daten ab der Stunde 13 beginnen lassen, dann muss der

¹ Quelle: https://commons.wikimedia.org/wiki/File:20110808_MalteSpitz.png

Arbeitsmaterial A1.3

	Wert der Variable <code>13</code> lauten (hier sind keine Anführungszeichen nötig!).
<code>stunde_ende</code>	Wenn die Daten nach einer Stunde gefiltert werden sollen, dann kann hier die Endstunde gesetzt werden. Möchte man z.B. die Daten ab Stunde 17 enden lassen, dann muss der Wert der Variable <code>17</code> lauten (hier sind keine Anführungszeichen nötig!).
<code>filter_nach_wochentag</code>	Wenn diese Variable den Wert <code>true</code> besitzt, dann werden die Daten von Malte Spitz vor der Visualisierung nach einem Wochentag gefiltert. Wenn keine Filterung nach einem Wochentag vorgenommen werden soll, muss die Variable den Wert <code>false</code> besitzen.
<code>wochentag</code>	Wenn die Daten nach einem Wochentag gefiltert werden sollen, dann kann hier der Wochentag gesetzt werden. Möchte man z.B. die Daten auf den Donnerstag eingrenzen, dann muss der Wert der Variable <code>5</code> lauten (hier sind keine Anführungszeichen nötig!). Folgende Zahlenkodierung ist zu beachten: 1 für Sonntag, 2 für Montag, 3 für Dienstag, 4 für Mittwoch, 5 für Donnerstag, 6 für Freitag und 7 für Samstag.

Tasten	Beschreibung
Pfeiltasten	Mit den <i>Pfeiltasten</i> lässt sich die Karte in die entsprechende Richtung bewegen.
+/-/Mausrad	Um in die Karte rein und raus zu zoomen, kann das <i>Mausrad</i> oder die + bzw. – Taste verwendet werden.
Leertaste	Mit der <i>Leertaste</i> wird die Karte auf den letzten Bewegungspunkt zentriert.
b	Eigene Markierungen auf der Karte können mit der Taste <i>b</i> gemacht werden.
g	Mit der Taste <i>g</i> kann das Menü in dem Fenster ein und ausgeschaltet werden.
p/o	Die Tasten <i>p</i> und <i>o</i> erhöhen bzw. verringern die Geschwindigkeit bei der Visualisierung der Daten.
s	Um einen Screenshot der aktuellen Ansicht zu erstellen, kann die Taste <i>s</i> verwendet werden.
x	Die Visualisierung kann auch mit der Taste <i>x</i> gestartet und pausiert werden.
z	Da beim Zoomen manchmal die Schrift unleserlich wird, hilft es mit der Taste <i>z</i> die Karte neu zu skalieren.

Aufgaben

1. Finde heraus, wo Malte Spitz wohnt.

Tipp: Was machst du meistens nachts zwischen 3 und 4 Uhr?

2. Schau dir den Tweet von Malte Spitz an. Wo in Bayern war er und wie ist er dort hin- und wieder zurückgekommen?

nach zwei Tagen in Bayern wieder im Büro und x Texte fertig gemacht. Jetzt gehts zur VZ-Gruppe zum Gespräch zu Daten- und Verbraucherschutz

6:42 AM - 2 Sep 09 via web · Diesen Tweet integrieren

3. Du weißt nun schon einiges über Malte Spitz, aber schau dir auch folgenden Twitter-Auszug an. Welche Informationen lassen sich hieraus gewinnen?

Abschlusskundgebung läuft, es wird kühler, freu mich das über 500 Grüne aus dem gesamten Bundesgebiet dabei waren, Danke! #fsao9

7:06 PM - 12 Sep 09 via mobile web · Diesen Tweet integrieren

Tipp: Um was für eine Abschlusskundgebung handelt es sich? Was lässt sich im Internet über diese Kundgebung in Erfahrung bringen? Was verrät dies über die politische Gesinnung von Malte Spitz?

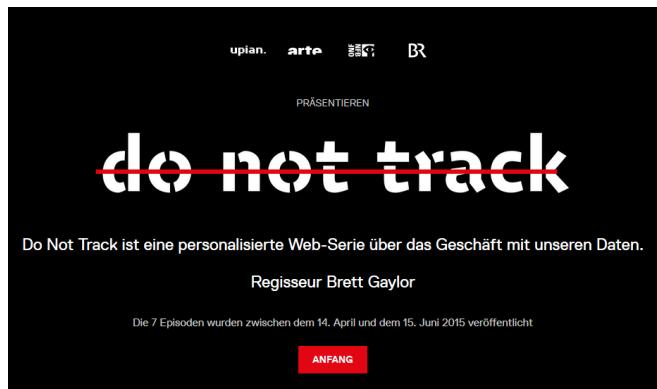
4. Wie oft war Malte Spitz im Bundestag?
5. Jemand behauptet, dass er am 02.09.2010 zwischen 20 und 21 Uhr mit ihm telefoniert hat. Stimmt das?
6. Irgendwann im Oktober 2009 war er an der Ostseeküste. Hat er dort Urlaub gemacht?
7. Wo wohnen Verwandte von Malte Spitz?
8. Wie aussagekräftig sind die hier ermittelten Informationen wirklich? Was sagen die Daten aus?

Hausaufgabe

Dein Smartphone zeichnet deine persönlichen Daten auf und kennt viele deiner Geheimnisse. Aber erzählt es auch jemandem davon? Und wenn ja, wem? In der nächsten Stunde wollen wir uns tiefer mit dieser Thematik auseinander setzen.

Bereite dich auf die nächste Stunde mit dem folgenden Video vor.

1. Gehe auf die Internetseite <https://donottrack-doc.com>



2. Klicke auf „Anfang“ und wähle die Episode S01E04 „**Der Spion in Ihrer Tasche**“ aus.
3. Sieh dir den gesamten Film an. Zwischendurch wird es interaktiv und du wirst gebeten, Fragen zu beantworten, erst dann läuft das Video weiter (Dauer des Films inkl. Fragen ca. 20 – 25 min.)
4. Während du den Film schaust, mache dir Notizen und beantworte folgende Fragen:
 - a. Worauf greift das Spiel „Angry Birds“ zu?
 - b. Welche Daten werden von unserem Smartphone, bzw. den Apps darauf gesammelt?
 - c. Was ist eine Herodes-Klausel?

Der Spion in unserer Tasche

Das Smartphone ist heute unser täglicher Begleiter. Der kleine Taschencomputer ist Musikanlage, Kalender, Spielekonsole, Nachschlagewerk, Musikinstrument u.v.m. Für die verschiedenen Einsatzmöglichkeiten benötigen wir die entsprechenden Applikationen (Apps), die wir kostenlos, aber auch kostenpflichtig installieren können.

Dabei greifen viele Apps auf Ortungsdaten, das Adressbuch oder andere Informationen zu. Einige Apps übertragen sogar Informationen, die sie für ihre Funktionalität gar nicht benötigen, wie beispielsweise Spiele-Apps und Taschenlampen-Apps.



Aufgabe

1. Think (Einzelarbeit)

- Überlege einmal, welche Datenspuren du täglich hinterlässt? Was kann man über dich erfahren? Wo hast du einen Account und welche Apps hast du auf deinem Smartphone installiert?
- Wer hat Interesse an unseren Daten?
- Welche Bedeutung hat für dich der Datenschutz?

2. Pair (Partnerarbeit)

- Tausche dich über deine vorangegangenen Gedanken mit deiner Nachbarin oder deinem Nachbarn aus. Wer gibt mehr über sich Preis?
- Wer hat Interesse an unseren Daten?
- Habt ihr unterschiedliche Meinungen zum Datenschutz? Wie geht ihr beide damit um?

3. Share (Diskussion im Plenum)

- Wer hat Interesse an unseren Daten?
- Welche Vor- und Nachteile ergeben sich für uns Nutzer durch das dauerhafte Sammeln unserer Daten?
- Welche Bedeutung hat der Datenschutz?
- Was können wir selbst tun, um unsere Daten zu schützen? Gibt es Verhaltensregeln zum Schutz der Privatsphäre? Kann man technische Vorkehrungen treffen, um unsere Daten zu schützen?

Big Data - Werbung

Big Data bezeichnet große Datenmengen, die zu groß und komplex sind, um sie mit herkömmlichen Methoden auszuwerten. Gesammelt werden Daten, die wir täglich im Internet hinterlassen, beispielsweise durch die Nutzung von sozialen Netzwerken, Apps (Anwendungen) auf unserem Smartphone oder wenn wir eine Suchmaschine verwenden.

Mittlerweile werden in vielen Bereichen digitale Daten gesammelt und ausgewertet. Recherchiert dazu im Internet und versucht mehr über die Einsatzmöglichkeiten von „Big Data“ zu erfahren.



Aufgabe

1. Erstellt ein Wandplakat zum Thema:

- Big Data und Werbung

2. Recherchiert dazu im Internet und/ oder in der Bibliothek. Verwendet die empfohlenen Links, recherchiert aber auch noch weitere Artikel im Internet.

Leitfrage: Wie wird Big Data für Werbung genutzt?

- Welche Daten werden dafür gesammelt und ausgewertet?
- Welche Vor- und Nachteile gibt es?
- Was bedeutet das Auswerten von Daten für uns?
- Wie ist eure Meinung dazu?

Linktipps:

- Marketing Im Web: <https://www.planet-schule.de/wissenspool/werbung/inhalt/hintergrund.html>
- E-Mail und die Detektive: <http://www.wdr5.de/sendungen/leonardo/emailunddiedetektive100.html>
- <https://irights.info/artikel/was-ist-und-wie-funktioniert-webtracking/23386>

3. Präsentiert euer Wandplakat zum Thema „Big Data und Werbung“ euren Mitschülerinnen und Mitschülern.

Big Data – Verkehrswesen und Mobilität

Big Data bezeichnet große Datenmengen, die zu groß und komplex sind, um sie mit herkömmlichen Methoden auszuwerten. Gesammelt werden Daten, die wir täglich im Internet hinterlassen, beispielsweise durch die Nutzung von sozialen Netzwerken, Apps (Anwendungen) auf unserem Smartphone oder wenn wir eine Suchmaschine verwenden.

Mittlerweile werden in vielen Bereichen digitale Daten gesammelt und ausgewertet. Recherchiert dazu im Internet und versucht mehr über die Einsatzmöglichkeiten von „Big Data“ zu erfahren.



Aufgabe

1. Erstellt ein Wandplakat zum Thema:

- Big Data und Verkehrswesen/ Mobilität

2. Recherchiert dazu im Internet und/ oder in der Bibliothek. Verwendet die empfohlenen Links, recherchiert aber auch noch weitere Artikel im Internet.

Leitfrage: Welche Bedeutung hat Big Data für den Verkehr?

- Welche Daten werden dafür gesammelt und ausgewertet?
- Welche Vor- und Nachteile gibt es?
- Was bedeutet das Auswerten von Daten für uns?
- Wie ist eure Meinung dazu?

Linktipps:

- Pkw-Maut: <https://heise.de/-2442285>
- Im Stau dank Edward Snowden: <https://rights.info/artikel/im-stau-dank-edward-snowden/23927>
- Kampf gegen den Stau: <http://www.welt.de/motor/article134215779/Dieser-Mann-kaempft-mit-Big-Data-gegen-den-Stau.html>

3. Präsentiert euer Wandplakat zum Thema „Big Data und Verkehrswesen/ Mobilität“ euren Mitschülerinnen und Mitschülern.

Big Data – Schule und Bildung

Big Data bezeichnet große Datenmengen, die zu groß und komplex sind, um sie mit herkömmlichen Methoden auszuwerten. Gesammelt werden Daten, die wir täglich im Internet hinterlassen, beispielsweise durch die Nutzung von sozialen Netzwerken, Apps (Anwendungen) auf unserem Smartphone oder wenn wir eine Suchmaschine verwenden.

Mittlerweile werden in vielen Bereichen digitale Daten gesammelt und ausgewertet. Recherchiert dazu im Internet und versucht mehr über die Einsatzmöglichkeiten von „Big Data“ zu erfahren.



Aufgabe

1. Erstellt ein Wandplakat zum Thema:

- Big Data und Schule/ Bildung

2. Recherchiert dazu im Internet und/ oder in der Bibliothek. Verwendet die empfohlenen Links, recherchiert aber auch noch weitere Artikel im Internet.

Leitfrage: Welche Bedeutung hat Big Data für das Lernen?

- Welche Daten werden dafür gesammelt und ausgewertet?
- Welche Vor- und Nachteile gibt es?
- Was bedeutet das Auswerten von Daten für uns?
- Wie ist eure Meinung dazu?

Linktipps:

- Aussieben kann nicht Aufgabe von Schule sein:
<http://www.sueddeutsche.de/bildung/neue-lernkonzepte-aussieben-kann-nicht-aufgabe-von-schule-sein-1.2157053>
- Big Data in der Bildung: <http://www.abida.de/sites/default/files/Education.pdf>
- Humboldt gegen Orwell: <http://www.zeit.de/2015/39/digitalisierung-bildung-internet-computer-lehrplan>

3. Präsentiert euer Wandpaket zum Thema „Big Data und Schule/ Bildung“ euren Mitschülerinnen und Mitschülern.

Big Data – Gesundheitswesen/ Medizin

Big Data bezeichnet große Datenmengen, die zu groß und komplex sind, um sie mit herkömmlichen Methoden auszuwerten. Gesammelt werden Daten, die wir täglich im Internet hinterlassen, beispielsweise durch die Nutzung von sozialen Netzwerken, Apps (Anwendungen) auf unserem Smartphone oder wenn wir eine Suchmaschine verwenden.

Mittlerweile werden in vielen Bereichen digitale Daten gesammelt und ausgewertet. Recherchiert dazu im Internet und versucht mehr über die Einsatzmöglichkeiten von „Big Data“ zu erfahren.



Aufgabe

1. Erstellt ein Wandplakat zum Thema:

- Big Data und Gesundheitswesen/ Medizin

2. Recherchiert dazu im Internet und/ oder in der Bibliothek. Verwendet die empfohlenen Links, recherchiert aber auch noch weitere Artikel im Internet.

Leitfrage: Welche Bedeutung hat Big Data im Bereich des Gesundheitswesens und der Medizin?

- Welche Daten werden dafür gesammelt und ausgewertet?
- Welche Vor- und Nachteile gibt es?
- Was bedeutet das Auswerten von Daten für uns?
- Wie ist eure Meinung dazu?

Linktipps:

- Sprechstunde beim Superrechner:
<http://www.spiegel.de/wissenschaft/medizin/big-data-wundermittel-auch-fuer-die-medizin-a-911333.html>
- Big Data in der Medizin: <https://irights.info/artikel/big-data-in-der-medizin/21054>
- An der Leine der Versicherer: <http://www.zeit.de/wirtschaft/2015-12/fitnesstracker-krankenkassen-gesundheitsapps>

3. Präsentiert euer Wandpaket zum Thema „Big Data und Gesundheitswesen/ Medizin“ euren Mitschülerinnen und Mitschülern.

Big Data – Politik/ Demokratie

Big Data bezeichnet große Datenmengen, die zu groß und komplex sind, um sie mit herkömmlichen Methoden auszuwerten. Gesammelt werden Daten, die wir täglich im Internet hinterlassen, beispielsweise durch die Nutzung von sozialen Netzwerken, Apps (Applikationen) auf unserem Smartphone oder wenn wir eine Suchmaschine verwenden.

Mittlerweile werden in vielen Bereichen digitale Daten gesammelt und ausgewertet. Recherchiert dazu im Internet und erfahrt mehr über die Einsatzmöglichkeiten von „Big Data“.



Aufgabe

1. Erstellt ein Wandplakat zum Thema:

- Big Data und Politik/ Gemeinwesen

2. Recherchiert dazu im Internet und/ oder in der Bibliothek.

Leitfrage: Welche Bedeutung hat Big Data für die Politik?

- Welche Daten werden dafür gesammelt und ausgewertet?
- Welche Vor- und Nachteile gibt es?
- Was bedeutet das Auswerten von Daten für uns?
- Wie ist eure Meinung dazu?

Linktipps:

- Brauchen wir noch Gesetze, wenn Rechner herrschen?:
<http://www.faz.net/aktuell/feuilleton/debatten/die-digital-debatte/big-data-und-politik-brauchen-wir-noch-gesetze-wenn-rechner-herrschen-13356219.html>
- BigData im US-Wahlkampf: <https://heise.de/-3328287>
- Big Data in Kommunen: <https://bigdatablog.de/2015/04/27/smart-community-so hilft big data in städten kommunen und gemeinden/>

3. Präsentiert euer Wandplakat zum Thema „Big Data und Politik/ Demokratie“ euren Mitschülerinnen und Mitschülern.

Big Data – SpiegelMining

Big Data bezeichnet große Datenmengen, die zu groß und komplex sind, um sie mit herkömmlichen Methoden auszuwerten. Gesammelt werden Daten, die wir täglich im Internet hinterlassen, beispielsweise durch die Nutzung von sozialen Netzwerken, Apps (Applikationen) auf unserem Smartphone oder wenn wir eine Suchmaschine verwenden.

Mittlerweile werden in vielen Bereichen digitale Daten gesammelt und ausgewertet. Recherchiert dazu im Internet und erfahrt mehr über die Einsatzmöglichkeiten von „Big Data“.



Aufgabe

1. Erstellt ein Wandplakat zum Thema:

- Big Data und SpiegelMining

2. Schau dir das folgende Video SpiegelMining – Reverse Engineering von Spiegel-Online (<https://youtu.be/-YpwsdRKt8Q>) an.

Leitfrage: Welche überraschenden Erkenntnisse konnten aus einem Nachrichtenportal abgeleitet werden?

- Welche Daten wurden benutzt?
- Welche Erkenntnisse konnten aus den Daten abgeleitet werden?
- Wie sicher sind die neuen Erkenntnisse?
- Welche Daten aus der Schule würdest du gern für einen anderen Zweck auswerten?

3. Präsentiert euer Wandpaket zum Thema „Big Data und SpiegelMining“ euren Mitschülerinnen und Mitschülern.

Musterlösungen

A1.4

1. Finde heraus, wo Malte Spitz wohnt.

Tipp: Was machst du meistens nachts zwischen 3 und 4 Uhr?

Lösung: 132-mal zu dieser Uhrzeit bei Position 52,532 / 13,404 Zwischen U Senefelderplatz, Rosenthaler Platz und Rosa-Luxemburg-Platz

2. Schau dir den Tweet von Malte Spitz an. Wo in Bayern war er und wie ist er dort hin- und wieder zurückgekommen?

Lösung: Er war in Erlangen und München. Was er in Erlangen und München gemacht hat, verrät die Webseite von seiner Arbeit und sein persönlicher Twitter-Account: In Erlangen referierte er zum Thema "Internetzensur", in München traf er sich im Biergarten "Das Kloster" mit Unbekannten.

Malte Spitz
@maltespitz

nach zwei Tagen in Bayern wieder im Büro und x Texte fertig gemacht. Jetzt gehts zur VZ-Gruppe zum Gespräch zu Daten- und Verbraucherschutz

6:42 AM - 2 Sep 09 via web · Diesen Tweet integrieren

Folgen

3. Du weißt nun schon einiges über Malte Spitz, aber schau dir auch folgenden Twitter-Auszug an. Welche Informationen lassen sich hieraus gewinnen?

Tipp: Um was für eine

Abschlusskundgebung handelt es sich? Was lässt sich im Internet über diese Kundgebung in Erfahrung bringen? Was verrät dies über die politische Gesinnung von Malte Spitz?

Malte Spitz
@maltespitz

Abschlusskundgebung läuft, es wird kühler, freu mich das über 500 Grüne aus dem gesamten Bundesgebiet dabei waren, Danke! #fsao9

7:06 PM - 12 Sep 09 via mobile web · Diesen Tweet integrieren

Folgen

4. Ist Malte Spitz ein fleißiger Politiker? Wie oft war er im Bundestag?

Lösung: Der Datumsfilter wird auf den Gesamtzeitraum und der Zeitfilter auf 10 bis 17 Uhr (angenommene Arbeitszeit) gesetzt. Anschließend lässt sich eine Markierung auf Bundestag setzen (b-Taste wenn Maus über dem Bundestag ist) und laufen lassen.

Absolut Häufigkeit: 1090

Relative Häufigkeit: $1090/13041 * 100 = 8,36\%$

Ob er ein fleißiger Politiker ist, ist daraus natürlich nicht ableitbar.

5. Jemand behauptet, dass er am 02.09.2010 zwischen 20 und 21 Uhr mit ihm telefoniert hat. Stimmt das?

Lösung: Der Datumsfilter wird auf 02.09.2010 00:00:00 bis 03.09.2010 00:00:00 und der Stundenfilter auf 20 bis 21 Uhr gesetzt.

Nein, im fraglichen Zeitraum hat Malte eine SMS empfangen, aber nicht telefoniert.

6. Irgendwann im Oktober 2009 war er an der Ostseeküste. Hat er dort Urlaub gemacht?

Lösung: Der Datumsfilter wird 01.10.2009 00:00:00 bis 1.11.2009 00:00:00 und der Stundenfilter auf 0 bis 24 Uhr gesetzt.

Er war in Rostock vom 23.10.2009 bis zum 25.10.2009. Recherche mit einer Suchmaschine ergibt, dass er dort nicht Urlaub gemacht hat, sondern auf der Bundesdelegiertenkonferenz war.

7. Wo wohnen Verwandte von Malte Spitz?

Lösung: Wenn man den Zeitraum auf Weihnachten einschränkt, dann lässt sich vermuten, dass Verwandte von ihm in Koblenz wohnen. Wenn man den gesamten Zeitraum betrachtet, sieht man auch, dass er häufig in Münster ist. Dort könnten also auch Verwandte wohnen.

8. Wie aussagekräftig sind die hier ermittelten Informationen wirklich?

Lösung: Wirklich sicher lassen sich einige Fragen natürlich nicht beantworten, wie zum Beispiel die Fragen 4, 5 oder 8. Dennoch sieht man, dass sich trotzdem viele (und sicher noch viele weitere) Informationen aus den Daten ermitteln lassen.

IT2School

Gemeinsam IT entdecken



Modul A2 – Kryptologie

Kryptologie

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

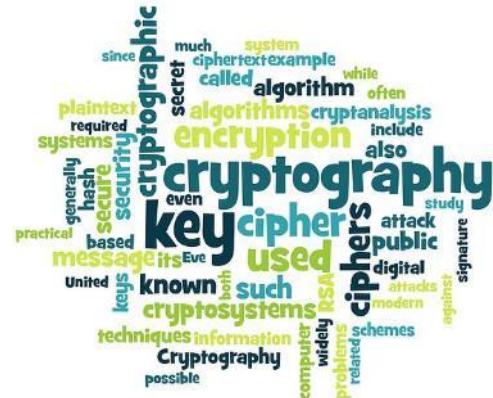
1	Kryptologie	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Die Rolle des Unternehmensvertreterin/des Unternehmensvertreters	4
5	Inhalte des Moduls.....	4
5.1	Schaubild zur Kryptologie	7
5.2	Moderne Kryptologie	7
6	Unterrichtliche Umsetzung.....	9
6.1	Datensicherheit im Alltag	9
6.2	Grober Unterrichtsplan (exemplarisch).....	10
6.3	Stundenverlaufsskizzen	11
7	Einbettung in verschiedene Fächer und Themen	14
8	Anschlussthemen.....	15
9	Literatur und Links	15
10	Arbeitsmaterialien	15
11	Glossar.....	16

1 Kryptologie

In diesem Modul befassen sich die Schülerinnen und Schüler mit dem Ver- und Entschlüssel von Informationen. Dabei werden Sicherheitsaspekte bei Kommunikationsvorgängen im Alltag aufgezeigt und verschiedene Verfahren zur Verschlüsselung aus der Vergangenheit bis zur heutigen Moderne vorgestellt.

In Anlehnung an ein „Text-Adventure“ erhalten die Schülerinnen und Schüler einen Überblick über verschiedene Verschlüsselungsverfahren. Sie müssen dabei kleinere Aufgaben lösen, während sich im Lauf der Zeit die Geschichte entfaltet.

Zum Abschluss können die Schülerinnen und Schüler die eigene Veröffentlichung von persönlichen Informationen sowie deren Kommunikation reflektieren und sich entsprechend absichern.



Lernfeld/Cluster:	Kommunikation erkunden
Zielgruppe/Klassenstufe:	4. bis 5. Klasse
	X 6. bis 7. Klasse
	X 8. bis 10. Klasse
	X 11. bis 12. Klasse
Geschätzter Zeitaufwand:	6 Einzelstunden
Lernziele:	<ul style="list-style-type: none">• Bedeutung von Verschlüsselung im Alltag und Arbeitswelt kennenlernen• Kryptographische und kryptoanalytische Verfahren kennenlernen• Ausgewählte Verfahren anwenden und „knacken“ können• Eigenen Umgang mit persönlichen Informationen reflektieren und anschließend schützen
Vorkenntnisse der Schülerinnen und Schüler:	Keine
Vorkenntnisse der/des Lehrenden:	Keine
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters:	Keine
Sonstige Voraussetzungen:	Keine

2 Warum gibt es das Modul?

Die Geschichte der Kryptologie ist eine alte Geschichte, die bis ins alte Ägypten und Griechenland zurückgeht. Obwohl sie in den Anfängen hauptsächlich für militärische Zwecke genutzt wurde, fand sie trotzdem den Weg in unseren Alltag.

Täglich haben wir es mit Verschlüsselung zu tun, bewusst oder unbewusst: Beim Schreiben einer WhatsApp-Nachricht, beim Online-Banking, beim Fernsehen von Bezahl-Sendern oder beim Bezahlen mit der EC-Karte. Dass in diesen Beispielen nicht jeder die übermittelten Daten einfach so lesen soll, ist sofort ersichtlich.

Für Unternehmen ist das Thema Verschlüsselung in Zeiten von Betriebsspionage und Cyberkriminalität von besonderer Bedeutung. Laut dem Bundeskriminalamt beläuft sich der jährliche Schaden in Deutschland auf ca. 50 Milliarden Euro, wobei von einer hohen Dunkelziffer ausgegangen wird. Sensible Inhalte, wie personenbezogene Daten sowie geistiges Eigentum müssen daher auch in Betrieben und Wirtschaftunternehmen geschützt werden.

Durch die NSA-Enthüllungen der letzten Jahre wurde das Thema der Verschlüsselung besonders präsent und immer mehr Menschen fangen an darüber nachzudenken, was mit ihren Daten passiert oder wie sie diese schützen können.

3 Ziele des Moduls

- Bedeutung von Verschlüsselung im Alltag und Arbeitswelt kennenlernen.
- Kryptographische und kryptoanalytische Verfahren kennenlernen.
- Ausgewählte Verfahren anwenden und „knacken“ können.
- Eigenen Umgang mit persönlichen Informationen reflektieren und anschließend schützen.

4 Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters

Im Modul *A2 – Kryptologie* hat die Unternehmensvertreterin/der Unternehmensvertreter mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

- Special-Guest: Kann über eigene Sicherheitsaspekte im Unternehmen berichten
- Kann beim Text-Adventure mitmachen

5 Inhalte des Moduls

Im Rahmen dieses Moduls geht es um *Geheime Kommunikation*, dabei werden die Begriffe *Kryptologie*, *Steganografie* sowie *Codierung* näher betrachtet und klar definiert. Als **Kryptologie** wird die Wissenschaft der Informationssicherheit bezeichnet. Dabei unterteilt sich diese in zwei Unterbereiche auf: Die **Kryptographie** beschäftigt sich mit der Verschlüsselung von Informationen und die **Kryptoanalyse** mit deren Entschlüsselung.

Die Geschichte der Kryptographie ist bereits sehr alt und spielte insbesondere für das Militär eine große Rolle. Die historischen Verschlüsselungsvarianten finden heute keine Verwendung mehr, aber anhand der Beispiele lassen sich die Grundlagen verdeutlichen.

Die in der Kryptographie gebräuchlichen Verfahren nennt man **Substitution** und **Transposition**.

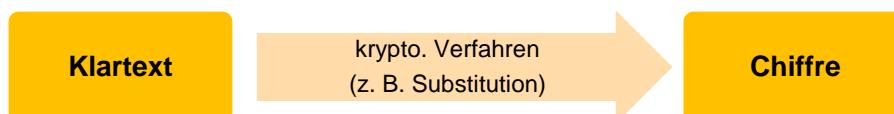
Bei der sogenannten **Transposition** werden Buchstaben oder Wörter im Klartext verschoben, so dass ein Sinnzusammenhang nicht direkt erkennbar wird bzw. der Klartext nicht so leicht lesbar ist. Als ein einfaches historisches Beispiel ist die griechische Skytale zu nennen. Mit Hilfe eines Holzstabes verschlüsselten die Griechen um ca. 400 v. Chr. ihre Nachrichten. Dabei wurde ein Papier- oder Lederstreifen um einen Holzstab gewickelt und dann darauf die Nachricht verfasst. Der Durchmesser des Skytale war der entscheidende Schlüssel zum entschlüsseln der Chiffre.



Skytale
<https://de.wikipedia.org/wiki/Skytale>

Ein weiteres Beispiel ist die Pallisaden- oder Gartenzaun-Chiffre. Die Buchstaben des Textes werden abwechselnd auf zwei Zeilen geschrieben, so dass der erste auf der oberen, der zweite auf der unteren, der dritte Buchstabe wieder auf der oberen Zeile steht und so weiter. Anschließend fügt man das Ganze zeilenweise wieder zusammen.

Bei der **Substitution** werden einzelne oder mehrere Buchstaben oder ganze Wörter innerhalb eines *Klartextes* (dem Text bzw. der Information vor Anwendung eines kryptographischen Verfahrens) vertauscht, wodurch aus diesem die *Chiffre* entsteht.



Eines der bekanntesten Beispiele ist die Caesar-Verschlüsselung, die schon in *Modul B1 – Vom Blinzen zum Verschlüsseln* behandelt wurde. Die Nachricht wird verschlüsselt indem jeder Buchstabe durch einen Buchstaben ersetzt wird, der um eine bestimmte Stelle im Alphabet versetzt wurde. Bei einer Verschiebung von 3 Stellen wird beispielsweise der Buchstabe A zu D usw.

Ein ähnliches Verfahren nutzten die Bewohner von Palestina in der Zeit von ca. 600-500 v. Chr. Bei der sogenannten *Altbash-Verschlüsselung* wurde der erste Buchstabe des Alphabets mit dem letzten Buchstaben, der Zweite mit dem Vorletzten usw. ersetzt (A=Z; B=Y,...)

Um ca. 755 n. Chr. herum gelang es dem arabischen Philosophen Abu-Yusuf Ya'qub ibn Ishaq al-Kindi als erster ein kryptoanalytisches Verfahren zum Knacken des Substitutionsverfahrens zu Beschreiben: **die Häufigkeitsanalyse**. Hierbei wird eine Chiffre danach untersucht, welche Symbole, Buchstaben oder Zahlen besonders häufig vorkommen. Anschließend wird geprüft, ob die am häufigsten vorkommenden Buchstaben bzw. Symbole aus der Chiffre mit den am häufigsten vorkommenden Buchstaben der jeweiligen Sprache bzw. Schrift ersetzt werden können (siehe auch Modul B1).

Anfang der zwanziger Jahre entwickelte Arthur Scherbius als erster eine Maschine zur Codierung, die so genannte *Enigma*. Sie bestand aus mehreren Chiffierungszylinern, die jeweils unterschiedliche Substitutionen innerhalb des Alphabets vornahm. Durch die hohe Anzahl an verschiedenen Walzen und Konfigurationsmöglichkeiten ergaben sich viele

Verschlüsselungsmöglichkeiten, weshalb sie zu damaliger Zeit als sehr sicher galt. Die Deutschen nutzten während des 2. Weltkrieges diese Form der Verschlüsselung. Im Jahr 1940 gelang es Marian Rejewski und Alan Turing die Enigma-Entschlüsselung zu knacken, wodurch sie den Ausgang des 2. Weltkrieges entscheidend beeinflussten.¹

Die **Codierung** wird verwendet, um Daten für eine entsprechende Anwendung in ein geeignetes Format zu bringen. Innerhalb der Basismodule haben wir schon Codes kennengelernt. Beispielsweise in Modul B1 den Morsecode oder im Modul B3 den Bar- und QR-Code. Mit Kenntnis des Codes ist die Entschlüsselung des Inhalts unproblematisch, hat man diesen nicht, hilft manchmal nur ein Zufallsfund, wie beispielsweise der Stein von Rosetta, der bei der Dechiffrierung der ägyptischen Hieroglyphen eine entscheidende Rolle spielte.

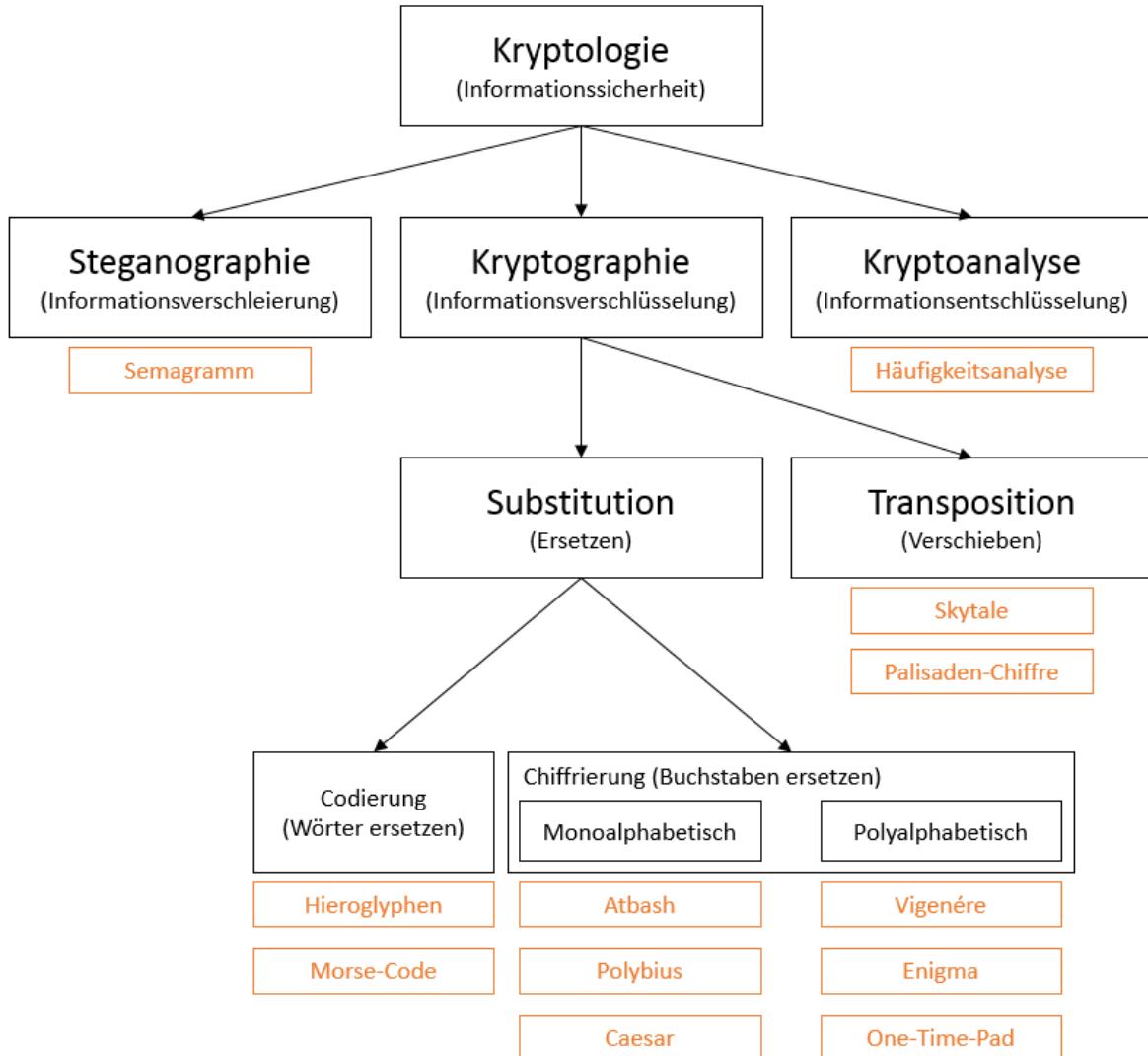
Bei der **Steganographie** steht die Verschleierung der Informationen im Vordergrund. Es gibt verschiedene Arten von steganographischen Verfahren. Historische und auch eher klassische Beispiele stellen dabei die unsichtbare Tinte (Zitronensaft), doppelte Böden in Paketen oder Briefumschlägen dar. Es gibt aber auch Verfahren, die mit der Sprache und der Codierung der Sprache arbeiten, wie beispielweise Semagramme. Hierbei handelt es sich um Bilder, in denen kleine Details versteckt sind, die allerdings die codierten Geheiminformationen darstellen. Betrachtet man zum Beispiel das folgende Bild, so fällt dem Betrachter die geheime Nachricht nicht direkt auf. Erst wenn man weiß, dass es sich bei den Grashalmen um Morsecode handelt, kann der Betrachter die Nachricht decodieren.



Auch im digitalen Bereich gibt Einsatzgebiete für Semagramme. Innerhalb einer MP3-Audio-Datei oder eines Bildes im JPG-Format lassen sich zusätzliche Bytes einfügen ohne dass sich die ursprüngliche Melodie oder das Bild ändert.

¹ Hierzu gibt es mehrere Verfilmungen wie zum Beispiel *The Imitation Game* oder die Arte Dokumentation *Wie ein Mathegenie Hitler knackte*.

5.1 Schaubild zur Kryptologie



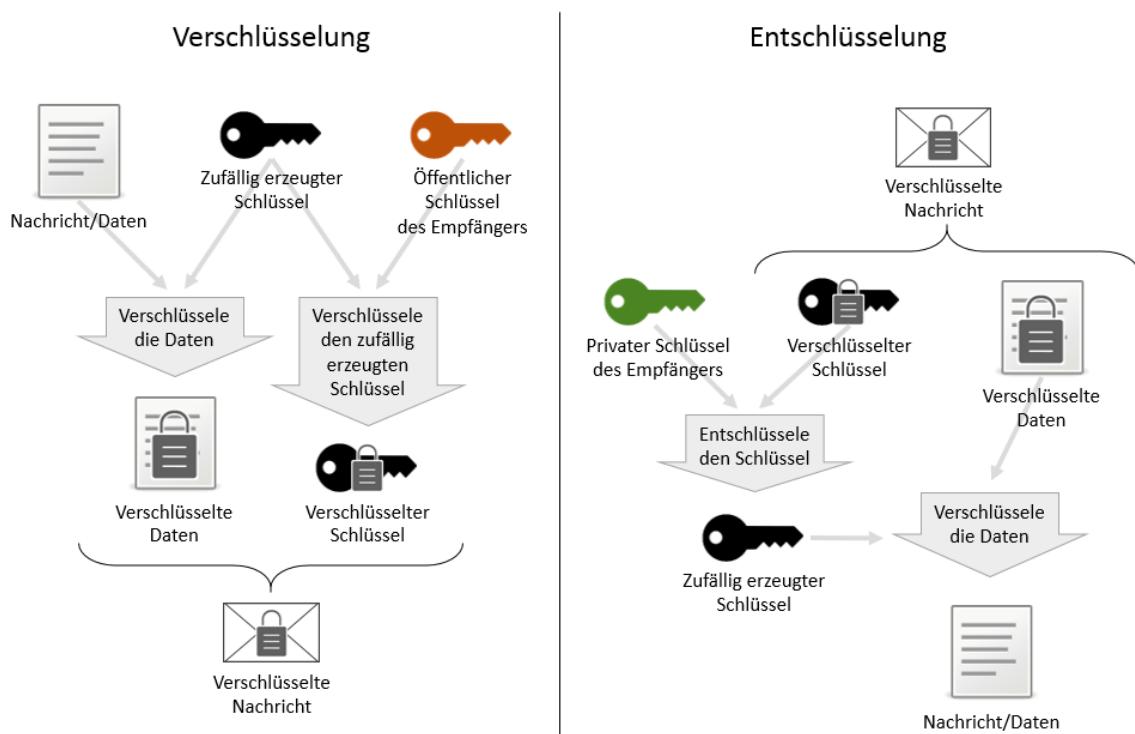
5.2 Moderne Kryptologie

In der heutigen Zeit gibt es immer noch viele verschiedene Arten von Verschlüsselung, bei der allerdings die Mathematik, im Gegensatz zu den historischen Beispielen, eine viel größere Rolle spielt. Aber auch Begriffe wie Public- und Private-Key oder symmetrische und asymmetrische Verschlüsselung finden in diesem Zusammenhang Verwendung.

Wichtig für das Verständnis ist zunächst einmal die Abgrenzung von symmetrischer und asymmetrischer Verschlüsselung. Hierbei werden Verfahren in zwei Gruppen eingeteilt, wobei die uns bekannten Verfahren alle zur Gruppe der symmetrischen Verschlüsselungsverfahren gehören. Bei der symmetrischen Verschlüsselung erfolgt die Ver- und Entschlüsselung mit dem selben Schlüssel bzw. anhand des selben Verfahrens (z.B. Verschiebung des Alphabets bei Caesar). Anders funktionieren die asymmetrischen Verschlüsselungsverfahren (wie z. B. RSA), bei der zur Ver- und Entschlüsselung unterschiedliche Schlüssel verwendet werden. Diese unterschiedlichen Schlüssel bei der asymmetrischen Verschlüsselung werden dann auch Public- und Private-Key genannt. Verständlicher wird die asymmetrische Verschlüsselung, wenn man sich als Beispiel die Funktionsweise von RSA genauer anschaut.

Die drei Buchstaben von RSA stehen für die Namen Rivest, Shamir und Adleman, die dieses Verfahren 1977 entwickelten. RSA arbeitet mit mathematischen Verfahren der Zahlentheorie und der modularen Arithmetik. Für ein tiefergehendes Verständnis ist daher Wissen über die Division mit Rest und die Kongruenzrelation notwendig. Beides soll hier jedoch nicht weiter vertieft werden, da die eigentliche Funktionsweise zum Verständnis der asymmetrischen Verschlüsselung nicht zwingend erforderlich ist. Entscheidend ist, dass für die Sicherheit des Verfahrens sehr große Primzahlen p und q verwendet werden. Es werden anschließend mit mathematischen Verfahren zwei weitere Zahlen e und d ermittelt. Das Paar (e, pq) bildet dann den öffentlichen Schlüssel und das Paar (d, pq) den privaten Schlüssel. Der öffentliche Schlüssel kann von einer Person veröffentlicht werden, damit andere Personen ihm verschlüsselte Nachrichten zulassen können. Eine Schritt-für-Schritt-Anleitung und Schaubild zu RSA befindet sich hierzu auch im Arbeitsmaterial V3.7.

Ein hybrides Verfahren der symmetrischen und asymmetrischen Verschlüsselung stellt PGP dar. PGP steht für Pretty Good Privacy und ist ein Programm zur Verschlüsselung und zum Unterschreiben, das von Phil Zimmermann entwickelt wurde und wird häufig bei der Verschlüsselung von E-Mails verwendet. Die erste Version, die 1991 entstand, benutzte RSA zum Verschlüsseln von Daten. In den späteren Versionen wird jedoch auf Elgamal als Verschlüsselungsverfahren zurückgegriffen. Um den Aufwand der Verschlüsselung möglichst gering zu halten, werden nicht die gesamten Daten mit einem asymmetrischen Verfahren verschlüsselt. Zunächst wird ein zufälliger Schlüssel erzeugt und mit einem asymmetrischen Verfahren verschlüsselt. Dieser zufällige Schlüssel dient zur symmetrischen Verschlüsselung der eigentlich Daten. Das folgende Schaubild zeigt dies etwas genauer:



6 Unterrichtliche Umsetzung

Diese Unterrichtseinheit ist als eine Art „Text-Adventure“ geplant, daher wird sich im Verlauf der Unterrichtsreihe eine Detektivgeschichte entfalten. Als Einstieg und damit sich die Schülerinnen und Schüler die kryptographischen Verfahren spielerisch selbst aneignen, erhalten diese in der ersten Unterrichtsstunde eine E-Mail mit einem verschlüsselten Text.

Damit ist die kryptographische Schnitzeljagd eröffnet und die Schülerinnen und Schüler müssen verschiedene Verschlüsselungen knacken und helfen letztlich, den entführten Pudel Rex zu finden. In wie weit die Schülerinnen und Schülern durch Arbeitsmaterialien unterstützt werden können, liegt bei der Lehrkraft. Vorgesehen ist jedoch, dass diese sich selbst über die kryptographischen Verfahren informieren und ihren Wissenszuwachs kontinuierlich mittels einer MindMap darstellen. Ziel ist es, die Beziehungen zu den einzelnen Begriffen und Konzepten herzustellen.

Im Anschluss an die Detektivgeschichte befassen sich die Schülerinnen und Schüler mit heutigen Verschlüsselungsverfahren und den Fragestellungen zur Datensicherheit. Hierfür empfehlen wir zum einen, sichere Kommunikationswege im Alltag (siehe Abschnitt 6.1) der Schülerinnen und Schüler aufzuzeigen sowie deren Nutzen und Bedeutung zu erarbeiten. Zum anderen sollte auch die praktische Anwendung von Verschlüsselungen, zum Beispiel in Form von Datei- oder E-Mailverschlüsselung thematisiert und durchgeführt werden. Hierfür bietet sich unter anderem das Video „Men in Grey“ (<https://criticalengineering.org/projects/men-in-grey>) an, das zeigt, wie eine Gruppe von Datenschutzaktivisten die Daten eines öffentlichen WLAN abgreifen und anschließend über einen Monitor und Sprachausgabe zurück an ihre Umwelt wiedergeben können.

6.1 Datensicherheit im Alltag

Nach dem Text-Adventure und der Auseinandersetzung mit den heutigen Verschlüsselungsverfahren, besprechen und reflektieren die Schülerinnen und Schüler aktuelle Beispiele für Sicherheitslücken und Absicherung von Diensten anhand von lebensweltorientierten Themen wie z.B. WhatsApp. Der beliebte Messenger bietet seit einiger Zeit eine Ende-zu-Ende Verschlüsselung an, d.h. das lediglich Sender und Empfänger in der Lage sind, die Nachrichten zu Entschlüsseln.

Die folgende Auswahl zeigt weitere Beispiele aus dem Alltag:

Das Onlinebanking ist ein typisches Beispiel für eine Verschlüsselung der Kommunikation zwischen Browser des Kunden und Webserver der Bank. Beim Aufruf und dem späteren Anmelden sowie den anschließenden Tätigkeiten (Überweisungen etc.) werden sensible Daten ausgetauscht. Damit dieser Austausch möglichst sicher ist, wird auf das Protokoll HTTPS zurückgegriffen. Anders als das normale HTTP Protokoll findet hier noch eine Verschlüsselung der Daten statt.

Eine weitere Kommunikation zwischen Kunde und Bank, deren Sicherheit sehr wichtig ist, ist die Übermittlung von Daten bei der Nutzung der EC-Karte. Zwar befinden sich auf der EC-Karte Informationen wie Kontonummer, Bankinstitut etc., aber der PIN muss sicher übermittelt werden. Hierzu befindet sich auf der EC-Karte ein eigenes kleines Cryptosystem, das die eingegebene PIN verschlüsselt und dann an den Zentralcomputer der Bank schickt.

Zwar hörte man in den vergangenen Jahren viel über abgehörte Handytelefonate (auch von wichtigen Personen des öffentlichen Lebens wie Frau Merkel), obwohl auch das Telefonnetz

für Mobiltelefone verschlüsselt ist. Jedoch ist die Verschlüsselung sehr schwach, so dass es zwar für Privatpersonen zu aufwändig, aber für Behörden wie die NSA oder dem BND ohne weiteres möglich ist.

6.2 Grober Unterrichtsplan (exemplarisch)

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Schülerinnen und Schüler durchlaufen die Detektivgeschichte rund um den Pudel Rex und lernen dabei verschiedene kryptographische Verfahren kennen.
Vertiefung	Schülerinnen und Schüler lernen moderne Verfahren kennen und wenden diese praktisch an. Dabei verschlüsseln sie E-Mailnachrichten und Dateien.
Abschluss	Schülerinnen und Schüler sehen den Film zu „Men in Grey“ und besprechen die Bedeutung von sicherer Kommunikation und Verschlüsselung im Alltag.

6.3 Stundenverlaufsskizzen

Abkürzungen/Legende

AB = Arbeitsblatt/Arbeitsblätter; L = Lehrkraft; MuM = Mitschülerinnen und Mitschüler; SuS = Schülerinnen und Schüler;
UV = Unternehmensvertreterin/Unternehmensvertreter

Detektivgeschichte

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
		Vorbereitung	L sendet den SuS die erste E-Mail der Detektivgeschichte. Hier ist es notwendig, dass die E-Mailadressen vorher vom L gesammelt werden. Sie sollten sich gegebenenfalls informieren, ob nicht bereits ein E-Mailverteiler existiert.	A2.1
10 Min.	Einsteig	Plenum	Begrüßung der SuS; Erklärung des neuen Themenkomplexes; Aufgabenstellung (MindMap und E-Mail) erklären; Flipchart/ Metaplanpapier verteilen Frage: <i>Wie kann man die Nachricht entschlüsseln? Gibt es einen Hinweis auf die Verschlüsselung?</i> MindMap: Als einzige Vorgabe wird der Begriff <i>Kryptologie</i> in die Mitte geschrieben.	
25 Min.	Erarbeitung	Einzel-/Partner-/Gruppenarbeit ²	Die SuS bearbeiten die erste Nachricht und Antworten auf die E-Mail.	
10 Min.	Sicherung	Plenum	Besprechung des Vorgehens und Lösung.	

² Welche Sozialform hier gewählt wird, liegt vollkommen bei der Lehrkraft, da alle drei möglich sind.



Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
Vorbereitung			L sendet den SuS die zweite E-Mail der Detektivgeschichte.	A2.1
5 Min.	Einsteig	A2.1	Begrüßung der SuS; Erklärung der Aufgabenstellung (MindMap und E-Mail)	
30 Min.	Erarbeitung	Einzel-/Partner-/Gruppenarbeit	Die SuS bearbeiten die zweite Nachricht und Antworten auf die E-Mail.	
10 Min.	Sicherung	Plenum	Besprechung des Vorgehens und Lösung.	

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
Vorbereitung			L sendet den SuS die dritte E-Mail der Detektivgeschichte.	A2.1, A2.2
5 Min.	Einsteig		Begrüßung der SuS; Erklärung der Aufgabenstellung (MindMap und E-Mail)	
30 Min.	Erarbeitung	Einzel-/Partner-/Gruppenarbeit	Die SuS bearbeiten die dritte Nachricht und Antworten auf die E-Mail.	
10 Min.	Sicherung	A2.1	Besprechung des Vorgehens und Lösung. L sendet den SuS die vierte E-Mail der Detektivgeschichte.	A2.1
Hausaufgabe			Die SuS können ihre MindMap nochmals überarbeiten.	

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
	Vorbereitung		L sendet den SuS die fünfte E-Mail der Detektivgeschichte.	A2.1, A2.3
5 Min.	Einsteig	Plenum	Begrüßung der SuS; Aufgabenstellung klären (MindMap und E-Mail)	



30 Min.	Erarbeitung	Einzel-/Partner-/Gruppenarbeit	Die SuS bearbeiten die fünfte und sechste Nachricht und Antworten auf die E-Mail. Während der Bearbeitung der fünften E-Mail sendet L den SuS die sechste E-Mail der Detektivgeschichte.	A2.1, A2.4
10 Min.	Sicherung	Plenum	Besprechung des Vorgehens und Lösung. L sendet den SuS die siebte E-Mail der Detektivgeschichte.	A2.1, A2.5
Hausaufgabe		Die SuS senden dem L (Sarah und Max) die fertige MindMap zu.		

Abschluss

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
30 Min.	Einstieg	Lehrervortrag	L bespricht mit den SuS den bisherigen Fortschritt und leitet den Abschluss dieses Themas ein. L sollte dabei den SuS die Funktionsweise von Public- und Private-Keys erklären und den Unterschied zu den bisherigen einfachen Verfahren (Caesar, Vigenére ...) aufzeigen.	A2.6
30 Min.	Vertiefung II	Einzel-/Partnerarbeit	Die SuS bearbeiten das Arbeitsmaterial V3.8 und verschlüsseln Dateien mittels verschiedener Softwarelösungen.	A2.7
30 Min.	Reflexion	Think-Pair-Share	L zeigt den SuS den Film zu Men in grey; Besprechung verschiedener Fragestellungen zur sicheren Kommunikation etc.	

7 Einbettung in verschiedene Fächer und Themen

Als Einbettung in ein speziellen Unterrichtsfach bietet sich in erster Linie die Informatik oder Technik an. Da gerade die Informatik nicht in allen Bundesländern fester Bestandteil der Schulbildung ist und einige Schulen kein Unterrichtsfach in Richtung der Informatik anbieten, würde es sich als Alternative anbieten, dieses Modul fächerübergreifend im Rahmen einer Projekt- bzw. Themenwoche einzubinden.

Die folgenden Kompetenzen finden sich entweder in den Bildungsstandards der Kultusministerkonferenz oder in den einzelnen Rahmenlehrplänen der Länder wieder:

Informatik/Technik

Die Schülerinnen und Schüler ...

- kennen ausgewählte Beispiele von Algorithmen/Verfahren zum Ver- und Entschlüsseln von Nachrichten sowie zum Knacken eben dieser.
- kennen die Bedeutung von kryptographischen Verfahren bzgl. sicherer Kommunikationen sowie die damit verbundene gesellschaftliche aber auch wirtschaftliche Bedeutung.
- reflektieren ihren eigenen Umgang mit sicheren und unsicheren Übertragungswegen und können sich absichern.
- kommunizieren fachgerecht über informative Sachverhalte.
- veranschaulichen kryptologische Sachverhalte. (Optional)
- implementieren kryptographische Verfahren mittels geeigneten Verfahren. (Ausblick)

Mathematik

Für eine Lehrkraft bedeutet die Einbindung dieses Moduls im Mathematikunterricht wahrscheinlich eher eine Ausrichtung bezüglich der Verfahren, weshalb hier gilt:

Die Schülerinnen und Schüler ...

- können geeignete heuristische Hilfsmittel, Strategien und Prinzipien zum Problemlösen auswählen und anwenden.
- können die Plausibilität der Ergebnisse überprüfen sowie das Finden von Lösungsideen und die Lösungswege reflektieren.
- können Überlegungen, Lösungswege bzw. Ergebnisse dokumentieren, verständlich darstellen und präsentieren, auch unter Nutzung geeigneter Medien.



8 Anschlussthemen

Als Anschlussthemen im Zusammenhang mit IT2School bieten sich folgende Module an:

Beispiel: Programmieren



Gerade hinsichtlich dem Modul B5 und A3 empfiehlt sich eine mögliche Vertiefung dieses Moduls bei dem die Schülerinnen und Schüler die erlernten Verfahren selbstständig in Form von Programmen in Scratch bzw. Python implementieren. Besonders die aufwändigeren Verfahren zum entschlüsseln von kryptographischen Verfahren bietet sich hier an.

9 Literatur und Links

- Simon Singh. **Codes – Die Kunst der Verschlüsselung**. 2001. Hanser Verlag. ISBN: 3-446-20169-6
- Didaktik der Informatik der Universität Wuppertal. **Spioncamp**. URL: <http://ddi.uni-wuppertal.de/material/spioncamp.html>
- Informatik Schule. **Kryptologie**. URL: <http://www.inf-schule.de/kommunikation/kryptologie>
- Informatik im Kontext: **E-Mail (nur) für Dich?** URL: <http://www.informatik-im-kontext.de/>
- **Morsecode**: <http://morsecode.scphillips.com/translator.html>

10 Arbeitsmaterialien

Nr.	Titel	Beschreibung
😊 A2.1	Detektivgeschichte	Enthält die E-Mailnachrichten, welche den Schülerinnen und Schülern gesendet werden sollen.
😊 A2.2	Anhang für dritte E-Mail	Bild der zu knackenden fleißnerschen Schablone
😊 A2.3	Anhang für fünfte E-Mail	ZIP-Archiv, das OpenPuff mit Anleitung, schlueselwoerter.png und den Kalender enthält.
😊 A2.4	Anhang für sechste E-Mail	Anleitung zur Verschlüsselung von E-Mails.
😊 A2.5	Anhang für siebte E-Mail	Bild von Pudel Rex.



A2.6	Präsentation zur asynchronen Verschlüsselung	Präsentation kann für den Lehrervortrag genutzt werden.
A2.7	Anleitung zu VeraCrypt	Anleitung zur Dateiverschlüsselung mit dem Programm VeraCrypt.
A2.8 bis A2.12	Zusatzmaterial	Dieses Material kann als Alternative zur Detektivgeschichte verwendet werden.

Legende

- Material für Schülerinnen und Schüler
- Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
- Zusatzmaterial

11 Glossar

Begriff	Erläuterung
Atbash-Verschlüsselung	<p>Ein monoalphabetisches Substitutionsverfahren, bei dem der erste Buchstabe des Alphabets mit dem letzten, der Zweite mit dem vorletzten usw. ausgetauscht wird:</p> <ul style="list-style-type: none"> • A wird mit Z verschlüsselt und umgekehrt • B wird mit Y verschlüsselt und umgekehrt • Usw.
Caesar-Verschlüsselung	<p>Ein monoalphabetisches Substitutionsverfahren, bei dem das Ursprungsalphabet um einen festen Wert verschoben wird. Bei einer Verschiebung von 3 bedeutet dies:</p> <ul style="list-style-type: none"> • A wird mit D verschlüsselt • B wird mit E verschlüsselt • Usw.
Chiffre	Bezeichnet einen verschlüsselten Text.
Chiffrieren	Bezeichnet das Verschlüsseln eines Klartextes.
Enigma	Bie der Enigma handelt es sich um eine sogenannte Rotor-Schlüsselmaschine. Mittels mehrerer Rotoren erfolgt die Generierung vieler Geheimalphabete, welche zur elektronischen Verschlüsselung von Nachrichten im zweiten Weltkrieg von den Deutschen genutzt wurde. Geknackt wurde die Verschlüsselung der Enigma von Alan Turing.
Geheimalphabet	Ein neues permutiertes Alphabet, welches auf Grundlage des Urpsrunsalphabetes des Klartextes durch Verschiebung oder



	Austausch von Buchstaben entsteht.
Hashen	Bezeichnet die Anwendung einer Hash-Funktion zur Generierung eines Hash-Wertes.
Hash-Funktion	Generiert aus einem Wert, welcher aus Zeichen, Zahlen, Dateien etc. bestehen kann, einen neuen Hash-Wert, welcher keine Zurückführung auf den eigentlichen Wert ermöglicht.
Hash-Wert	Ein mittels einer Hash-Funktion generierter Wert, welcher aus Zeichen und Zahlen bestehen kann.
Häufigkeitsanalyse	Ein Verfahren, dass beim Knacken von Entschlüsselungen verwendet wird und gerade bei einfachen monoalphabetischen Verschlüsselungen großen Erfolg verspricht.
Klartext	Ein unverschlüsselter Text.
Kryptoanalyse	Der Teil der Kryptologie, welcher sich mit der Entschlüsselung und Sicherheit befasst.
Kryptographie	Der Teil der Kryptologie, welcher sich mit der Verschlüsselung von Informationen befasst.
Kryptologie	Die Wissenschaft, welche sich mit der Sicherheit von Informationen befasst.
Monoalphabetisches Substitution	Bei Verfahren diesen Typs erfolgt die Substitution mittels eines Geheimalphabets. Beispiele hierfür sind die Atbash- oder Caesar-Verschlüsselung.
Polyalphabetisches Substitution	Bei Verfahren diesen Typs erfolgt die Substitution mittels mehrerer Geheimalphabete. Beispiel hierfür ist die Vigenère-Verschlüsselung.
Polybius-Verschlüsselung	Ein monoalphabetisches Substitutionsverfahren, bei dem Buchstaben eines Alphabets in einer Matrix angeordnet werden und anschließend durch deren Koordinaten ersetzt werden.
Semagramme	Dies ist eine Bezeichnung aus der Steganographie und bezeichnet einen unverfänglichen Text, Bild etc., in dem sich eine versteckte Geheimnachricht befindet.
Skytale-Verschlüsselung	Ein Transpositionsverfahren, bei dem ein Papier- oder Lederstreifen um einen Stab mit bestimmten Durchmesser (der sogenannte Skytale) gewickelt wird und auf welchem die Nachricht geschrieben wird.
Steganographie	Der Teil der Kryptologie, welcher sich mit der Verschleierung von Informationen befasst.
Substitution	Fasst alle Verfahren zusammen, bei dem Buchstaben, Symbole oder Wörter eines Klartextes durch andere ersetzt werden.
Transposition	Fasst alle Verfahren zusammen, bei dem Buchstaben, Symbole oder Wörter nicht ersetzt, sondern verschoben werden.
Vigenère-Verschlüsselung	Ein polyalphabetisches Substitutionsverfahren, welches der Caesar-Verschlüsselung ähnelt, aber bei dem mehrere durch ein



Schlüsselwort bestimmte Geheimalphabete verwendet werden.



Detektivgeschichte

Nachfolgend finden Sie alle E-Mails zur Durchführung der Detektivgeschichte. Diese können Sie einfach in ihr E-Mail-Programm kopieren, um Sie ihren Schülerinnen und Schülern zuzusenden. Zusätzlich finden Sie Vorlagen für die folgenden sieben E-Mails als .eml-Format im digitalen Archiv, die sich mit gängigen E-Mail-Programmen öffnen und so einfach an Ihre Schülerinnen und Schüler versenden lassen.

Erste E-Mail

Hi,

cool, dass du unserem Detektivklub beigetreten bist. Wir (Sarah und Max) sind die Hauptdetektive unseres Klubs. Da es aber viel zu tun gibt, freuen wir uns sehr über deine Unterstützung und hoffen, dass wir zusammen viele Fälle lösen können!

Apropos lösen können ... wir haben da heute eine seltsame Nachricht per E-Mail zugesandt bekommen. Auch wenn Max meinte, dass das nur Spam ist, könnte sich mehr dahinter verbergen. Kannst du herausfinden, was dahintersteckt? Unten findest du den Betreff und den Inhalt der E-Mail.

Solltest du es herausgefunden haben, dann antworte auf diese E-Mail und schicke uns den Inhalt der obigen Nachricht. Außerdem ist es wichtig für unseren Klub, dass neue Sachen dokumentiert werden. Erstelle also bitte eine MindMap zum Thema Kryptologie. Viel Erfolg!

Liebe Grüße
Sarah und Max

Betreff: Apzia ivxc Gdifn

Cvggj Yzozfodqz,

dxc cvwz zdizi bzczdhzi Vpaomvb aüm zpxc. Yjxc qjmczm hpnn dxc rdnnzi, jw dcm nj bpo nzdy, rdz dcm wzcvpkozo piy ydznz Qzmnxcgpznzgpib fivxfzi fjzio.

Gdzwz Bmpznnz
yzm Piwzfviioz

Unverschlüsselte Nachricht – Nicht an die Schülerinnen und Schüler herausgeben!!!

Betreff: Fuenf nach Links

Hallo Detektive,

ich habe einen geheimen Auftrag für euch. Doch vorher muss ich wissen, ob ihr so gut seid, wie ihr behauptet und diese Verschluesselung knacken könnt.

Liebe Gruesse
der Unbekannte



Zweite E-Mail

Hi,

super Einfall von dir. Wir hätten nicht gedacht, dass es sich um eine Verschlüsselung handelt. Du bist wirklich eine Bereicherung für unseren Klub! Wir haben den Unbekannten kontaktiert und eine neue E-Mail von ihm erhalten, die wir dir unten mal weitergeleitet haben.

Es scheint wirklich dringend zu sein, du solltest versuchen, die Verschlüsselung zu knacken. Dieses Mal scheinen die Buchstaben durch Zahlen ersetzt worden zu sein. Es handelt sich also wieder um eine Substitution. Sobald du die Nachricht entschlüsselt hast, schick sie uns zu. Viel Erfolg!

Liebe Grüße
Sarah und Max

PS: Denke bitte an deine MindMap. Der Begriff Substitution scheint wichtig zu sein und sollte in der MindMap auftauchen.

Betreff: Eine Entführung

Gut gemacht Detektive!

Es freut mich, dass ihr doch so gut seid, wie ihr behauptet. Ich habe einen sehr dringenden Entführungsfall, um den ihr euch kümmern müsst. Bevor ihr fragt, ja ich bin mir sicher, dass es sich um eine Entführung handelt!

Den einzigen Hinweis, den ich bezüglich der Täter habe, ist folgender Zettel (siehe Anhang), den ich gefunden habe. Er scheint eine verschlüsselte Botschaft zu enthalten (daher auch der Test mit der Caesar-Verschlüsselung).

Bitte helft mir, es ist dringend!

Anhang (Nachricht verschlüsselt mit Polybius 5*5 Matrix und i=j)

14 11 43 55 24 15 31 12 15 21 24 33 14 15 44 43 24 13 23 24 32 32 15 42 51 34 33 11 13 23
44 12 24 43 55 15 23 33 45 23 42 14 42 11 45 43 43 15 33 51 34 42 14 15 32 23 11 45 43 ...
43 13 23 31 11 22 15 14 11 33 33 55 45 45 33 14 43 13 23 33 11 35 35 14 24 15 43 15 33 42
15 53 !
33 24 32 32 11 33 43 13 23 31 24 15 15 33 14 14 15 33 12 45 43 24 33 42 24 13 23 44 45 33
22 24 33 33 15 33 43 44 11 14 44 .
14 11 33 33 33 24 32 32 14 15 33 13 34 32 35 45 44 15 42 23 24 33 44 15 33 31 24 33 25 43
24 32 24 33 44 15 42 33 15 44 13 11 21 15 ... 14 45 21 24 33 14 15 43 44 14 34 42 44 52 15
24 44 15 42 15 11 33 52 15 24 43 45 33 22 15 33 31 34 15 43 13 23 15 14 11 33 11 13 23 14
24 15 11 33 52 15 24 43 45 33 22 15 33 !

Unverschlüsselte Nachricht

Das Ziel befindet sich immer von acht bis zehn Uhr draussen vor dem Haus ...
Schlage dann zu und schnapp diesen Rex!
Nimm anschließend den Bus in Richtung Innenstadt.
Dann nimm den Computer hinten links im Internetcafe ... du findest dort weitere Anweisungen.
Loesche danach die Anweisungen!



Dritte E-Mail

Hi,

wunderbare Arbeit von dir! Ich (Ingo, ebenfalls ein neues Mitglied) kümmere mich nun um die Kommunikation mit dem Unbekannten, Sarah und Max untersuchen die Spuren (Bus und Internetcafé). Wer wohl dieser Rex ist? Sobald Sarah und Max etwas Neues wissen, werde ich dich informieren.

Der Unbekannte hat uns übrigens noch einen weiteren Zettel (siehe Anhang) zukommen lassen, den er gefunden hat. Auf der Rückseite des Zettels steht „Passwort“ geschrieben. Das ist bestimmt wichtig! Es scheint, als wenn es dieses Mal keine Substitution ist. Was für eine Art von Verschlüsselung ist es diesmal? Kannst du mal schauen, ob du auf das Passwort kommst und es mir dann zuschicken? Es könnte sein, dass es Sarah und Max im Internetcafé brauchen.

Liebe Grüße
Ingo

PS: Ich bin froh, dass du alles in der MindMap festhältst, es werden langsam viele Informationen. Diese neue Art von Verschlüsselung müsste dort auch eingefügt werden.

Anhang (fleißnersche Schablone)

C

S	R	T	C
E	I	C	H
T	T	H	A
G	B	R	A

Schablone

Lösung – Nicht an die Schülerinnen und Schüler herausgeben!!!

SCHACHBRETTARTIG



Vierte E-Mail

Hi,

deine E-Mail kam genau zum richtigen Zeitpunkt!

Wir (Sarah und Max) konnten uns mit diesem Passwort als Benutzer am Computer anmelden. Nun versucht Lisa (auch ein Mitglied unseres Klubs), die privaten Dateien des Accounts auszuwerten.

Für dich heißt es momentan also Pause machen. Hast du dir auch verdient ;-). Sobald wir mit der Auswertung der Dateien fertig sind, werden wir dich informieren. Wir freuen uns schon auf deine MindMap! Bitte nutze die Zeit um sie vielleicht noch zu überarbeiten.

Übrigens wissen wir nun wer Rex ist! Das ist der Pudel des Unbekannten. Der Besitzer des Internetcafés hat zwar den Hund bemerkt, aber nicht den Entführer. Sehr schade ...

Liebe Grüße
Sarah und Max



Fünfte E-Mail

Hi,

die Auswertung der Dateien war sehr aufschlussreich. Wir haben einen Kalender gefunden, wobei die Einträge wieder verschlüsselt sind. Diesmal haben wir direkt versucht, die Verschlüsselung mit Caesar zu knacken, aber das hat nicht funktioniert. Das heißt, dass du mal wieder dran bist :-).

Außerdem haben wir noch eine Datei *schluesselwoerter.png* gefunden. Wir vermuten, dass die Entführer in diesem Bild eine weitere Datei versteckt haben. Zumindest meinte Lisa, dass das möglich ist. Sie hat aber gerade keine Zeit dafür. Daher brauchen wir auch hier deine Hilfe. Sie gab uns den Tipp das Programm OpenPuff zu verwenden (Anleitung ist angehängt). Vielleicht kannst du das notwendige Passwort aus dem Bild entnehmen.

Warte übrigens mit deiner Antwort ab. Lisa versucht gerade unsere E-Mails abzusichern. Du bekommst später noch eine weitere E-Mail.

Viel Erfolg!

Liebe Grüße

Sarah und Max

PS: Wie steht es um die MindMap? Füge auch die neue Art von Verfahren hinzu. Es sollte auch klar werden, wie sie sich zwei Verfahren der gleichen Art voneinander unterscheiden können.

Lösung – Nicht an die Schülerinnen und Schüler herausgeben!!!

Codewort für openpuff lautet FEHLERFREI (wenn man sich *schluesselwoerter.png* anschaut kommt man recht schnell drauf). Damit können aus dem Bild die folgenden Codewörter extrahiert werden:

SPORT
FREUNDE
AUFTRAG
NOTIZEN

Mit diesen Codewörtern lassen sich die Kalendereinträge wie folgt mittels des Vigenere-Verfahrens entschlüsseln (Tipp: <https://einklich.net/etc/vigenere.htm> ist eine gute Hilfe, sobald man das Vigenere-Verfahren bereits verstanden hat und Zeit sparen möchte):

02.09. 18:00	Xxhexkhgknavc	Fitnessstudio	(Sport)
05.09. 19:00	Ryc rbnofjg	Rex abholen	(Auftrag)
08.09. 21:00	Ryc zm Cogxj hec Erry ugzbkkn	Rex im Cubes bei Lars abgeben	(Auftrag)
09.09. 18:00	Xxhexkhgknavc	Fitnessstudio	(Sport)
13.09. 16:00	Tywfzn byn uex Vfgb	Termin bei der Bank	(Auftrag)
16.09. 18:00	Xxhexkhgknavc	Fitnessstudio	(Sport)
17.09. 20:00	Lvfœvxfxjyvhv msh Mes	Geburstagfeier von Jan	(Freunde)
23.09. 18:00	Xxhexkhgknavc	Fitnessstudio	(Sport)
27.09. 16:00	Yiizshr dmñ Oezie	Treffen mit Laura	(Freunde)
30.09. 18:00	Xxhexkhgknavc	Fitnessstudio	(Sport)
Notizen:	Uigldjhghxz onhtxv! Piumr -> Uitqtnfmwmwgeolad 3	Hundefutter kaufen. Cubes -> Baumgartenstrasse 3	(Notizen) (Notizen)
Geburtstage:	17. Orr 29. Mrrhrv	Jan Hannes	(Freunde) (Freunde)



Sechste E-Mail

Hi,

es ist besser, wenn wir ab sofort unsere E-Mails verschlüsseln. Wir verwenden dazu nun PGP. Habe dir eine Anleitung mit Erklärung sowie unseren Public-Key angehängt. Solltest du uns schreiben, dann verschlüssle deine E-Mail mit diesem Schlüssel.

Konntest du den Kalender entschlüsseln? Wenn ja, dann schick uns bitte den entschlüsselten Kalender. Sarah und Max koordinieren das weitere Vorgehen.

Liebe Grüße
Lisa



Siebte E-Mail

Hi,

wir konnten die Übergabe von Rex mit Hilfe der Polizei vereiteln! Die Entführer haben Rex entführt, weil dieser schon mehrere Schönheitswettbewerbe gewonnen hat. Nun befindet sich Rex aber wieder bei seinem Herrchen (unserem Biolehrer Herrn Meyer). Wer hätte gedacht, dass Herr Meyer der Unbekannte ist...

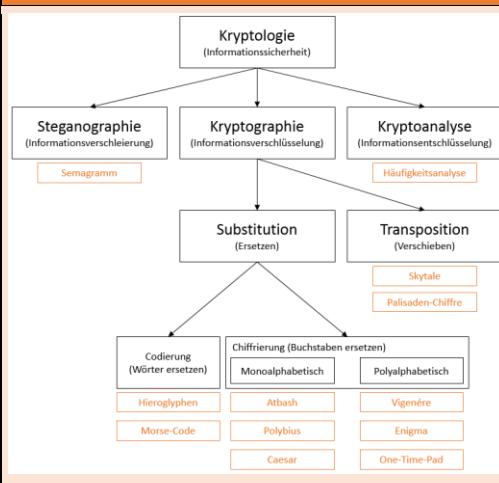
Das war sehr gute Arbeit von dir! Hier übrigens noch ein Bild vom nun wieder glücklichen Rex.



Jetzt bräuchten wir nur noch deine MindMap um den Fall für die Polizei auch vollständig dokumentieren zu können. Danke dafür!

Liebe Grüße
Sarah und Max

Mögliche Mind-Map (siehe Modulbeschreibung)



E-Mails verschlüsseln

Die folgende Anleitung zeigt dir, wie du deine E-Mails verschlüsseln kannst. Hierfür brauchst du natürlich eine E-Mailadresse. Sollte es der Fall sein, dass du bisher keine hast, oder das sicherhaltshalber noch nicht mit deiner eigenen E-Mailadresse testen möchtest, kannst du eine von deiner Lehrerin bzw. deinem Lehrer bereitgestellte E-Mailadresse benutzen.

Schritt 1 – Installation von Thunderbird

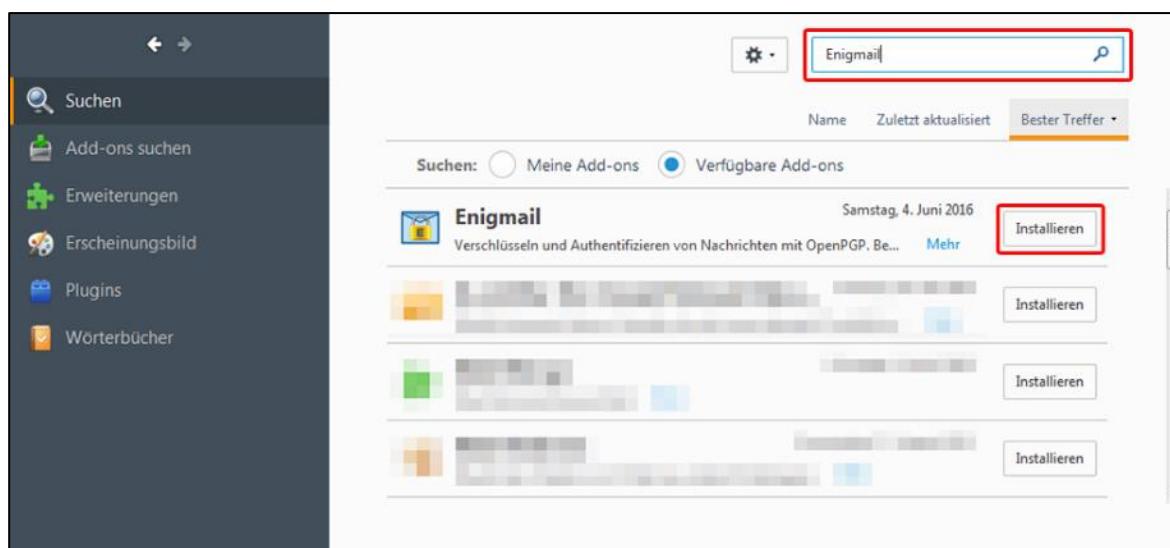
Als E-Mailprogramm verwenden wir Thunderbird. Sollte das Programm noch nicht auf dem Computer installiert sein, dann findest du unter folgender URL die Installationsdatei:

<https://www.mozilla.org/de/thunderbird/>

Anschließend solltest du das Programm installieren und einen E-Mailaccount einrichten.

Schritt 2 – Installation von Enigmail

Standardmäßig bietet Thunderbird nicht die Möglichkeit E-Mails zu verschlüsseln. Daher benötigen wir das Add-on *Enigmail*. Klicke dazu in der Menüleiste auf den Reiter *Extras* und wähle *Add-ons*. (Sollte die Menüleiste nicht sichtbar sein, musst du die *ALT*-Taste drücken.) Anschließend gibst du in das Suchfeld *Enigmail* ein und kannst das Add-on *installieren*.



Schritt 3 – GnuPG installieren

Nachdem das Add-on installiert wurde, kannst du Thunderbird erst einmal schließen (es sollte ein Neustart des Programms notwendig sein). Anschließend benötigen wir noch PGP bzw. die Implementierung GnuPG. Auf der offiziellen Webseite zu GnuPG solltest du ein entsprechendes Programm samt Download für dein Betriebssystem finden (ganz unten auf der Seite unter *GnuPG binary releases*):

<https://www.gnupg.org/download/index.html>

Für Windows kannst du auf Gpg4win klicken und die aktuellste Version herunterladen. Anschließend musst du das Programm noch installieren.



Schritt 3 – Enigmail einrichten

Nach der Installation des Add-ons ist es notwendig, dass du Thunderbird neu startest. Nach dem Neustart öffnet sich ein Fenster mit dem Einrichtungs-Assistenten von Enigmail. Sollte das Fenster nicht erscheinen, so kannst du es unter dem Menüpunkt *Enigmail > Einrichtungs-Assistent* aufrufen. Folge anschließend den folgenden Anweisungen.



Willkommen im Enigmail-Installations-Assistent

Es sieht aus, als ob Sie Enigmail auf diesem Computer zum ersten Mal gestartet haben. Um Enigmail zu nutzen, müssen Sie es zunächst einrichten.

Dieser Assistent führt Sie durch die Einrichtung.

Wollen Sie Enigmail jetzt einrichten, oder möchten Sie dies später tun?

Jetzt einrichten

Später

< Zurück

Weiter >

Abbrechen

Wähle *Jetzt einrichten* aus und klicke auf *Weiter >*. Anschließend kannst du auswählen, dass du die *Standard-Konfiguration bevorzugst* und dann auf *Weiter >* klicken. Sollte dir im nächsten Schritt angezeigt werden, dass GnuPG (eine Umsetzung von PGP) benötigt, dann wähle den entsprechenden Pfad zu GnuPG aus und klicke auf *Weiter >*. Nun solltest du die Aufforderung erhalten, einen OpenPGP-Schlüssel zu erzeugen.

OpenPGP-Schlüssel erzeugen

Erzeugen eines Schlüssels zum Unterschreiben und Verschlüsseln

Dieser Dialog wird ein Paar von zwei Schlüsseln erzeugen:

Mit Ihrem **öffentlichen Schlüssel** können **Andere** Mails an Sie verschlüsseln (und von Ihnen unterschriebene Nachrichten prüfen). Sie dürfen ihn jedem geben.

Ihr **geheimer, privater Schlüssel** ist nur für **Sie**, um damit Mails an Sie zu entschlüsseln und um Mails, die Sie schicken, zu unterschreiben. Diesen Schlüssel halten Sie geheim, Sie geben ihn niemandem.

Ihre **Passphrase** ist ein Passwort, mit dem GnuPG Ihren privaten Schlüssel schützt. Es soll Missbrauch Ihres privaten Schlüssels verhindern. Die Passphrase sollte ein Satz aus mindestens 8 Zeichen, Ziffern und Satzzeichen sein. Umlaute und andere sprachenspezifische Zeichen, zum Beispiel ä, é, ñ, sind **nicht empfehlenswert** (weil nicht jedes Programm damit richtig umgeht).

Konto / Benutzerkennung:

Passphrase

Bitte bestätigen Sie Ihre Passphrase durch erneutes Eingeben

Qualität der Passphrase:

< Zurück

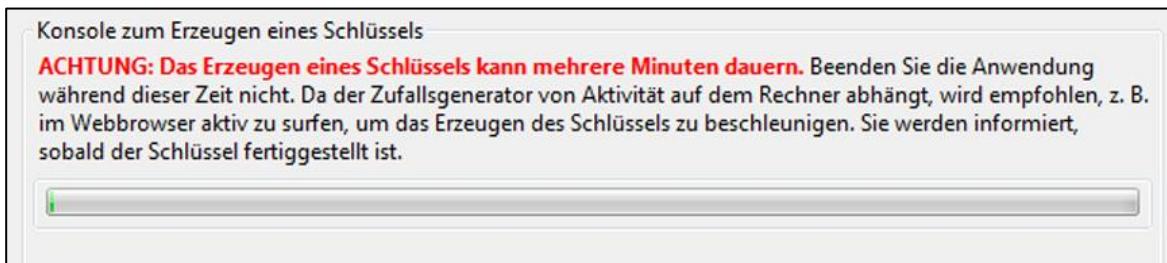
Weiter >

Abbrechen



Um ein Schlüsselpaar zu erzeugen benötigt der Assistent eine Passphrase (ein Passwort). Achte darauf, dass es möglichst sicher ist, d. h. dass die Qualitätsleiste möglichst voll ist.

Sobald du auf Weiter > geklickt hast, sollte das Schlüsselpaar generiert werden. Dies kann einige Zeit benötigen.



Speichere nun sicherheitshalber noch das Widerrufszertifikat ab. Dabei solltest du darauf achten, dass du es jetzt oder später an einen sicheren Ort ablegst. Das Widerrufszertifikat dient dazu, das soeben generierte Schlüsselpaar, dass nun mit deiner E-Mailadresse verknüpft ist, zu widerrufen. Mit einem Klick auf Weiter > ist die Einrichtung für deinen E-Mailaccount abgeschlossen.

Schritt 4 – Eine verschlüsselte E-Mail verschicken

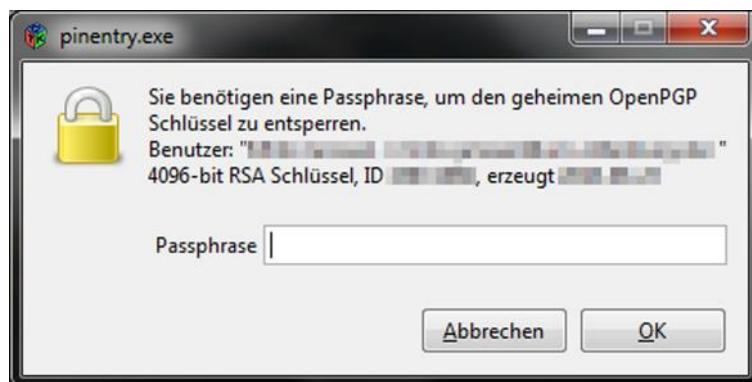
Sobald du nun in Thunderbird eine E-Mail verschicken willst, hast du die Möglichkeit diese zu unterschreiben und/oder zu verschlüsseln.



Beachte, dass du zum Verschlüsseln einer Nachricht den öffentlichen Schlüssel des Empfänger benötigst. Solltest du diesen nicht besitzen, wirst du vor dem Versenden gefragt, welchen Schlüssel du verwenden willst. Es empfiehlt sich also, nach dem Erzeugen eines Schlüsselpaares den eigenen öffentlichen Schlüsseln mit deinen Bekannten und Freunden auszutauschen. Beim unterschreiben einer E-Mail wird der öffentliche Schlüssel automatisch beigefügt, außerdem weiß der Empfänger so, dass du der Absender bist.



Versendest du eine verschlüsselte oder unterschriebene Nachricht, dann wirst du immer nach der dazugehörigen Passphrase gefragt.





IT2SCHOOL – GEMEINSAM IT ENTDECKEN

MODERNE KRYPTOLOGIE

VORWORT

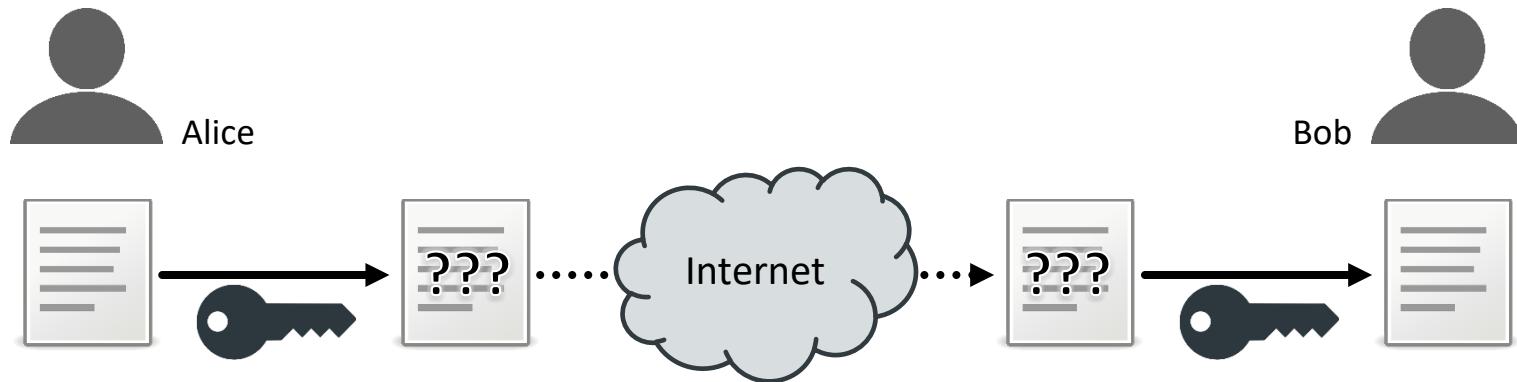
Die Mathematik ermöglicht es die uns bekannte Kryptologie zu modernisieren und komplexere Verfahren zu entwickeln!

Wie dies genauer funktioniert wird euch auf den folgenden Folien gezeigt.
Wichtige Grundlagen dafür sind:

- symmetrische und asymmetrische Verschlüsselung
- Public- und Private-Keys
- RSA
- PGP

SYMMETRISCHE VERSCHLÜSSELUNG

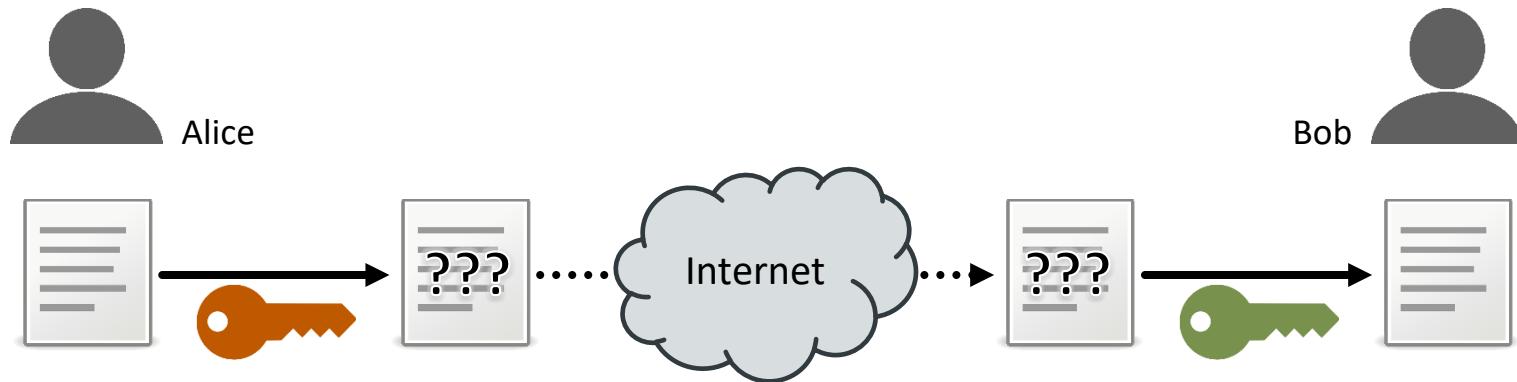
Alice schickt Bob eine verschlüsselte Nachricht über das Internet.
Zum Ver- und Entschlüsselt wird der selbe Schlüssel bzw. das selbe Verfahren verwendet.



ASYMMETRISCHE VERSCHLÜSSELUNG

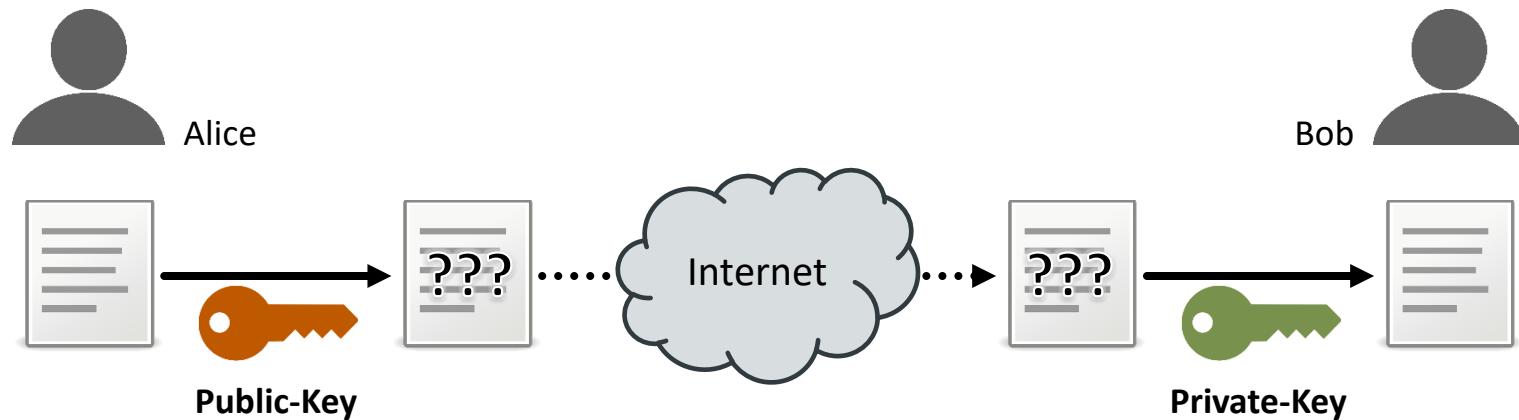
Alice schickt Bob eine verschlüsselte Nachricht über das Internet.

Zum Ver- und Entschlüsseln wird nicht der selbe Schlüssel bzw. das selbe Verfahren verwendet.



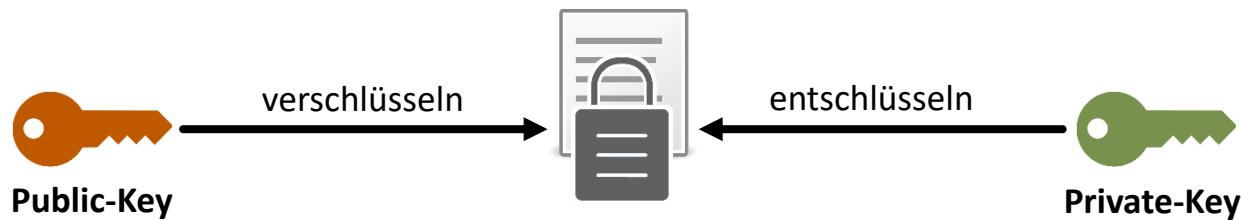
PUBLIC- UND PRIVAT-KEY

Bei der asymmetrischen Verschlüsselung wird ein Schlüssel bekannt gegeben, um die Nachrichten an eine Person zu verschlüsseln und einer wird geheim gehalten, um die verschlüsselten Nachrichten zu entschlüsseln.



PUBLIC- UND PRIVAT-KEY

Wichtig ist, dass der Public- und Private-Key ein Schlüsselpaar sind und immer zusammen gehören bzw. zusammen erstellt werden. Bildlich gesprochen, sind beide für das gleiche Sicherheitsschloss zuständig, nur einer dient zum zuschließen und der andere zum öffnen.



UND WAS HAT DAS MIT DEN BISHERIGEN VERFAHREN ZU TUN?

- Bisher hatten wir im Unterricht nur symmetrische Verfahren, meist auch nur mit Substitution
 - Zum Ver- und Entschlüsseln (bei Caesar, Vigenére etc.) wurde immer das gleiche Schlüsselwort oder Verfahren verwendet
 - Buchstaben werden nur durch andere Buchstaben ausgetauscht
- Bisher noch keine besondere Mathematik oder Logik in den Verfahren
 - Erlaubt uns Buchstaben zu codieren und dann zu verschlüsseln
 - Erlaubt uns Schlüsselpaare zu erzeugen
 - Ein Beispiel dafür ist RSA

RSA

- Benannt ist das Verfahren nach seinen drei Entwicklern:
 - Rivest
 - Shamir
 - Adleman
- Es gibt mehrere Phasen des Verfahrens:
 - Schlüsselpaar erzeugen
 - Nachricht verschlüsseln
 - Nachricht entschlüsseln

RSA – SCHLÜSSELPAAR ERZEUGEN

- Wir brauchen zwei sehr große Primzahlen p und q
- Es wird das Produkt $pq = F$ berechnet
 - Wichtig ist, dass man anhand des Produkt nicht so leicht erkennen kann, welche Primzahlen verwendet wurden! (Wichtig für die Sicherheit)
- Mit mathematischen Hilfsmitteln werden nun die Zahlen e und d ermittelt
 - (F, e) bilden den Public-Key
 - (F, d) bilden den Private-Key

RSA – NACHRICHT VERSCHLÜSSELN

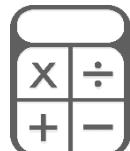
- Die Nachricht wird so codiert, dass sie aus Zahlen besteht
 - Die Codierung kann z. B. mit dem ASCII Code erfolgen
 - Wir bezeichnen die codierte Nachricht als N
- Zum Verschlüsseln brauchen wir den Public-Key
 - Die Verschlüsselte Nachricht M errechnet sich durch

$$M \equiv N^e \pmod{F}$$

RSA – NACHRICHT ENTSCHLÜSSELN

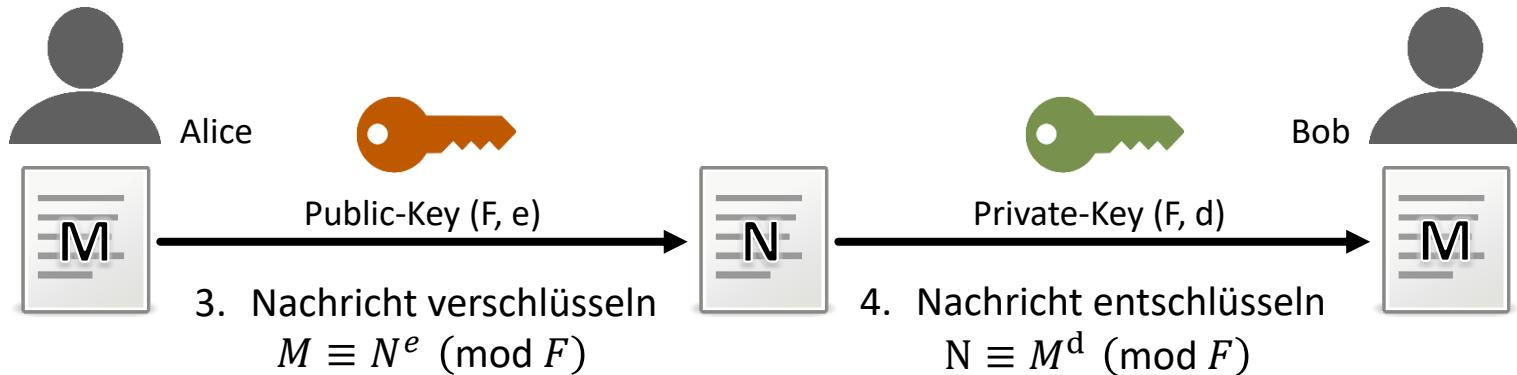
- Zum Entschlüsseln brauchen wir den Private-Key und die verschlüsselte Nachricht M
 - Die codierte Nachricht N errechnet sich durch
$$N \equiv M^d \pmod{F}$$
- Die Nachricht N kann nun so codiert werden, dass die ursprüngliche Nachricht lesbar ist
 - Wie die Nachricht codiert wurde, kann sowohl Sender als Empfänger bekannt sein

RSA – SCHAUBILD



1. Große Primzahlen p und q wählen
2. Berechnung von F, e und d

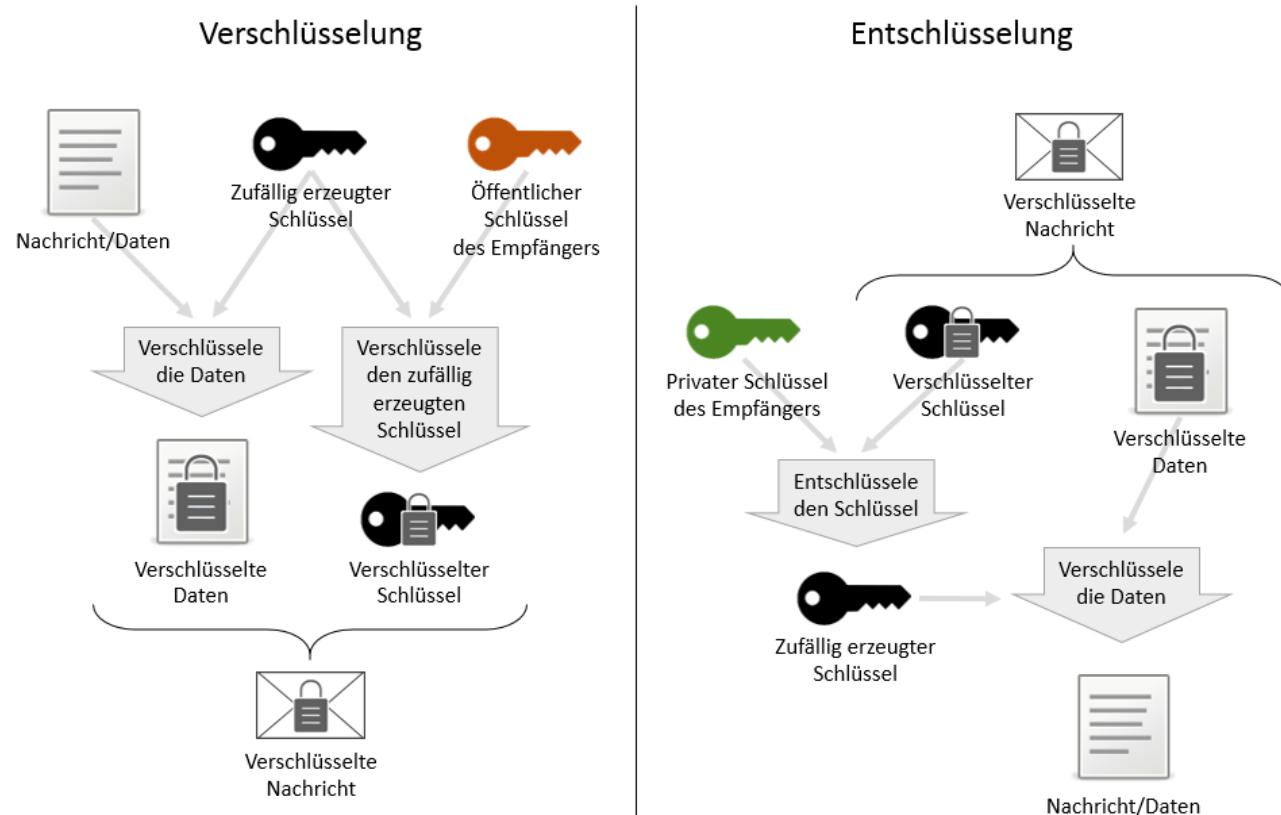
Wichtig ist hier natürlich, dass Bob seinen Public-Key für Alice bereitstellt, damit sie für ihn auch eine Nachricht verschlüsseln kann.



PGP

- PGP steht für Pretty Good Privacy
- Programm zum Verschlüsseln und Unterschreiben von Daten
- Verwendet hybrides Verfahren aus symmetrischer und asymmetrischer Verschlüsselung
- Verwendete in der ersten Version RSA zum Verschlüsseln (später Elgamal)
- Häufig verwendet zur Verschlüsselung von E-Mails

PGP – SCHAUBILD



Dateien verschlüsseln

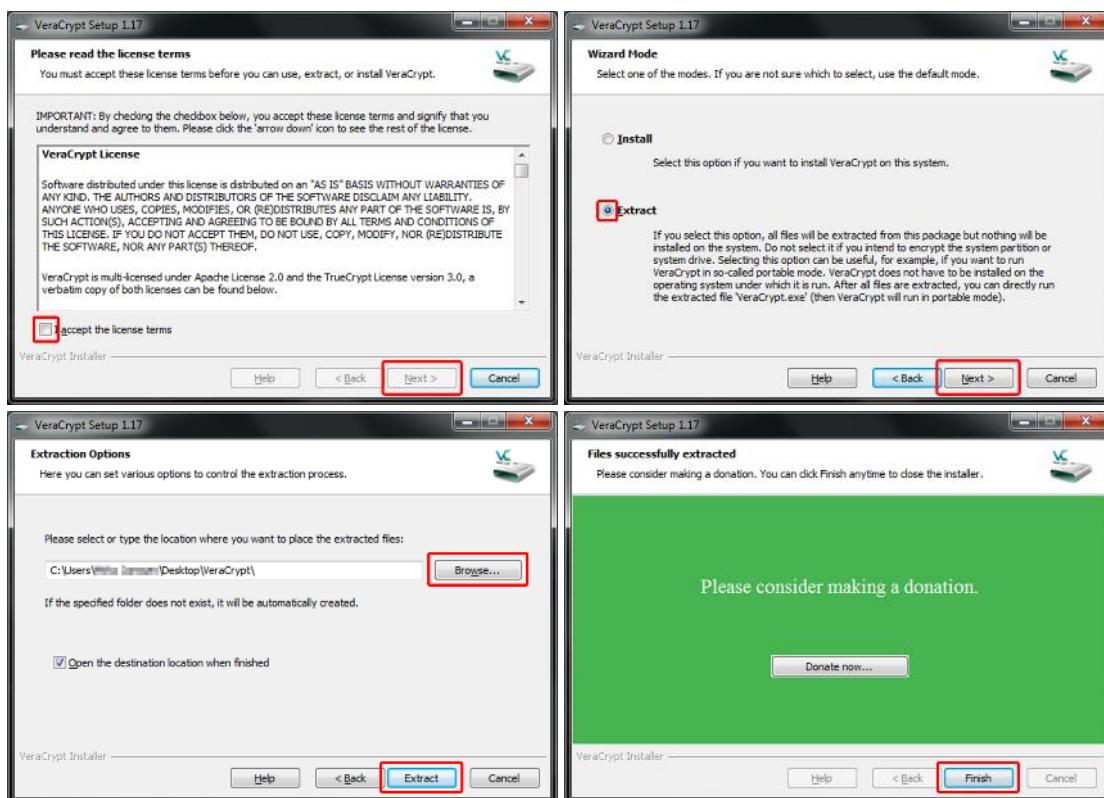
Zum Verschlüsseln von Dateien gibt es mehrere Programme. In dieser Anleitung verwenden wir die kostenlose, opensource Software VeryCrypt. Mit diesem Programm haben wir die Möglichkeit verschlüsselte Kontainer zu erstellen und einzubinden. Dies bedeutet, dass sich auf der Festplatte ein bestimmter Bereich mit verschlüsselten Daten befindet, der nur über diese Software und dem dazugehörigen Schlüssel/Passwort zugänglich gemacht werden kann.

Schritt 1 – Installation

VeryCrypt bietet die Möglichkeit, dass man das Programm auf dem Computer installiert oder aber eine portable Version entpackt. Zum Testen (und eigentlich auch zum Benutzen) reicht die portable Version aus, weshalb wir hier diese Variante benutzen wollen. Solltest du in Zukunft das Programm häufiger nutzen, kannst du es natürlich auch installieren.

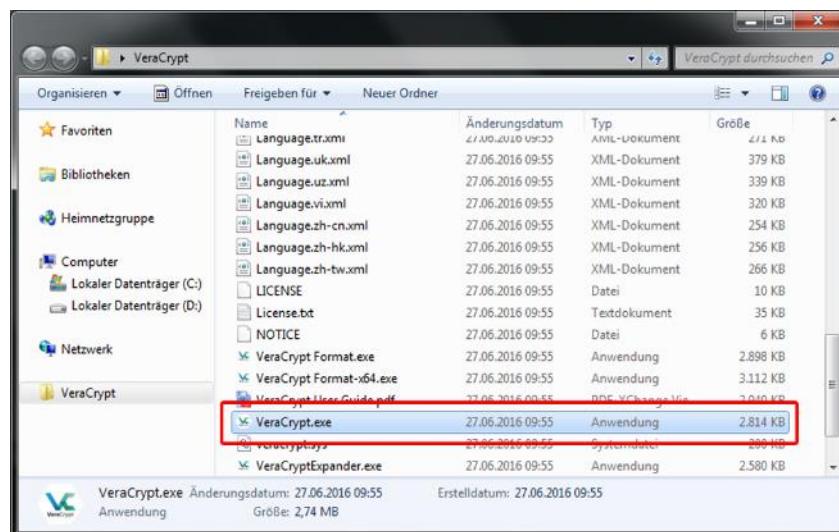
Starte die Installationsdatei und akzeptiere die Lizenzbedingungen. Klicke anschließend auf *Next >* und wähle *Extract* für die portable Variante aus. Klicke wieder auf *Next >*, wähle das Zielverzeichnis aus und klicke anschließend auf *Extract*. Nachdem das Extrahieren beendet ist, kannst du die „Installation“ mittels *Finish* beenden.

Hinweis: Im letzten Schritt wird lediglich um eine Spende (Donation) gebeten. Diese ist zur Nutzung des Programms nicht notwendig.

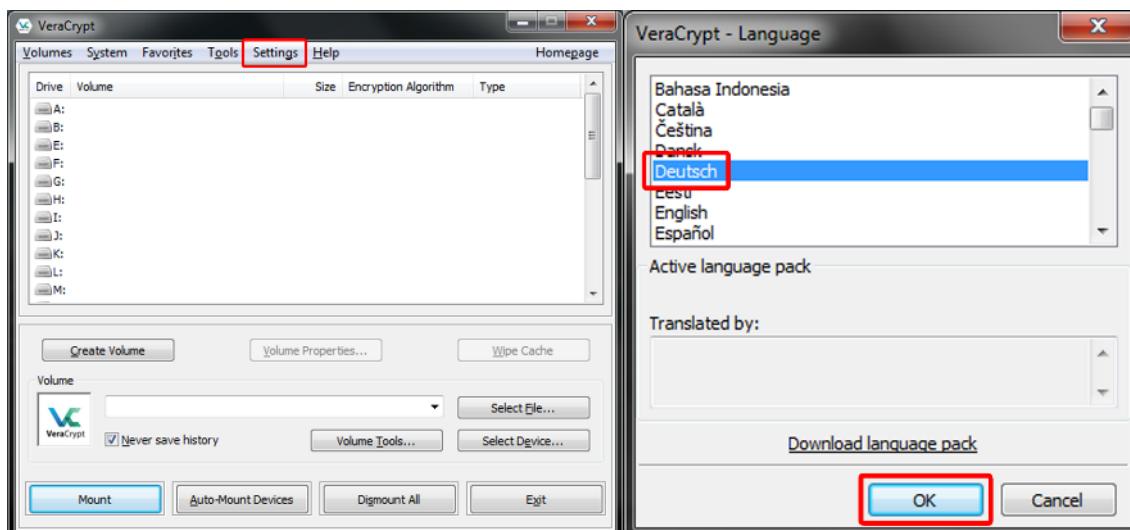


Schritt 2 – Mit VeraCrypt einen Kontainer erstellen

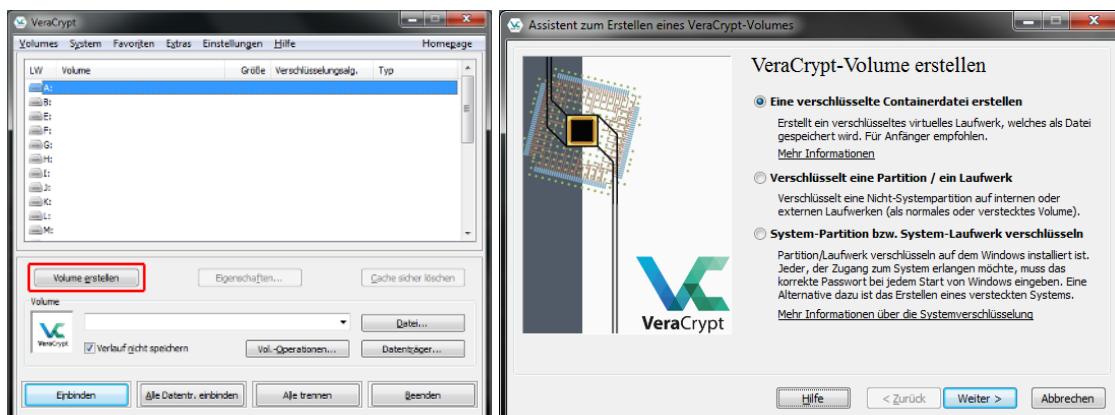
An dem Zielpfad, an dem das Programm extrahiert wurde, sollte sich nun die *VeraCrypt.exe* befinden. Führe die *exe-Datei* aus und starte damit das Programm.



Du solltest nun die englische Oberfläche des Programms sehen. Damit du diese auf Deutsch stellen kannst, wählst du unter *Settings* den Auswahlpunkt *Language...* aus. Anschließend solltest du die deutsche Oberfläche sehen.



Um nun einen Kontainer zu erstellen musst du auf *Volume erstellen* klicken, woraufhin sich ein neues Fenster mit einem Assistenten zur Erstellung des Kontainers öffnet.

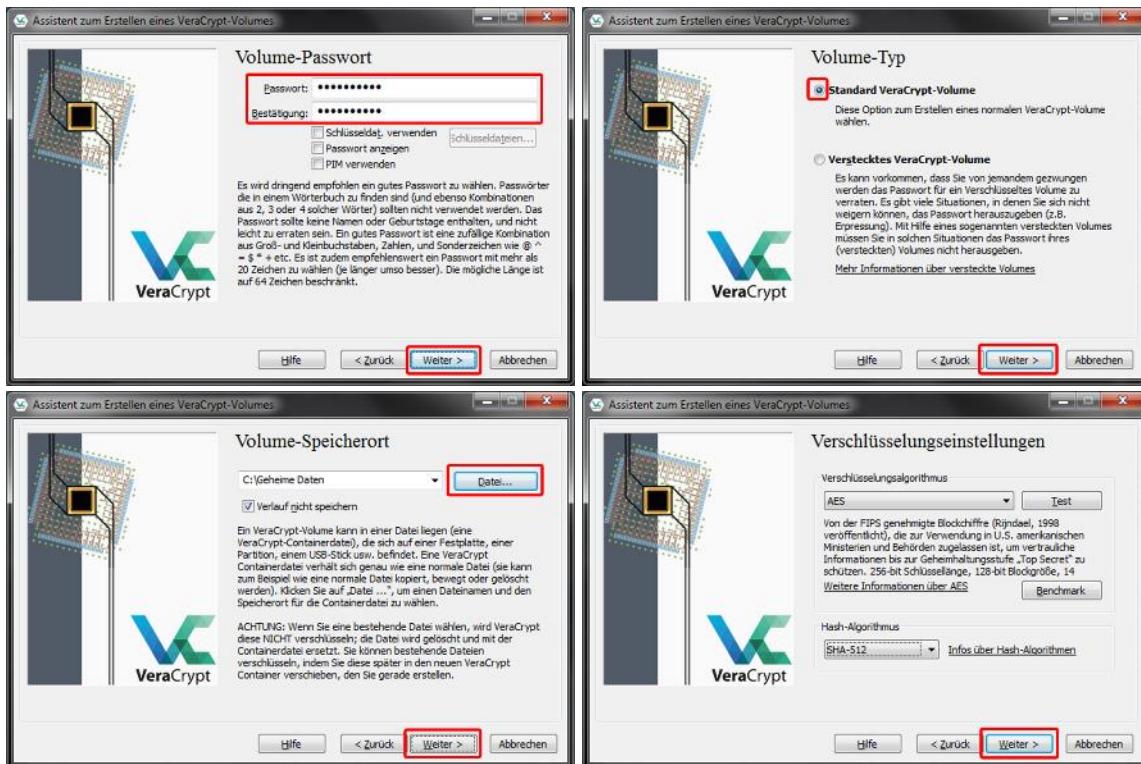


Diese Anleitung beschreibt den Assistenten nicht vollumfänglich, sondern geht sehr zügig durch die einzelnen Punkte. Bitte lies jede Seite des Assistenten ausführlich durch, bevor du

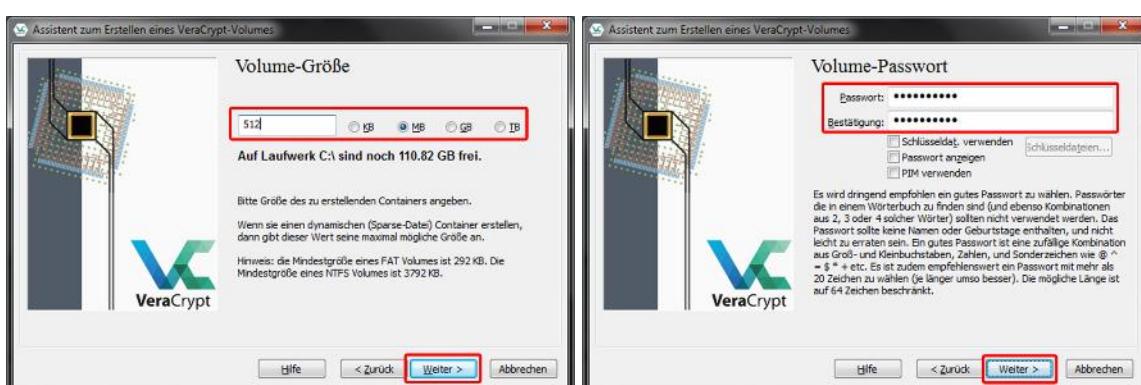


die von uns angegebenen Schritte ausführst. Beachte auch die Hinweise und Anmerkungen des Assistenten!

Wie bereits gesagt, wollen wir einen Kontainer und keine Partition oder gar das Systemlaufwerk verschlüsseln, wähle daher *Eine verschlüsselte Containerdatei erstellen* aus und klicke auf *Weiter >*. Wähle anschließend *Standard VeryCrypt-Volume* aus, da wir unseren Kontainer nicht auf dem System verstecken wollen. Danach musst du über *Datei... einen Zielpfad für den Kontainer auswählen* und kannst dann auf *Weiter >* klicken. Auf der anschließenden Seite kannst du das Verschlüsselungsverfahren auswählen.

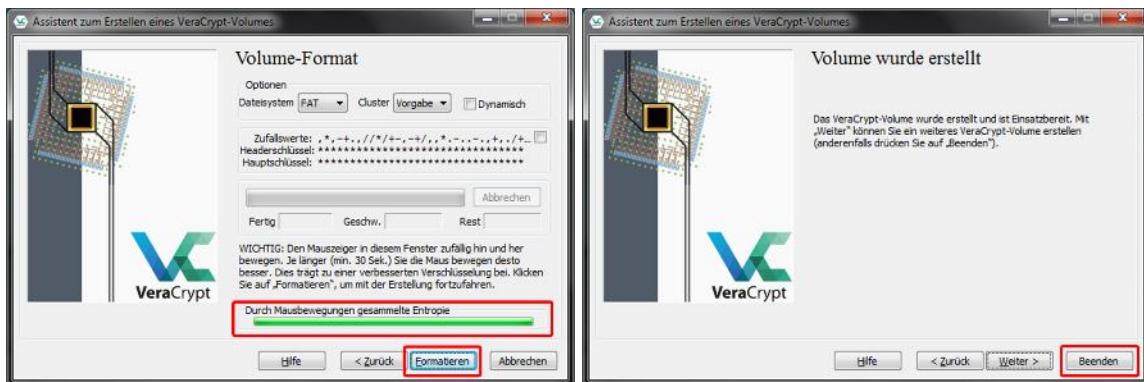


Belasse es zunächst beim Standard (AES mit SHA-512) und klicke auf *Weiter >*, um dann die Größe des Kontainers zu definieren. Für diese Anleitung verwendeten wir 512 MB. Klicke anschließend auf *Weiter >* und du wirst aufgefordert ein Passwort festzulegen. Trage in die entsprechenden Felder das Passwort ein und klicke auf *Weiter >*. Solltest du einen Hinweis bekommen, dass das Passwort kurz sei, dann wähle *Nein* um das Passwort zu ändern, oder *Ja* um trotzdem fortzufahren. (Für dieses Beispiel ist die Länge des Passworts ersteinmal egal.)



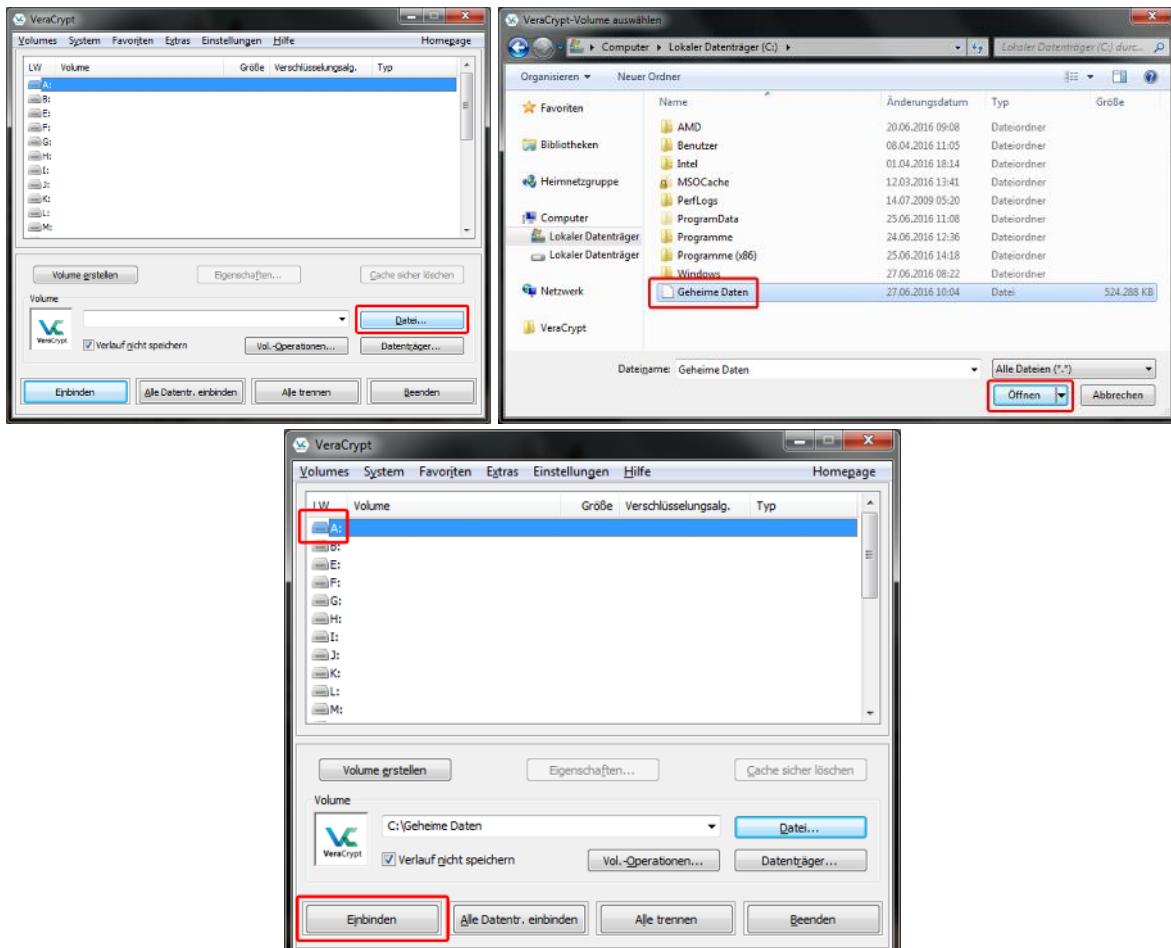
Auf der anschließenden Seite kannst du das Dateisystem des Kontainers auswählen sowie die Formatierung starten. Dafür musst du jedoch mit der Maus solange innerhalb des Fensters

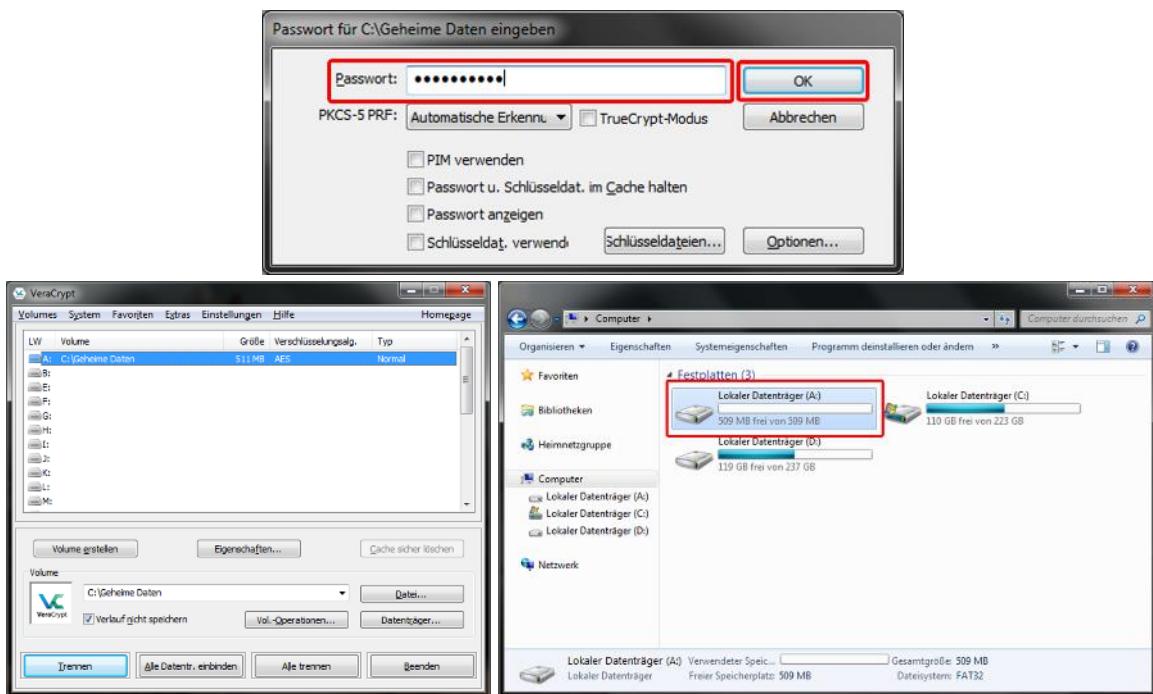
zufällige Bewegungen machen, bis die Leiste Durch *Mausbewegungen gesammelte Entropie* voll ist. Erst dann kannst du auf Formatieren klicken. Sobald der Kontainer erstellt wurde, kannst du den Assistenten mit *Beenden* beenden.



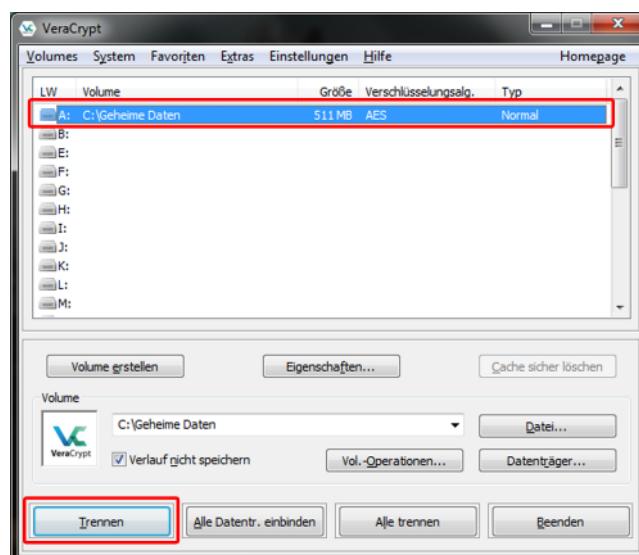
Schritt 3 – Kontainer einbinden

Um einen Kontainer nun einzubinden, das bedeutet, dass man Dateien dort ablegen und somit verschlüsseln lassen kann, musst du zunächst auf *Datei...* klicken und die entsprechende Datei des Kontainers auswählen. Anschließend kannst du ein Laufwerk aus der oberen Liste auswählen und mit *Einbinden* den Kontainer einbinden. Im Anschluss musst du nun das Passwort für den Kontainer eingeben und kannst dann sehen, dass der Container eingebunden wurde. Danach kannst diesen Kontainer wie ein ganz normales Laufwerk verwenden.





Zum Trennen des Kontainers (damit das Laufwerk nicht mehr verfügbar ist), kannst du diesen in der oberen Liste auswählen und auf *Trennen* klicken.



Vorsicht

Denke immer daran, dass wenn du einmal das Passwort vergessen hast, dann ist es sehr schwer bis gar nicht möglich, die verschlüsselten Dateien wiederherzustellen. Dies ist gerade dann sehr ärgerlich, wenn du die Partition des Betriebssystems (also z. B. dein Windows) verschlüsselt hast. Verwende also immer sichere Passwörter, die du dir auch merken kannst. Wie das am besten geht, zeigt dir folgendes Beispiel:

Angenommen du brauchst ein sicheres Beispiel. Wähle ein Lied, Gedicht oder eine Stelle aus deinem Lieblingsbuch aus. Wir nehmen hier als Beispiel die erste Zeile des Erlkönigs:

Wer reitet so spät durch Nacht und Wind?



Arbeitsmaterial A2.7

Anschließend ersetzen wir bestimmte Buchstaben durch Zahlen. Zum Beispiel e durch 3, i durch 1, a durch 4, s durch 5, t durch / und o durch 0. Dann erhalten wir:

W3r r31|3< 50 5pä| durch N4ch| und W1nd?

Danach ersetzen wir bestimmte Buchstaben oder Wörter durch Sonderzeichen. Zum Beispiel und durch &, c durch < und n durch \V:

W3r r31|3| 50 5pä| dur<h \V4<h| & W1\Vd?

Zum Abschluss entfernt man alle Leerzeichen und erhält es sehr sicheres Passwort:

W3rr31|3|505pä|dur<h\V4<h|&W1\Vd?

Natürlich bedarf es etwas Übung, bis man es ohne große Mühe am Stück eingeben kann.

Substitution – Teil 1

Verschlüsseln von Texten ist mit vielen verschiedenen Verfahren möglich. Einige dieser Verfahren lassen sich durch den Begriff *Substitution* beschreiben. Bei der Substitution werden Buchstaben oder sogar ganze Wörter des Klartextes durch andere Buchstaben, Wörter oder auch Symbole ersetzt. Ein Beispiel, das euch vielleicht bereits bekannt ist, stellt die Caesar-Verschlüsselung dar. Es gibt jedoch auch noch die Verschlüsselung nach Atbash und mittels Polybius-Tafel.

Atbash- und Ceasar-Verschlüsselung

Bei beiden Verschlüsselungen werden die Buchstaben des Alphabets durch andere Buchstaben ausgetauscht. Dies kann jedoch auf auf verschiedene Arten erfolgen.

So wird bei Atbash der erste Buchstabe des Alphabets mit dem Letzten, der Zweite mit dem Vorletzten und so weiter ausgetauscht.

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
Z	Y	X	W	V	U	Z	S	R	Q	P	O	N	M	L	K	J	I	H	G	F	E	D	C	B	A
Atbash Geheimalphabet																									

Anders erfolgt bei Caesar eine Verschiebung des Alphabets. Hier wird jeder Buchstabe mit dem Buchstaben ersetzt, der um einen festen Wert später im Alphabet auftaucht.

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
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	F
Caesar Geheimalphabet mit einer Verschiebung von 6																									

Aufgabe 1

Verschlüssle die folgende Nachrichten mit Atbash:

TREFFEN UM DREI

Aufgabe 2

Entschlüssle die folgende Nachricht mit Atbash:

WVI TVRVI SZG WRV YVFGV



Verschlüsselung mittels Polybius-Tafel

Die Buchstaben eines Klartextes müssen jedoch nicht immer nur durch Buchstaben ersetzt werden. Es ist auch möglich sie durch Zahlen zu ersetzen, wie das Verfahren mittels Polybius-Tafel zeigt.

Die Polybius-Tafel stellt eine Art Tabelle (oder auch Matrix genannt) dar, in der jeder Buchstaben des Alphabets eingetragen wird und eine X- und Y-Koordinate erhält.

		Y				
		1	2	3	4	5
X	1	A	B	C	D	E
	2	F	G	H	I/J	K
	3	L	M	N	O	P
	4	Q	R	S	T	U
	5	V	W	X	Y	Z

Die Anzahl der Zeilen und Spalten kann dabei beliebig angepasst werden. Im obigen Beispiel wird nun das A mit „1 1“, „11“ bzw. „1,1“ oder B mit „1 2“, „12“ bzw. „1,2“ verschlüsselt (für die Schreibweise der Koordinaten gibt es noch etliche weitere Möglichkeiten).

Aufgabe 3

Verschlüssle die Nachricht aus Aufgabe 1 mittels der obigen Polybius-Tafel.

Aufgabe 4

Entschlüssle die folgende Nachricht:

14 15 42

43 13 23 31 45 15 43 43 15 31

31 24 15 22 44

11 32

21 31 45 43 43



Transposition

Verschlüsseln von Texten ist mit vielen verschiedenen Verfahren möglich. Einige dieser Verfahren lassen sich durch dem Begriff *Transposition* beschreiben. Bei der Transposition werden (meist) Buchstaben des Klartextes so verschoben, dass die eigentlich Nachricht nicht mehr direkt lesbar ist. Dabei kann es vorkommen, dass zusätzliche Gegenstände zum Entschlüsseln notwendig sind (siehe Skytale), dies ist aber nicht zwingend der Fall (siehe fleißnersche Schablone, Gartenzaun oder Krebs-Verfahren).

Skytale-Verschlüsselung

Bei der Skytale-Verschlüsselung wird ein Holzstab (der sogenannte Skytale) mit einem bestimmten Durchmesser verwendet. Um diesen Holzstab wird dann das Papier (oder damals Leder) herumgewickelt und darauf geschrieben (siehe rechtes Bild). Der Durchmesser ist der Schlüssel.



Aufgabe 1

Verschlüssle die folgende Nachricht mit einer der bereitgestellten Skytales:

Austausch heute Abend am bekannten Ort

Aufgabe 2

Suche dir eine Partnerin oder einen Partner und verschlüsselt jeweils eine Nachricht mit einem der bereitgestellten Skytales.

Verschlüsselung mittels fleißnersche Schablone

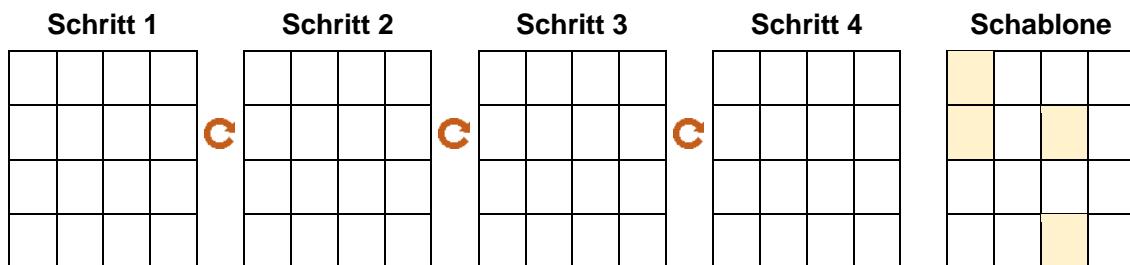
Ein anderes Hilfsmittel stellt die fleißnersche Schablone dar (die nicht zwingend physisch vorhanden sein muss). Der Klartext wird hier in eine Art Tabelle bzw. Matrix geschrieben, wobei eine Schablone vorgibt, wo etwas eingetragen werden darf. Sind alle erlaubten Felder genutzt, wird die Schablone mit oder gegen den Uhrzeigersinn (aber immer einheitlich) gedreht und der verbleibende Text mittels Schablone eingetragen. Das folgende Beispiel zeigt wie der Text „Ich mag Informatik“ verschlüsselt wird. Die gelben Kästchen geben die erlaubten Felder der Schablone an.

Schritt 1		Schritt 2		Schritt 3		Schritt 4		Schablone	
I				I	A	F	G		
				I					
C		H		C	N	H			
	M			M					



Aufgabe 3

Verschlüssle einen Satz mittels fleißnerscher Schablone.



Aufgabe 4

Entschlüssle die folgende Nachricht.

Nachricht	Schablone
B R H T	
R H C E	
A I K U	
A M U M	



Substitution – Teil 2

Verschlüsseln von Texten ist mit vielen verschiedenen Verfahren möglich. Einige dieser Verfahren lassen sich durch den Begriff *Substitution* beschreiben. Weiter gibt es aber auch noch die Unterteilung in *Mono-* und *Polyalphabetisch*. Bei Ersterem findet nur ein Alphabet Verwendung (siehe Caesar- oder Atbash-Verschlüsselung), was das „Knacken“ mittels einer Häufigkeitsanalyse einfach ermöglicht. Bei polyalphabetischen Verfahren werden mehrere Geheimalphabete verwendet und somit gleiche Buchstaben durch mehrere verschiedene Buchstaben ersetzt. Ein Beispiel für ein solches Verfahren ist die Vigenére-Verschlüsselung.

Vigenére-Verschlüsselung

Dieses Verfahren ähnelt der Caesar-Verschlüsselung, nur mit dem Unterschied, dass man hier nicht ein Geheimalphabet (das verschobene Alphabet) verwendet, sondern mehrere. Damit die Person, die die Nachricht verschlüsselt und die Person, die die Nachricht entschlüsselt, wissen welche und wieviele Geheimalphabete verwendet werden, einigen sie sich auf ein *Schlüsselwort*.

Die Positionen der einzelnen Buchstaben des Schlüsselwortes im Alphabet geben an, um wieviel das Alphabet verschoben wird. Die Anzahl der Geheimalphabete entspricht dabei der Länge des Schlüsselwortes. Wird zum Beispiel das Schlüsselwort *GEHEIM* verwendet, dann werden zum Verschlüssel des Klartextes sechs Geheimalphabete verwendet:

1. Geheimalphabet ist das Alphabet um **sechs** Buchstaben verschoben.
2. Geheimalphabet ist das Alphabet um **vier** Buchstaben verschoben.
3. Geheimalphabet ist das Alphabet um **sieben** Buchstaben verschoben.
4. Geheimalphabet ist das Alphabet um **vier** Buchstaben verschoben.
5. Geheimalphabet ist das Alphabet um **acht** Buchstaben verschoben.
6. Geheimalphabet ist das Alphabet um **zwölf** Buchstaben verschoben.

Die Tabelle auf der folgenden Seite veranschaulicht dies nochmal für alle Buchstaben.

Soll nun der Text *ICH MAG INFORMATIK* verschlüsselt werden, dann wir der erste Buchstabe (I) mit dem ersten Geheimalphabet verschlüsselt, der zweite mit dem zweiten und so weiter, bis man wieder beim ersten Geheimalphabet anfängt:

I	^G →	O	I	^G →	O	A	^G →	G
C	^E →	G	N	^E →	R	T	^E →	X
H	^H →	O	F	^H →	M	I	^H →	P
M	^E →	Q	O	^E →	S	K	^E →	O
A	^I →	I	R	^I →	Z			
G	^M →	S	M	^M →	Y			

Aus „ICH MAG INFORMATIK“ wird also „OGO QIS ORMSZYGXPO“.



Buchstabe des Klartextes																											
	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	
A	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	
B	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	A	
C	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	B	
D	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	C	
E	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	D	
F	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	E	
G	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	F	
H	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	G	
I	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	H	
J	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	I	
K	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	J	
L	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	K	
M	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	L	
N	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	M	
O	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	N	
P	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	O	
Q	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	P	
R	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	Q	
S	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	R	
T	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	S	
U	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	
V	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	U	
W	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	V	
X	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	W	
Y	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	
Z	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	Y	

Aufgabe 1

Verschlüssle die folgende Nachricht mit der Vigenére-Verschlüsselung und dem Schlüsselwort GEHEIM:

TREFFEN UM DREI

Aufgabe 2

Entschlüssle die folgende Nachricht mit dem Schlüsselwort GEHEIM.

JIY KMUKV OEB POI IICFK



Vigenére knacken

Das Knacken der Vigenére-Verschlüsselung ist nicht so einfach wie bei der Caesar-Verschlüsselung, jedoch ähnlich, da auch hier eine Häufigkeitsanalyse stattfindet. Die Schwierigkeit ist aber, auf das Schlüsselwort zu kommen. Hier gibt es Möglichkeiten, die das Ermitteln des Schlüsselwertes überaus schwierig gestalten. Zunächst sollen jedoch einfache Schlüsselwörter betrachtet werden, um das Vorgehen einmal grundlegend zu verstehen.

Es sind drei Schritte notwendig, um die Verschlüsselung zu knacken:

1. Wiederkehrende Buchstabenfolgen im verschlüsselten Text werden gesucht.
2. Anschließend wird hiermit die Länge des Schlüsselwertes ermittelt.
3. Letztlich erfolgt dann die Bestimmung des Schlüsselwertes.

Wie aber auch schon bei der Häufigkeitsanalyse bei Caesar, fällt das Knacken leichter, wenn der Geheimtext sehr lang ist. Also wenn man eine große Datengrundlage benutzt.

Schritt 1: Wiederkehrende Buchstabenfolgen suchen

Der Geheimtext wird zunächst auf wiederkehrende Buchstabenfolgen untersucht. Diese sollten im Idealfall eine Mindestlänge von drei Buchstaben haben und innerhalb des Textes markiert werden. Die Buchstabenfolgen zeigen mit hoher Wahrscheinlichkeit, die gleichen Buchstaben des Geheimalphabets, mit dem verschlüsselt wurde. Das bedeutet, dass häufig vorkommende Buchstabenfolgen innerhalb des Geheimtextes auf häufig vorkommende Buchstabenfolgen (z. B. ein, der, die, das, sch ...) aus der deutschen Sprache beruhen.

```
ETYJANSUVYECVPRIDEIIYPMNQLGHPCXEIERINSXIXXIRLMIRPDAICOIIYQECSPVWP
YRETYWOWNLETYJANSIREPBTZQZPGPYZOYMYCSDXAMPRBPDMTKEHIPTQMPCAIPOIRVZQ
MPYSBOTISPCLIIYQECSEIXEYYNPTRFLNLZFVRANVINTDXWTCHSTNLZPTKEYLFECOMED
PVETYJANSITPIXIDEWINSRPTRFLNLECKYKYLGKPYELDLRDCIETYJANSITPIXEPTR
ETYJANSIRDNLFPWSPWQANSXEDLYCSPMNQLGHPCIIYPRETYJANSINEPBTKFONLNOEY
ZFETYJANSSSDPCRINSXETYJANSHIPDIRPTRFLNLEPBHTVDOTGHDTGHPCJOCOIRY
```

Schritt 2: Länge des Schlüsselwertes ermitteln

Nun wird für jede Buchstabenfolge ermittelt, wie groß der Abstand vom ersten Buchstaben der Buchstabenfolge zum erneuten Auftreten der Folge ist. Zum Beispiel wäre für XYZ der Abstand innerhalb von AAXYZAAAAXYZAAAAAAXYZAA einmal Sieben und einmal Neun.

Aus der Tatsache, dass wiederkehrende Buchstabenfolgen innerhalb des Geheimtextes auf gleichen Buchstabenfolgen innerhalb des Klartextes zurückzuschließen sind, ergibt sich, dass die Länge des Schlüsselwertes ein Teiler des oben erwähnten Abstandes sein muss.

Das bedeutet, dass nun für jeden Abstand geschaut werden muss, welche Teiler dieser besitzt und welcher Teiler für alle Buchstabenfolgen in Frage kommen. Orientierung kann hier der ggT (größte gemeinsame Teiler) der einzelnen Abstände bieten. Wenn ein oder mehrere Teiler gefunden wurden, dann wird die Länge des Schlüsselwertes einem davon entsprechen. Wenn mehrere Teiler in Frage kommen, muss Schritt 3 gegebenenfalls mehrfach durchgeführt werden, bis das Schlüsselwort ermittelt wurde.

Buchstabenfolge	Abstand vom ersten Buchstaben zum ersten Buchstaben	ggT der Abstände pro Buchstabenfolge
ETYJANS	$76 = 2^2 * 19$, $124 = 2^2 * 31$, $48 = 2^4 * 3$, $60 = 2^2 * 3 * 5$, $24 = 2^2 * 3$, $16 = 2^4$	2^2
IITY	$36 = 2^2 * 3^2$, $88 = 2^3 * 11$, $160 = 2^6 * 5$	2^2
NLE	$152 = 2^3 * 19$, $140 = 2^2 * 5 * 7$	2^2
PTRFL	$64 = 2^6$, $140 = 2^2 * 5 * 7$	2^2

Schritt 3: Schlüsselwort ermitteln

Da nun bekannt ist, wie lang das Schlüsselwort ist, kann eine Häufigkeitsanalyse angewendet werden. Dafür wird der Text in Abschnitte von der Länge des Schlüsselwörteres unterteilt. Da bei Vigenère der erste, zweite, dritte usw. Buchstabe eines solchen Abschnitts immer mit dem ersten, zweiten, dritten usw. Geheimalphabet verschlüsselt wurde, muss nun ermittelt werden, welcher Buchstabe innerhalb dieser Menge am häufigsten vorkommt. Angenommen für den ersten Buchstaben des Schlüsselwörteres ergibt sich, dass X am häufigsten verschlüsselt wurde, dann ist der Buchstabe gesucht, mit dem das E auf X abgebildet wird.

ETYJ ANSD UVYE CVPR IDEI IYPM NQLG HPCX EIER INSX IXXI RLMI RPDA
 ICOI IYQE CSPV WPYR ETYW OWNL ETYJ ANSI REPB TQZP GPYZ OYMY CSDX
 AMPR BPDM TKEH IPTQ MPCP IPOI RVZQ MPYS BOTI SPCI IYQE CSEI XEYY
 NPTR FLNL ZFVR ANVI NTDX WTCH STNL ZPTK EYLF ECOM EDPV ETYJ ANSI
 TPIX IDEW INSI RPTR FLNL ECKY KYLG KPYE LDLR DPCI ETYJ ANSI TPIX
 EPTR ETYJ ANSI RDNL LFPW SPWQ ANSX EDLY CSPM NQLG HPCP IYPR ETYJ
 ANSI NEPB TKFO NLNO EYZF ETYJ ANS DPCR INSX ETYJ ANSH IPDI RPTR
 FLNL EEPB THTV DOTG HDTG HPCJ OCOI RY

Buchstabe des Schlüsselwörteres	Vorkommende Buchstaben mit ihrer Häufigkeit	Vermuteter Buchstabe des Schlüsselwörteres
1. Buchstabe	E = 18, I = 14, A = 11, R = 8, ...	E → E
2. Buchstabe	P = 24, N = 13, T = 12, Y = 9, ...	E → P
3. Buchstabe	Y = 15, S = 12, P = 11, T = 10, ...	E → Y
4. Buchstabe	I = 19, R = 12, J = 9, X = 8, ...	E → I

In dem aufgeführten Beispiel ergibt sich also, dass das Schlüsselwort „ALUE“ sein soll. Das dieses Verfahren nicht immer auf Anhieb den richtigen Schlüssel ergibt, lässt sich dann erkennen, wenn man versucht mit diesem Schlüsselwort zu ermitteln. Das eigentlich Schlüsselwort lautete nämlich „ALLE“ und das Verfahren zum Knacken der Verschlüsselung hätte sicherlich besser geklappt, wenn die Nachricht länger ist.

Entschlüsselt man nun den Text ergibt sich:



EINFACH ZU KNACKEN IST EIN EINFACHER TEXT NICHT IMMER ABER ES
WIRD EINFACHER WENN EIN SOLCH EINFACHER TEXT FOLGEN VON
BUCHSTABEN BESITZT DIE IMMER WIEDERKOMMEN

OB DIESER EINFACH TEXT NUN EINFACH ZU KNACKEN IST WIRD SICH
ZEIGEN

ABER DIESER EINFACHE TEXT IST SICHER EINFACHER ZU KNACKEN ALS
ANDERE EINFACHE TEXTE

EIN EINFACHER SCHLUESSEL MACHT ES AUCH EINFACHER EINEN
EINFACHEN TEXT ZU KNACKEN

OB EINFACH ODER NICHT EINFACH DIESER EINFACHE TEXT WIRD DICH
SICHER FORDERN

Aufgabe 1

Knacke die Verschlüsselung anhand des folgenden Textes und entschlüssle die ersten Zeilen des Geheimtextes¹:

PWTMYTBADKDGPWPFYWFGUESOTLUPNVYWAPKCSOOJWWASTLSUZUSJMJBRS
TIMGPYSXOJWWASMMZQLCHJQWGYDHKOJWWASTMFPADWIPVKLHONZWPDPWRA
AGOPRKNJCNPKGPJJLTHYOWOHPGYJWCUEKUZLGAOKHOGPESMZMRWPBKVFV
ZTQNLAGSFMSVWTDPWRAAGQPRKNJCNPTGTKEOMSGVLYVCHKBVKLOFOBLGNC
IVXWPLYBZAAEEOOWKEWEODZKZOGPGOMSWMPWTIFFLCTUTYGUOSLZSILYOH
EWEODSRVYHSFAVHHWGIPTGHYHCWJVLERGJWKPDHGJWTUTQNBXGZEUKTW
IAZPPMOGPWGJQWGYDHKNJCNPSOVWTZPFOMNQUQFGOWPYTQNBAIVOSXNSNZ
NVHMSPAHCXBWVDTFJRWFLASXAGPHYHCWJVLEANOWKUPTXIYGUFFSQLLHZR
KZFGPYTXIYGUOKVAEAOEAOBBCVOSXVWKUMSGVLYVCHKBODYOSTSGGUYSTA
APKYWIPLBRSRIKULYJUVWKUPFHMDKLMWMMFRLCGUVKQSWAGVVWYNVLZSI
LYROMKKJSBAZSMOWKHMILSCKZAIRPWZHMGPYSXLWTNCIVXWPIP NOMZGUS
SXIMUIPYUUEGUKICMDEOPFMZMRPGOMYGOZSBOKLGWKTWHYLUKVEWZDAG
VEKUOSYBWPZDHKTDGUFBJEWNJSSLZSILYYUMFPAPAGVKVLWZKV

Tipp: Von den interessanten Buchstabenfolgen wurde das erste Vorkommen bereits markiert. In Schritt 2 sollte nicht nur auf den ggT geachtet werden, sondern eher auf alle vorkommenden Teiler.

Buchstabenfolge	Abstand vom ersten Buchstaben zum ersten Buchstaben	ggT der Abstände pro Buchstabenfolge
OJWWAS		

¹ QUELLE: http://medienwissenschaft.uni-bayreuth.de/inik/email_nur_fuer_dich/3_verschluesseln/3.2_Vigenere/AB%20Vigenere%20knacken.pdf



TXIYGU		
YHCWJVLE		
QPRKNJ		
AEO		
PWT		

Buchstabe des Schlüsselwortes	Vorkommende Buchstaben mit ihrer Häufigkeit	Vermuteter Buchstabe des Schlüsselwortes
1. Buchstabe		E →
2. Buchstabe		E →
3. Buchstabe		E →

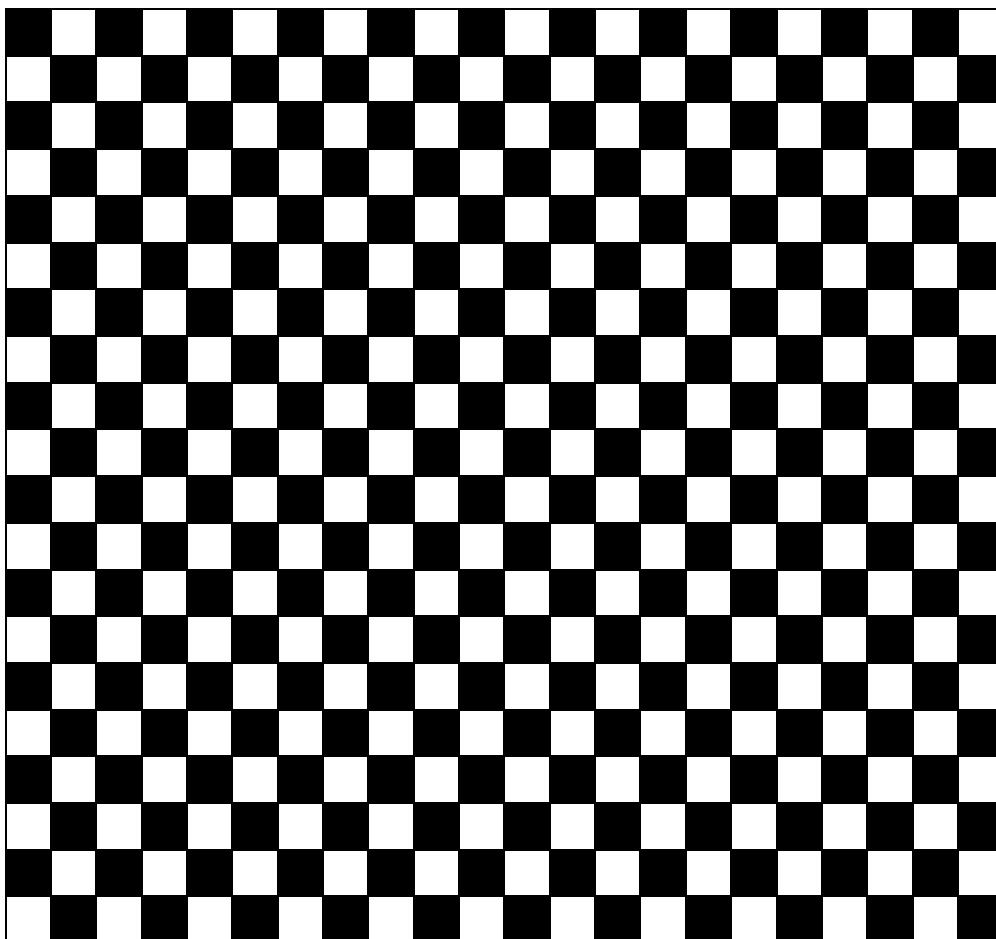
Das Schlüsselwort lautet: _____

Die ersten Zeilen des Klartextes lauten:



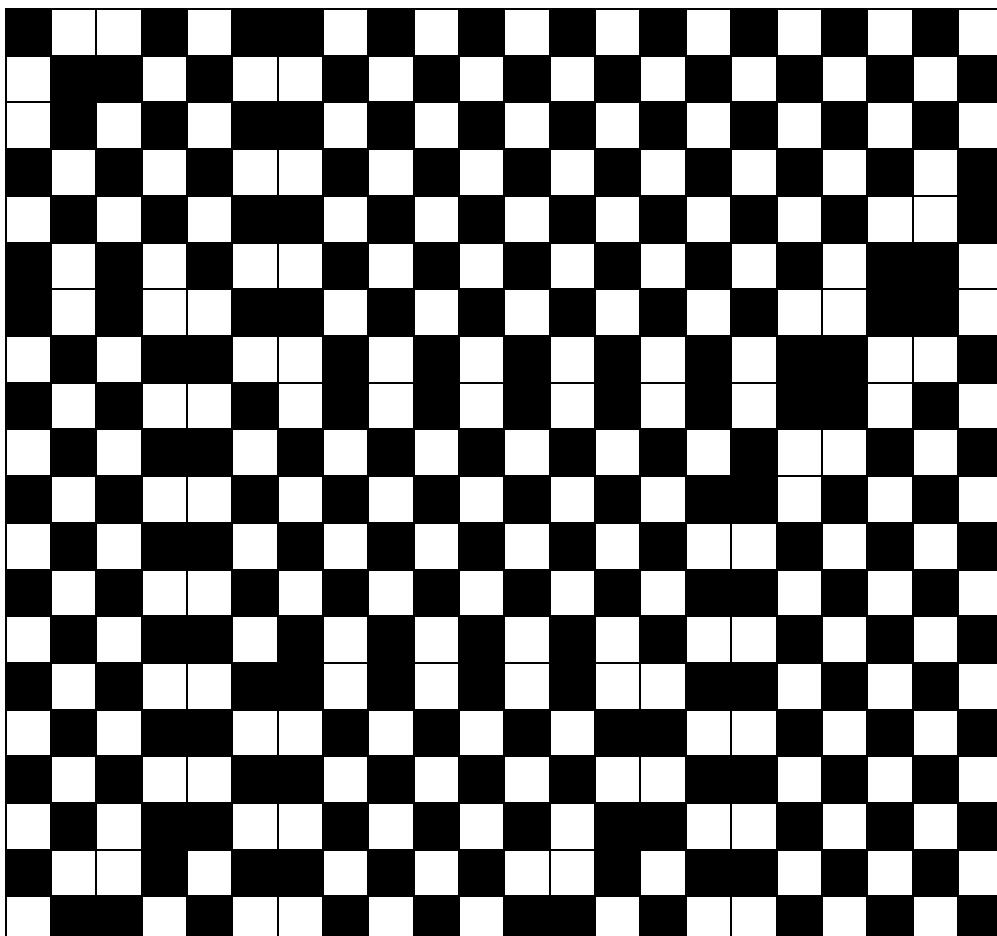
Zu Aufgabe 4

Bitte diese Seiten auf Folie ausdrucken.



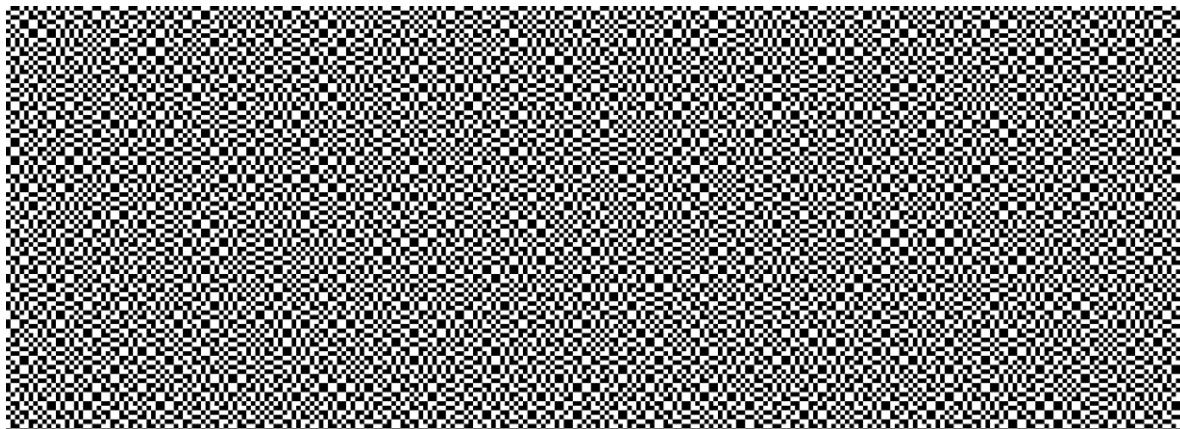
Zu Aufgabe 4

Bitte diese Seiten auf Folie ausdrucken.



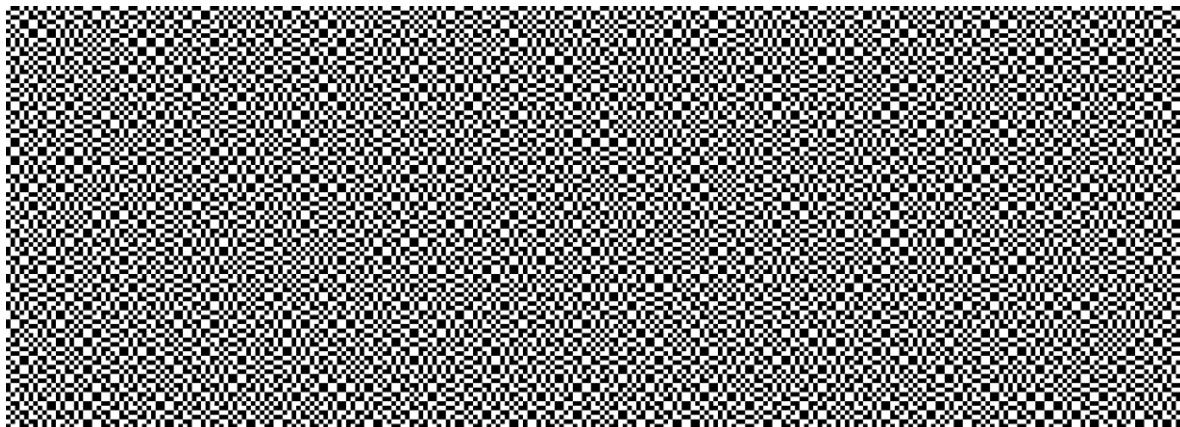
Zu Aufgabe 4

Bitte diese Seiten auf Folie ausdrucken.



Zu Aufgabe 4

Bitte diese Seiten auf Folie ausdrucken.



Steganographie

Um Informationen sicher zu Übertragen ist es manchmal sinnvoll, dass die Nachricht gar nicht erst den Anschein macht, dass es sich hierbei um eine verschlüsselte Nachricht handelt.

Angenommen Alice sitzt nach einem lohnenswerten Geldraub im Gefängnis und möchte Bob eine Nachricht schicken, um ihm den Ort des versteckten Geldes mitzuteilen. Sie könnte zwar diese Nachricht sehr gut verschlüsseln, was zur Folge hat, dass die daraus resultierende Nachricht zwar nicht vom Gefängniswärter Walter entschlüsselt wird, aber vermutlich würde dieser die Nachricht dann auch nicht an Bob weiterleiten. Also schreibt Alice lieber ein (belangloses) Liebesgedicht, bei dem die Anfangs- und/oder Endbuchstaben der Zeilen hintereinander gelesen die eigentliche Nachricht bilden, die dann von Walter wahrscheinlicher an Bob weitergeleitet wird.

Dieses Themenfeld, bei denen die Verschleierung von Informationen im Vordergrund stehen, nennt man Stenographie. Hierbei gibt es viele verschiedene Verfahren, die mit ganz unterschiedlichen Hilfsmittel funktionieren.

Einfache Beispiele

Aus der Geschichte gibt es viele einfache Beispiele, wie Geheimtinte (Zitronensaft) oder doppelte Böden in Briefumschlägen, Schubladen etc. Weitere aufwendigere Verfahren sind zum Beispiel Mikrofilme (bekannt aus alten Agentenfilmen) oder Wasserzeichen (Machine Identification Codes).

Aufgabe 1

Informiere dich über die obigen Beispiele im Internet und fasse diese kurz zusammen.

Linguistische Verfeinerungen

Abseits der verwendeten Hilfsmittel wie Tinte oder doppelte Böden, gibt es auch Verfahren, die mehr mit der Sprache und Codierung ebendieser arbeiten. Ein Beispiel hierfür ist das obige Liebesgedicht von Alice an Bob. Um jedoch auch andere Mittel als nur die gängige Sprache zu nutzen, müssen Informationen codiert werden. Damit wiederum solche Codierungen nicht auffallen, werden diese in sogenannte Semagramme eingebettet. Hierbei handelt es sich um Bilder, Melodien etc. in denen kleine Details, die in Wirklichkeit die codierten Geheiminformationen darstellen, enthalten. Im digitalen Bereich können dies auch ein paar zusätzliche Bytes innerhalb einer MP3 oder eines Bildes sein, die an der ursprünglichen Melodie oder dem Bild nichts ändern.

Aufgabe 2

Betrachte das Beispiel und versuche herauszufinden, wie die Informationen verschleiert wurden. Die verschleierte Nachricht lautete: Um drei am Treffpunkt.



Aufgabe 3

Versuche nun selbst mittels einer beliebigen Methode eine Nachricht bzw. Information zu verschleiern.

Visuelle Kryptographie

Einzelnen betrachtet lässt sich den Folien kein Sinn entnehmen, aber was passiert, wenn man sie übereinander legt? (Wie das Verfahren funktioniert lässt sich am leichtesten mit dem groben händisch erstellten Raster erkennen.)

Aufgabe 4

Versuche zu beschreiben, wie die visuelle Kryptographie funktioniert und erstelle ein eigenes Beispiel auf einer neuen Folie. Du kannst die einfache Karofolie als Gegenstück verwenden.



Musterlösungen

A2.8

Aufgabe 1

GIVUUVM FN WIVR

Aufgabe 2

DER GEIER HAT DIE BEUTE

Aufgabe 3

63 43 51 61 61 51 72 (= TREFFEN)

73 62 (= UM)

41 43 51 22 (= DREI)

Aufgabe 4

DER SCHLUESSEL LIEGT AM FLUSS

A2.9

Aufgabe 1

Die Lösungen sind abhängig von der Anzahl der Flächen, die der Skytale besitzt. Hier mögliche Lösungen:

Anzahl der Flächen	Nachricht
4	AAHTEMAETUUHENBNNSSEADENOTCUBAKTR
5	AUEBMNOUSUEBNRSCTNETTHEDKEAHAAAAN
6	ASTDAOUCLEANRSHAMNTTHBBTAEEEEEUNKN
7	ACABEUBENSHEKOTENARAUDNTUTANSEMT
8	AHEATUHNNSEDNTUATATMEUEBNSAEOCBKR

Aufgabe 4 - BEI MARK UM ACHT UHR

A2.10

Aufgabe 1 - ZVLJNQT BQ PXIP

Aufgabe 2 - DER GEIER HAT DIE BEUTE

A2.11

Aufgabe 1

Schritt 1

PWTMYTBADKDGPWPFYWFGUESOTLUPNVYWAPKCSO **OJWWAS** TLSUZUSJMJBRS
 TIMGPYSX**OJWWAS**MMZQLCHJQWGYDHKO **OJWWAS**TMFPADWIPVKLHONZWPDPWRA
 AG**QPRKNJ**CNPKGJJLTHYOWOHPGYJWCUEKUZLGAOWKHOGPESMRWPBKVF
 ZTQNLAGSFMSVWTDPWRAAG**QPRKNJ**CNPTGTKEOMSGVLYVCHKBVKLOFOBLGNC
 IVXWPLYBZA**AEO**OWKEWEODZKZOGPWGOMSWMP**T**IFFLCTUTYGUOSLZSILYOH
 EWEODSRVVYHSFAVVHHWGIPTGH**YHCWJVLE**RGJWKPDHGJWTUTQNBXGZEUKTW
 IAZPPMOGPWGJQWGYDHKNJCNPSOVWTZPFOMNQUQFGOWPYTQNBAIVOSXNSNZ
 NVHMSPAHCXBWVDTFJRWFLASXAGPH**YHCWJVLE**OANWKUPT**XIYGU**FFSQLLHZR
 KZFGPYTXIYGUOKVAE**O**EAOBBCVOSXVWKUMSGVLYVCHKBGOYOSTSGGUYSTA
 APKYWIPLBRSRIKULYJUVWKUPFHMDKLMWMMFRLCGUVKQSWAGVVWYNVLZSI
 LYROMKKJSBAZSWMOWKHMILSCKZAIRPWZHMGPYSXLWTNCIVXWPIP NOMZGUS
 SXIMUIPYUUEGUKICMDEOPFMZMRPGOMYGOZSBOKLGWKTWHYLUKVEWZDAG
 VEKUOSYBWPZDHKTGUFBJEWNJSSLZSILYYUMFPAPAGVKVLWZKV

Schritt 2

Buchstabenfolge	Abstand vom ersten Buchstaben zum ersten Buchstaben	ggT der Abstände pro Buchstabenfolge
OJWWAS	$28 = 2^2 * 7$ $21 = 3 * 7$	7
TXIYGU	$21 = 3 * 7$	/
YHCWJVLE	$119 = 7 * 17$	/
PRKNJ	$77 = 7 * 11$	/
AEO	$238 = 2 * 7 * 17$	/
PWT	$266 = 2 * 7 * 19$	/

Schritt 3

Buchstabe des Schlüsselwortes	Vorkommende Buchstaben mit ihrer Häufigkeit	Vermuteter Buchstabe des Schlüsselwortes		
1. Buchstabe	P = 21, Y = 12, ...	E	L →	P
2. Buchstabe	S = 26, W = 13, ...	E	O →	S
3. Buchstabe	K = 20, G = 10, O = 10, X = 10, ...	E	G →	K
4. Buchstabe	V = 16, M = 15, Z = 11, ...	E	I →	M
5. Buchstabe	W = 24, J = 10, ...	E	S →	W
6. Buchstabe	G = 25, P = 12, ...	E	C →	G
7. Buchstabe	L = 18, U = 15, ...	E	H →	L

Der entschlüsselte Text lautet:

EINE GRUPPE VON PERSONEN TEILT SICH SO IN DREI GRUPPEN DASS JEDER ZU GENAU EINER GRUPPE GEHOERT DIE ERSTE GRUPPE NENNT SICH DIE WAHREN WEIL SIE JEDE FRAGE WAHRHEITSGEMAESS BEANTWORTET DIE ZWEITE GRUPPE NENNT SICH DIE LUEGNER WEIL SIE JEDE FRAGE FALSCH BEANTWORTET DIE DRITTE GRUPPE NENNT SICH DIE WECHSLER WEIL SIE AUF EINANDERFOLGENDE FRAGEN ABWECHSELND WAHR UND FALSCH BEANTWORTET DABEI IST ABER NICHT FESTGELEGT OB JEWELLS DIE ERSTE FRAGE EINER SERIE VON FRAGEN RICHTIG ODER FALSCH BEANTWORTET WIRD JEDE PERSON ANTWORTET AUF EINE FRAGE NUR MIT JA ODER NEIN FRAGEN DIE NICHT MIT JA ODER NEIN BEANTWORTET WERDEN KOENNEN SIND NICHT ZUGELASSEN VON EINER BELIEBIGEN PERSON SOLL MAN DURCH FRAGEN DIE SICH NUR AUF DIE ZUGEHOERIGKEIT ZU EINER DER GRUPPEN BEZIEHEN HERAUS BEKOMMEN ZU WELCHER GRUPPE SIE GEHOERT WIE VIELE FRAGEN MUSS MAN MINDESTENS STELLEN UND WELCHE FRAGEN KOENNTE MAN STELLEN

IT2School

Gemeinsam IT entdecken



Modul A3 – Programmieren II

Objektorientierte Programmierung mit
Python

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	Objektorientierte Programmierung mit Python	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters.....	4
5	Inhalte des Moduls.....	5
5.1	Python	5
5.2	Entwicklungsumgebung Geany	5
6	Unterrichtliche Umsetzung.....	6
6.1	Grober Unterrichtsplan.....	7
6.2	Stundenverlaufsskizzen	8
6.2.1	Einstieg in Python.....	8
6.2.2	Einführung in die objektorientierte Programmierung.....	8
6.2.3	Entwicklung eines Memory-Spiels.....	9
6.2.4	Vertiefung in die objektorientierte Programmierung.....	10
7	Einbettung in verschiedene Fächer und Themen	11
8	Anschlussthemen.....	11
9	Literatur und Links	12
10	Arbeitsmaterialien	12
11	Glossar.....	13

1 Objektorientierte Programmierung mit Python

Möchte man größere und aufwendigere Programme schreiben, dann ist es zwingend erforderlich, sich auch tiefergehend mit der Programmierung an sich zu beschäftigen.



Die Schülerinnen und Schüler befassen sich im Rahmen dieses Moduls mit der textuellen Programmiersprache Python und steigen dabei mit einfach Beispielen, wie das Nachprogrammieren eines Programms von Scratch nach Python ein. Im Anschluss daran lernen die Schülerinnen und Schüler Grundkonzepte der Programmierung kennen, wie beispielsweise die objektorientierte Programmierung, Vererbung, Datenkapselung und Entwurfsmuster. Alle Inhalte werden praktisch angewendet, so haben Schülerinnen und Schüler die Möglichkeit eigene Spiele mit einer grafischen Oberfläche zu programmieren

Lernfeld/Cluster:	Daten erforschen
Zielgruppe/Klassenstufe:	4. bis 5. Klasse
	6. bis 7. Klasse
	X 8. bis 10. Klasse
	X 11. bis 12. Klasse
Geschätzter Zeitaufwand:	8 – 10 Einzelstunden
Lernziele:	<ul style="list-style-type: none">• Kennenlernen von Python als Programmiersprache.• Überführung einfacher Scratchprogramme in Python.• Prinzip der objektorientierten Programmierung am Beispiel von Python kennenlernen.• Datenkapselung, Vererbung und Entwurfsmuster (speziell MVC) im Zusammenhang mit der objektorientierten Programmierung kennlernen.• Entwicklung und Ausbau von Programme in Python unter Verwendung der objektorientierten Programmierung.• Gestaltung von grafischen Oberflächen in Python mit der Bibliothek <i>Tkinter</i>.
Vorkenntnisse der Schülerinnen und Schüler:	Erforderlich: <ul style="list-style-type: none">• Grunderfahrung in der Programmierung mit Scratch
Vorkenntnisse der/des Lehrenden:	Erforderlich: <ul style="list-style-type: none">• Grunderfahrung in der Programmierung in Scratch Empfohlen: <ul style="list-style-type: none">• Weitere Programmiererfahrung mit anderen Programmiersprachen

Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters	Keine
Sonstige Voraussetzungen:	<p>Erforderlich:</p> <ul style="list-style-type: none"> • Ausreichende Anzahl an Computern • Python und Geany sollten vorinstalliert sein

2 Warum gibt es das Modul?

Dieses Modul baut auf das Modul *B5 – Programmieren* auf. Die Erfahrungen mit der Programmieroberfläche Scratch werden in diesem Modul aufgegriffen und in eine textuelle Programmiersprache übersetzt. Die Schülerinnen und Schüler erhalten damit einen Einblick in die erweiterte Programmierung und deren komplexe Entwicklung.

Der Schwerpunkt dieses Moduls liegt auf der objektorientierten Programmierung, die ein wichtiger Bestandteil der zeitgemäßen und praxisnahen Anwendungsentwicklung ist. Darüber hinaus wird der Gebrauch von Entwurfsmustern aufgezeigt. Dies ermöglicht es für wiederkehrende Probleme und Anforderungen Lösungsschemata zu verwenden.

Zielsetzung dieses Moduls ist es, die Schülerinnen und Schüler für die Programmierung von komplexeren Anwendungen zu motivieren und ihnen einen praxisnahen Einblick zu gewähren. Durch die Umsetzung von eigenen Projekten können die Schülerinnen und Schüler kreative Ideen umsetzen und die Möglichkeit zur Mitgestaltung in der digitalen Gesellschaft erfahren.

3 Ziele des Moduls

- Kennenlernen von Python als Programmiersprache.
- Überführung einfacher Scratchprogramme in Python.
- Prinzip der objektorientierten Programmierung am Beispiel von Python kennenlernen.
- Datenkapselung, Vererbung und Entwurfsmuster (speziell MVC) im Zusammenhang mit der objektorientierten Programmierung kennlernen.
- Entwicklung und Ausbau von Programme in Python unter Verwendung der objektorientierten Programmierung.
- Gestaltung von grafischen Oberflächen in Python mit der Bibliothek *Tkinter*.

4 Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters

Im *Modul A3 – Programmieren II* hat die Unternehmensvertreterin/der Unternehmensvertreter mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

- Unterstützung der Lehrkraft, z. B. bei der Einführung in die Programmierung.

- Unterstützung der Schülerinnen und Schüler bei der Umsetzung ihrer eigenen Projekte.
- Sie oder er kann zur Abschlusspräsentation der Projektergebnisse als Special-Guest eingeladen werden.

5 Inhalte des Moduls

5.1 Python

Bei Python handelt es sich, ähnlich wie bei Scratch aus B5 Programmieren, um eine Programmiersprache. Anders als Scratch handelt es sich bei Python jedoch um eine textuelle Programmiersprache, die universell einsetzbar ist und auf verschiedenen Systemen Anwendung findet. Dies bedeutet, dass die Anweisungen, die vom Programm ausgeführt werden, in Textform (dem sogenannten Quellcode bzw. Sourcecode) verfasst wird. Anschließend kann dieser Quellcode auf jedem System, das über die Programmiersprache (bzw. den Interpreten) Python verfügt, ausgeführt werden.

Für die Entwicklung von Programmen existierten viele verschiedene Umgebungen. In diesem Modul beschränken wir uns auf die Entwicklungsumgebung Geany. Jedoch ist es auch möglich, das Programm mit dem ganz gewöhnlichen Editor (nicht zu verwechseln mit Word oder Wordpad) zu schreiben und dem Standardterminal unter Windows, Linux etc. auszuführen.

Python liegt in zwei Versionen vor und kann von der offiziellen Webseite <http://python.org/download/> ganz einfach als Setup runtergeladen und installiert werden. Wir verwenden die Version 2.7 in diesem Modul.

Im Gegensatz zu anderen Programmiersprachen werden in Python Befehle nicht durch ein Semikolon beendet und Schleifen oder Bedingungen durch geschweifte Klammern eingegrenzt. Im letzten Fall werden stattdessen die entsprechenden Anweisungen eingerückt. Dies soll dafür sorgen, dass der Quellcode im Vergleich zu anderen Sprachen möglichst lesbar und damit anfängerfreundlich ist.

5.2 Entwicklungsumgebung Geany

Geany ist eine Entwicklungsumgebung für verschiedene Programmiersprachen (darunter auch Python) und kann unter Linux, OS X und Windows verwendet werden. Die Entwicklungsumgebung besitzt hilfreiche Funktionen, wie zum Beispiel Übersichten von Methoden oder Auto vervollständigung. Geany lässt sich dabei ganz einfach wie übliche Entwicklungsumgebungen benutzen. Es folgen eine Grafik der Entwicklungsumgebung, inklusiver der Kennzeichnung einiger wichtiger Elemente sowie deren Erläuterung:

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3  #
4  # Handelssimulator_Controller.py
5  #
6  #
7  #
8  from Handelssimulator_View import *
9  from Handelssimulator_Model import *
10
11
12 class Handelssimulator_Controller:
13     ##### Initialisierung des Controllers
14     # Mit dieser Methode wird der Controller initialisiert, das Programm aber noch nicht
15     def __init__(self):
16         # Lade Model und View
17         self.__model = Handelssimulator_Model()
18         self.__view = Handelssimulator_View()
19
20         # Hole Daten aus Model
21         aktuelle_stadt = self.__model.gib_aktuelle_stadt()
22         gold = self.__model.gib_gold()
23         lagerplatz = self.__model.berechne_lagerplatz()
24         anzahl_schiffe = self.__model.gib_anzahl_schiffe()
25         staedte = self.__model.gib_staedte()
26         waren = self.__model.gib_waren()
27         lager = self.__model.gib_lager()
28         preise = self.__model.gib_preise()
29
30         #Aktualisiere die komplette GUI
31         self.__view aktualisiere_status(aktuelle_stadt, gold, lagerplatz, anzahl_schiffe)

```

1. Hier kann eine neue Datei angelegt werden. Wichtig zu erwähnen ist hier, dass verschiedene Formate (.py für Python) als Vorlagen ausgewählt werden können.
2. In diesem Bereich befindet sich der Quellcode des Programms.
3. Hier werden die Klassen und deren Methoden aus der aktuellen Datei aufgelistet.
4. Mit diesem Button kann das Programm ausgeführt werden.
5. In diesem Bereich werden alle Benachrichtigungen in Bezug auf das Kompilieren bzw. Ausführen des Programmes angezeigt. Fehlermeldungen werden grundsätzlich im Terminal des ausgeführten Programmes angezeigt.

6 Unterrichtliche Umsetzung

In diesem Modul steht die selbstständige Arbeit der Schülerinnen und Schüler im Vordergrund. Zu Beginn werden die Schülerinnen und Schüler angeleitet, neue Konzepte zu lernen. Danach können die Schülerinnen und Schülern sich eigenverantwortlich in die Thematik vertiefen und auch Arbeitsaufträge umsetzen.

Als Lehrkraft sind sie daher zum großen Teil als Lernbegleitung im Einsatz. Dies bedeutet, dass sie Materialien und Aufgabenstellungen vorgeben, jedoch bei der Problemlösung nur wenig eingreifen bzw. den Weg zur Lösung nicht direkt vorgeben.



Nach Abschluss des Moduls verfügen die Schülerinnen und Schüler über ein erweitertes Verständnis bzgl. der Programmierung (im Vergleich zu dem Modul B5). Die Schülerinnen und Schüler sind nun in der Lage, eigene Projekte und Ideen zu entwickeln und umzusetzen.

6.1 Grober Unterrichtsplan

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Python sowie Geany werden vorgestellt und die Schülerinnen und Schüler überführen ein Programm von Scratch nach Python.
Erarbeitung	Die Schülerinnen und Schüler machen erste Erfahrungen mit dem Konzept der objektorientierten Programmierung und der Entwicklung von grafischen Oberflächen.
Vertiefung	Die Schülerinnen und Schüler programmieren ohne direkte Anleitung ein Memory mit grafischer Oberfläche.
Erarbeitung	Das Konzept der objektorientierten Programmierung wird vertieft und Themen wie Datenkapselung, Vererbung und Entwurfsmuster werden behandelt.
Vertiefung	Die Schülerinnen und Schüler arbeiten sich in ein vorgegebenes unfertiges Spiel (Handelssimulator) ein und stellen dieses fertig.

6.2 Stundenverlaufsskizzen

Abkürzungen/Legende

AB = Arbeitsblatt/Arbeitsblätter; L = Lehrkraft; MuM = Mitschülerinnen und Mitschüler; SuS = Schülerinnen und Schüler;
UV = Unternehmensvertreterin/Unternehmensvertreter

6.2.1 Einstieg in Python

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Vorstellung von Python und Geany	
25 Min.	Erarbeitung	Einzel-/Partnerarbeit	Die SuS befassen sich mit A3.1 beschäftigen. L steht bei Problemen mit der Entwicklungsumgebung zur Verfügung. Wichtig ist, dass die L nicht direkt Lösungen vorgibt, da die Anleitung zum einen das Machen eines bestimmten Fehlers vorsieht und zum anderen sehr detailliert ist. Zum Abschluss kann der Lehrer auch die Befehlsübersicht A3.8 austeilen.	A3.1, Computer, ggf. A3.8
15 Min.	Abschluss	Plenum	Einige Programme können von den SuS präsentiert und Aufgabe 4 kann besprochen werden. Zum Abschluss kann die L noch einen Überblick über die kommenden Stunden geben.	Computer, Beamer

6.2.2 Einführung in die objektorientierte Programmierung

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Ablauf der Unterrichtsstunde klären	

15 Min.	Erarbeitung	Lehrervortrag oder Einzel-/Partnerarbeit	Einstieg in die objektorientierter Programmierung; Möglich durch Lehrervortrag, oder durch Austeilung und selbstständiges Arbeiten mittels A3.2. Hier sollte deutlich werden, dass die objektorientierte Programmierung wichtig für die kommenden Stunden ist: Bei der Programmierung einer grafischen Oberfläche wird ausschließlich mit Klassen und Objekten gearbeitet!	A3.2
20 Min.	Vertiefung	Einzel-/Partnerarbeit	SuS arbeiten sich in die Grundlage der grafischen Oberflächen mittels des Tutorials A3.3 ein.	A3.3, Computer
5 Min.	Abschluss	Plenum	Im Plenum sollte der Inhalt der Unterrichtsstunde kurz reflektiert und besprochen werden. Ein Ausblick auf die kommende Stunde kann ebenfalls gegeben werden.	

6.2.3 Entwicklung eines Memory-Spiels

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Besprechung des Ablaufs dieser und nächsten Unterrichtsstunde	
70 Min. ¹	Erarbeitung	Einzel-/Partnerarbeit	Die SuS entwickeln selbstständig ein eigenes Memory-Spiel. Als Grundlage erhalten sie das AB A3.4 und das ZIP-Archiv A3.5. Das AB A3.4 bietet ein paar Hilfestellungen für mögliche Probleme, ebenso wie die L als Hilfestellung bereitstehen sollte.	A3.4, A3.5, Computer
15 Min.	Abschluss	Plenum	Zum Abschluss können alle oder einige Programme im Plenum präsentiert werden. Es ist auch möglich, die verschiedenen Programm zum Austausch bereitzustellen.	

¹ Wir gehen hier handlicherweise von einer Doppelstunde aus, jedoch kann der Inhalt auch auf zwei Einzelstunden verteilt werden.

6.2.4 Vertiefung in die objektorientierte Programmierung

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
5 Min.	Einstieg	Plenum	Begrüßung; Ablauf der Unterrichtsstunde klären	
20 Min.	Erarbeitung	Lehrervortrag oder Einzel-/Partnerarbeit	<p>Einstieg in den zweiten Teil der objektorientierter Programmierung; Möglich durch Lehrervortrag, oder durch Austeilen und selbstständiges Arbeiten mittels A3.6.</p> <p>Hier sollte deutlich werden, dass die objektorientierte Programmierung wichtig für die kommenden Stunden ist: Bei der Programmierung einer grafischen Oberfläche wird ausschließlich mit Klassen und Objekten gearbeitet!</p>	A3.6
95 Min. ²	Erarbeitung	Einzel-/Partnerarbeit	Die SuS arbeiten sich in den Handelsimulator ein und erweitern diesen.	A3.7, Computer
15 Min.	Abschluss	Plenum	Zum Abschluss können alle oder einige Programme im Plenum präsentiert werden. Es ist auch möglich, die verschiedenen Programme zum Austausch bereitzustellen.	

² Bei einer starken Klasse könnten auch nur zwei Unterrichtsstunden reichen, jedoch sind drei zu empfehlen.

7 Einbettung in verschiedene Fächer und Themen

Als Einbettung in ein speziellen Unterrichtsfach bietet sich in erster Linie die Informatik an. Dies liegt vor allem in der Tatsache, dass es sich bei diesem Modul um eine Vertiefung der Programmierung handelt. Da gerade die Informatik nicht in allen Bundesländern fester Bestandteil der Schulbildung ist und einige Schulen kein Unterrichtsfach in Richtung der Informatik anbieten, würde es sich als Alternative anbieten, dieses Modul fächerübergreifend im Rahmen einer Projekt- bzw. Themenwoche einzubinden.

Die folgenden Kompetenzen finden sich entweder in den Bildungsstandards der Kultusministerkonferenz oder in den einzelnen Rahmenlehrplänen der Länder wieder:

Informatik

Die Schülerinnen und Schüler ...

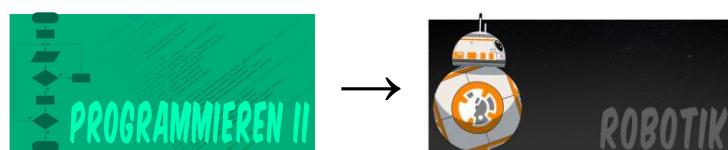
- erstellen Produkte unter Anwendung fortgeschrittenen Techniken von Standardsoftware; falls es das Produkt erfordert, arbeiten sie sich in geringem Umfang in Spezialsoftware ein.
- erwerben beim Bearbeiten von Softwareprojekten in angemessenem Umfang Kenntnisse über Analyse- und Modellierungsverfahren sowie Projektmanagement.
- verstehen Programmabläufe und die Arbeitsweise von Schnittstellen.
- können erdachte Systeme in technische Systeme übertragen.
- kennen sich in Entwicklungsumgebungen/Programmierumgebungen aus.
- eine Spielumgebung entsprechend der Spielidee gestalten.
- eine Spielvariante mit informatischen Werkzeugen entwerfen und implementieren.

8 Anschlussthemen

Als Anschlussthemen im Zusammenhang mit IT2School bietet sich folgendes Modul an:

Beispiel: Ein Einstieg in eine weitere textuelle Programmiersprache

Im Erweiterungsmodul E3 Robotik haben die Schülerinnen und Schüler die Möglichkeit einen eigenen BB8 (ein Roboter aus dem Star Wars Universum) zu bauen. Bei diesem Bauvorhaben kann auf den Arduino zurückgegriffen werden. Dies bietet die Möglichkeit sich nochmals einer anderen textuellen Programmiersprache zu widmen. Ein wichtiger Unterschied im Vergleich zu der bisherigen Programmierung ist dann, dass mit der dort verwendeten Programmiersprache unter anderem Motoren angesteuert werden können.



Beispiel: Verschiedene Vertiefungsmöglichkeiten

Verfügen die Schülerinnen und Schüler erst einmal über ein gewisses Potenzial in der Programmierung, so kann hierauf hinsichtlich verschiedener Vertiefungsmöglichkeiten aufgebaut werden. So wäre es beispielsweise möglich sich intensiver mit dem Entwicklungsprozess bzw. der Planung und Konzeption von Anwendungen zu beschäftigen. Schülerinnen und Schüler könnten sich in dem Fall in dem Bereich von UML-Diagrammen oder Ähnlichem vertiefen. Eine weitere Vertiefungsmöglichkeit stellt auch das Thema Datenbanken dar. Hier könnten die Schülerinnen und Schüler eigene Datenbanken für gewisse Aufgabenstellungen entwerfen und anschließend in ihre Programme integrieren.

9 Literatur und Links

- Glöde, Martina & Reit, Birgit (Hg.) (2015): **Programmieren supereasy. Einfacher Einstieg in Scratch und Python.** München: Dorling Kindersley.
- Online-Kurs bzw. -Tutorial zu den Grundlagen von Python – online: <http://www.python-kurs.eu/kurs.php>
- **Online-Kurs bzw. -Tutorial zu den Grundlagen der Gestaltung von grafischen Oberflächen mit Tkinter in Python** – online: <http://www.tkdocs.com/tutorial/index.html>
- **Lernplattform für Python** sowie die Auswertung von Daten mittels Python – online: <https://www.datacamp.com>

10 Arbeitsmaterialien

Nr.	Titel	Beschreibung
😊 A3.1	Einstieg	Bietet einen Einstieg in Python, wobei ein Programm/Spiel von Scratch in Python überführt wird.
😊 A3.2	Objektorientierte Programmierung – Teil 1	Einstieg in die objektorientierte Programmierung. Erklärt, was es ist und gibt Beispiele zur Verwendung von Klassen und Objekten.
😊 A3.3	Einstieg grafische Oberflächen	Vorstellung der Bibliothek Tkinter zur Gestaltung von grafischen Oberflächen. Enthält auch ein Beispiel Programm als Tutorial.
😊 A3.4	Memory	Aufgabenstellung zur Gestaltung eines eigenen Memorys. Beinhaltet auch Problemstellungen inklusive deren Lösungen.
😊 A3.5	Memory – Vorlage	Vorlage für Schülerinnen und Schüler zur Aufgabenstellung in A3.4.
😊 A3.6	Objektorientierte Programmierung – Teil 2	Der zweite Teil der objektorientierten Programmierung wird vorgestellt. Inhaltlich umfasst dieses Material die Vererbung, Datenkapselung und das Entwurfsmuster MVC.



 A3.7	Handelssimulator	Aufgabenstellung zur Einarbeitung in eine Rohvariante des Handelssimulator sowie der zustzlichen Aufgabenstellung zur Erweiterung dieses Programms/Spiels.
 A3.8	Befehlsübersicht	Eine Gegenüberstellung einiger Bausteine aus Scratch zu den Befehlen aus Python.
 A3.9	Installationsanleitung	Kurze Einführung in Python und Geany sowie eine Installationsanleitung.

Legende

-  Material für Schülerinnen und Schüler
-  Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
-  Zusatzmaterial

11 Glossar

Begriff	Erläuterung
DatenkapSELung	Bezeichnet die Zugriffsbegrenzung von Attribute (und damit auch werden Daten/werten). Bei der Definition einer Klasse können deren Attribute als öffentlich, privat oder geschützt gekennzeichnet werden.
Entwurfsmuster	Als Entwurfsmuster werden Lösungsschablonen zu wiederkehrenden Entwicklungsproblemen bezeichnet. Dies kann zum Beispiel das Problem bzw. der Wunsch sein, dann bei der Entwicklung einer Anwendung die Logik, die Datenquellen sowie die grafischen Oberflächen im Quellcode von einander getrennt werden.
Geany	Eine Entwicklungsumgebung für mehrere Programmiersprachen, so auch für Python.
MVC	Ein Entwurfsmuster, dann bei der Entwicklung einer Anwendung zum Einsatz kommt, wenn die Logik, die Datenquellen sowie die grafischen Oberflächen im Quellcode von einander getrennt werden sollen.
Objektorientierte Programmierung	Ein Paradigma der Programmierung, bei dem Objekte und Klassen als Grundstrukturen der Wirklichkeit in der Konzeption von Anwendungen entworfen und verwendet werden.
Scratch	Eine blockbasierte Programmiersprache, welche sehr gut für junge Schülerinnen und Schüler als Einstieg in die Programmierung geeignet ist.
Vererbung	Ein grundlegendsten Konzepte der objektorientierten Programmierung, welche Beziehungen zwischen

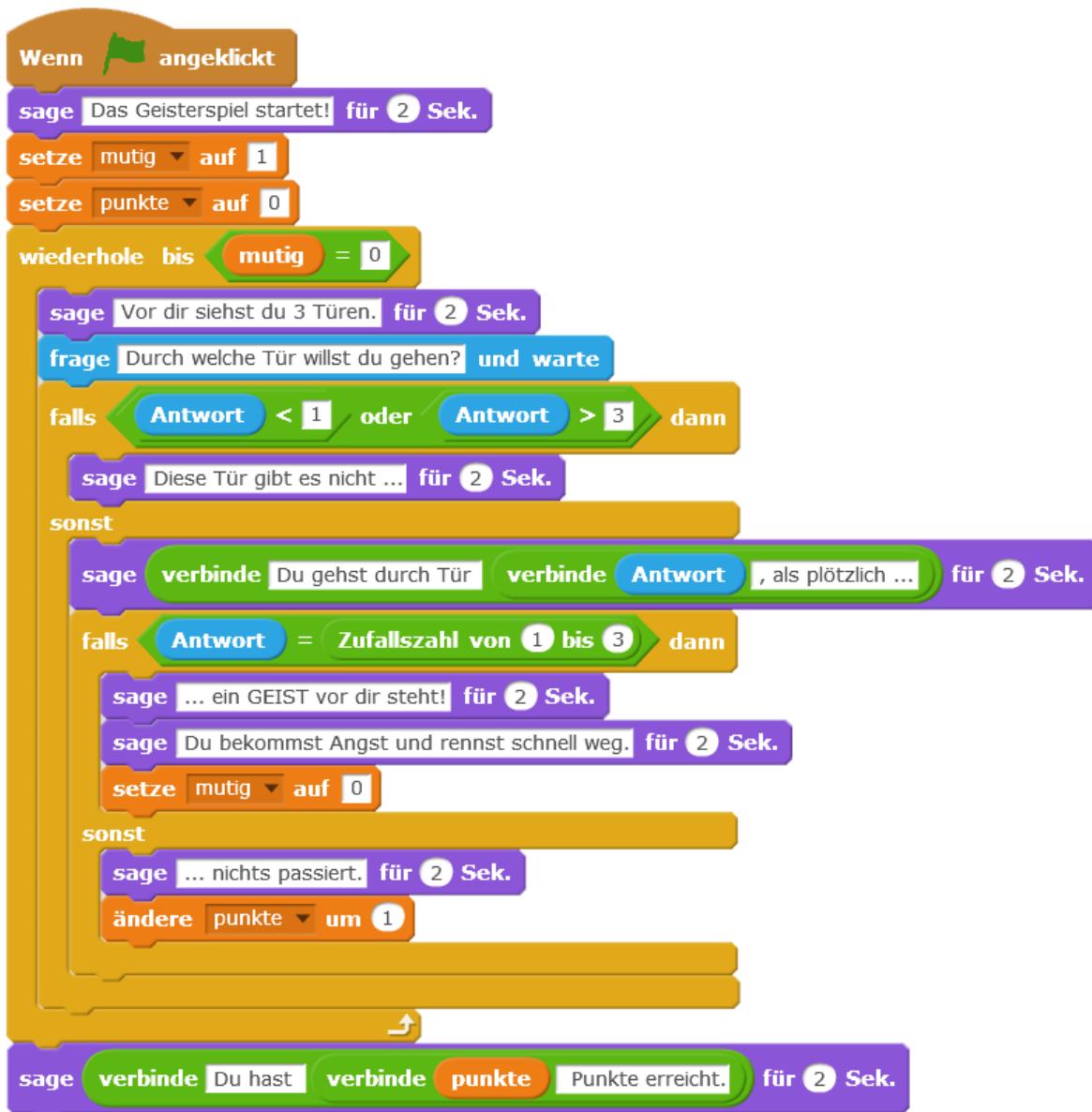


Klassen/Objekten zulässt. Methoden und Attribute können dabei von der Mutterklasse auch die abgeleiteten Klasse vererbt werden.



Das Geisterspiel

Zum Einstieg in Python, wollen wir zunächst ein kleines Spiel programmieren. Dieses Spiel handelt von mehreren Türen und einem Geist. Damit du weißt, was genau auf dich zu kommt, kannst du dir das folgende Scratch-Skript anschauen. Hier wurde das Spiel bereits einmal programmiert.



Aufgabe 1

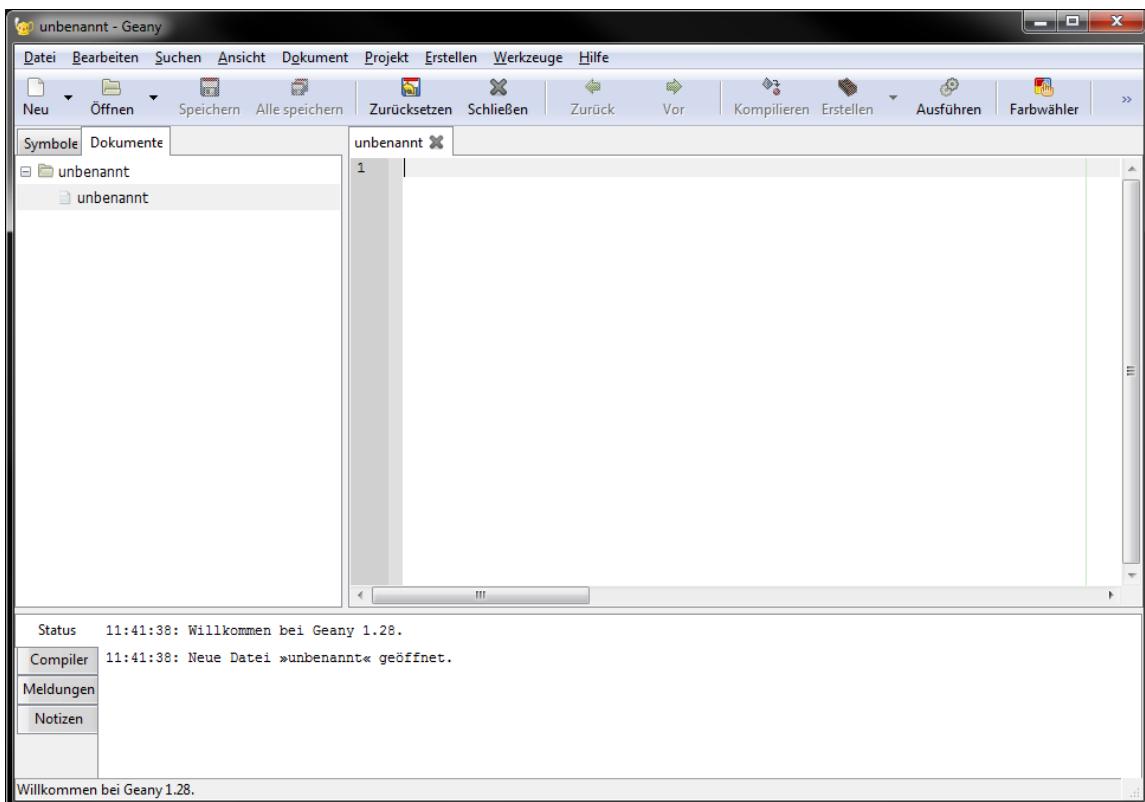
Schau dir das obige Skript genau an und versuche die einzelnen Abschnitte zu verstehen. Beschreibe kurz für dich den Ablauf des Spiels.

Aufgabe 2

Arbeite die folgenden Schritte sorgfältig durch.

Schritt 1: Die Entwicklungsumgebung starten

Zunächst einmal musst du die Entwicklungsumgebung für Python starten. Sofern dir dein Lehrer nichts anderes gesagt hat, startest du das Programm *Geany*. Du findest es entweder auf dem *Desktop* oder im *Startmenü* und *Alle Programme*. Das Fenster sieht dann ungefähr so aus:



Schritt 2: Die Programmierung

Wir werden nun die Bausteine aus dem Scratch-Skript Baustein für Baustein von oben nach unten durchgehen. Dabei kannst du immer wieder vergleichen, welcher Befehl in Python zu dem Baustein in Scratch passt.

Den ersten Baustein *Wenn grüne Fahne angeklickt* können wir ignorieren, daher starten wir erst einmal mit den weiteren drei Bausteinen. Schreibe folgendes (**ohne** die Zeilennummern) in *Geany* ab:

```
01 print("Das Geisterspiel startet!")
02 mutig = 1
03 punkte = 0
```

Anmerkung: Der Befehl *print* sorgt dafür, dass Ausgaben auf dem Bildschirm angezeigt werden.
Das Setzen von Varianten ist in Python auch ganz einfach. Du kannst einfach die Variable gleich ihrem Wert setzen.

Bei dem nächsten Baustein handelt es sich um eine Schleife, die solange durchlaufen wird, wie der Spieler mutig ist. Anschließend soll dem Spieler gesagt werden, dass er vor drei Türen steht. In Python sieht dies dann so aus:

```
04 while(mutig == 1):
05     print("Vor dir siehst du 3 Türen.")
```



Anmerkung: Bei vergleichenden Aussagen (wie zum Beispiel ist mutig gleich 1), wird immer ein doppeltes Gleichheitszeichen verwendet. Damit unterscheidet man Zuweisungen von Werten (einfaches Gleichheitszeichen) von Vergleichungen (doppeltes Gleichheitszeichen).

Sehr wichtig ist, dass in Python die Zeilen innerhalb einer Schleife oder Bedingung immer eingerückt werden. Sehen kannst du dies auch in Scratch, wo die weiteren Bausteine auch etwas nach rechts rücken. Drücke in Geany dazu einfach die Tabulator Taste.

Nun soll der Spieler aufgefordert werden, sich für eine Tür zu entscheiden. In Python verwenden wir hierfür:

```
06     Antwort = input("Durch welche Tür willst du gehen?")
```

Anmerkung: Eingabeaufforderungen werden in Python mit dem Befehl *input* eingeleitet. Vor dem Befehl muss die erwartete Eingabe einer Variable zugewiesen werden. Dabei muss diese Variable nicht zwingend Antwort heißen.

Nun erfolgt zum ersten Mal etwas typisches für Python. Bei Scratch musstest du dir keine Gedanken um Datentypen von Variablen machen. Du konntest ganz einfach die Zahl 1 mit der Eingabe 1 bzw. dem Text bestehend aus der Ziffer 1 vergleichen. In Python ist dies nicht so einfach möglich. Eingaben werden immer als Strings bzw. Texte behandelt. Das bedeutet, dass bei einem Vergleich die Eingabe nicht direkt mit der Zahl 1 verglichen werden kann. Um dies doch zu tun, müssen wir die Eingabe in einen Integer bzw. ganze Zahl umwandeln.

Der folgende Quellcode zeigt dir, wie ein *falls-dann*-Baustein in Python aussehen muss:

```
07     if((int(Antwort) < 1) or (int(Antwort) > 3)):
08         print("Diese Tür gibt es nicht ...")
09     else:
10         print("Du gehst durch Tür " + Antwort + ", als plötzlich ...")
```

Anmerkung: Mit dem *int* Befehl kann man zum Beispiel aus einem String einen Integer machen. Um im *print* Befehl mehrere Strings (Texte, Variablen) zu verketten, kann man einfach das + verwenden.
Bei Python ist immer auf den Datentyp von Variablen zu achten. Wäre zum Beispiel die Variable *Antwort* kein String, so wäre das Verketten nicht so direkt möglich. (Dazu später mehr.)

Weiter geht es mit der Frage, ob hinter der ausgewählten Tür sich nun der Geist befindet. Es gibt also wieder einen *falls-dann*-Baustein mit entsprechenden Ausgaben und Variablenzuweisungen, der in Python dann so aussieht:

```
11     if((int(Antwort) == randint(1, 3))):
12         print("... ein GEIST vor dir steht!")
13         print("Du bekommst Angst und rennst schnell weg.")
14         mutig = 0
15     else:
16         print("... nichts passiert.")
17         punkte = punkte + 1
```

Anmerkung: In Python werden *falls-dann*-Bausteine zu *if-else* Anweisungen.
Der Befehl *randint* ermittelt eine ganzstellige Zufallszahl. Bei dem obigen Beispiel wird eine Zufallszahl zwischen 1 und 3 ermittelt.
Mit dem *int* Befehl kann man zum Beispiel aus einem String einen Integer machen. Um im *print* Befehl mehrere Strings (Texte, Variablen) zu verketten, kann man einfach

das + verwenden.

Bei Python ist immer auf den Datentyp von Variablen zu achten. Wäre zum Beispiel die Variable *Antwort* kein String, so wäre das Verketten nicht so direkt möglich. (Dazu später mehr.)

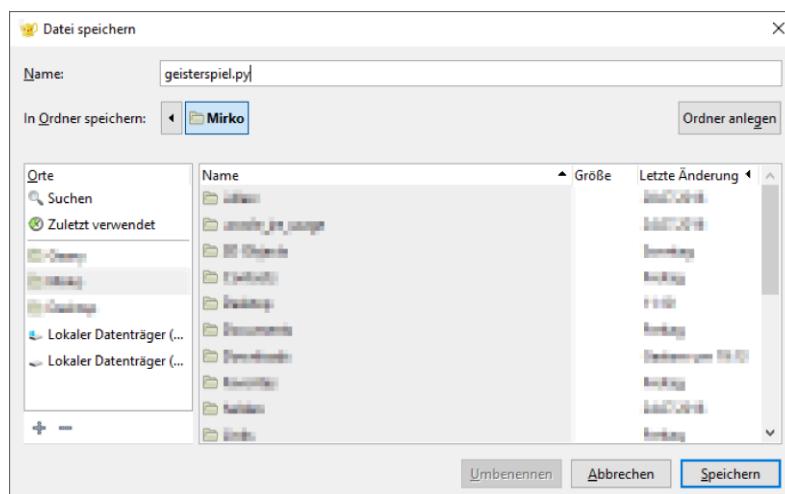
Am Ende des Programms, sobald der Spieler nicht mehr mutig ist, sollen noch die Punkte angezeigt werden:

```
18 print("Du hast " + str(punkte) + " Punkte erreicht.")
```

Anmerkung: Auch hier ist auf den Datentyp der Variable *punkte* zu achten. Da die Variable ein Integer ist (durch die Zuweisung von 0 am Anfang), müssen wir diese für die Ausgabe kurzzeitig zu einem String machen. Dies geht mittels *str* Befehl.

Schritt 3: Erste Test mit vorprogrammiertem Crash

Speichere zunächst dein Programm ab. Klicke dazu auf Speichern, wähle einen passenden Speicherort und achte beim Dateinamen auch auf das richtige Dateiformat für Python. Als Dateinamen empfehlen wir *geisterspiel.py*.



Sobald du das Programm abgespeichert hast, sollte dir auffallen, dass deine Befehle im Programm farbig sind. Geany markiert dir so bestimmte Befehle aus Python, damit du dich besser im Quellcode orientieren kannst. Wenn du fertig bist, sollte das ganze so aussehen:

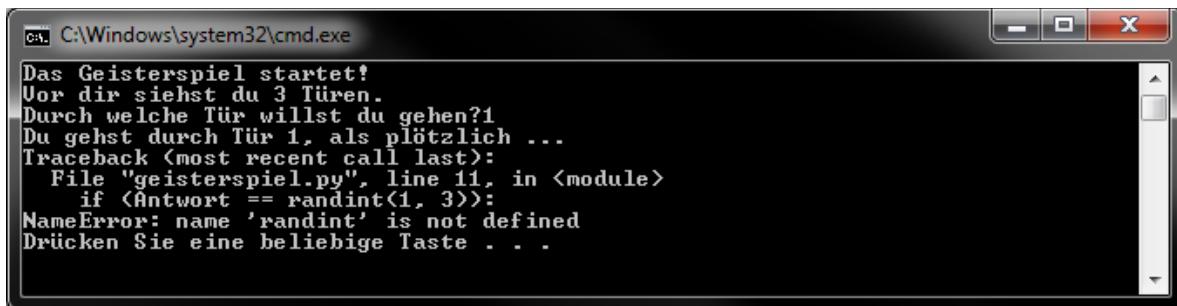
```

1  print ("Das Geisterspiel startet!")
2  mutig = 1
3  punkte = 0
4  while (mutig == 1):
5      print ("Vor dir siehst du 3 Türen.")
6      Antwort = input("Durch welche Tür willst du gehen?")
7      if (int(Antwort) < 1) or (int(Antwort) > 3):
8          print ("Diese Tür gibt es nicht ...")
9      else:
10         print ("Du gehst durch Tür " + str(Antwort) + ", als plötzlich ...")
11         if (Antwort == randint(1, 3)):
12             print ("... ein GEIST vor dir steht!")
13             print ("Du bekommst Angst und rennst schnell weg.")
14             mutig = 0
15         else:
16             print ("... nichts passiert.")
17             punkte = punkte + 1
18     print ("Du hast " + str(punkte) + " Punkte erreicht.")
19

```



Nun wollen wir das ganze aber auch einmal testen. Du musst auf Ausführen klicken, um dein Programm zu starten.



A screenshot of a Windows Command Prompt window titled 'cmd' with the path 'C:\Windows\system32\cmd.exe'. The window displays the following text:

```
Das Geisterspiel startet!
Vor dir siehst du 3 Türen.
Durch welche Tür willst du gehen?1
Du gehst durch Tür 1, als plötzlich ...
Traceback (most recent call last):
  File "geisterspiel.py", line 11, in <module>
    if (Antwort == randint(1, 3)):
NameError: name 'randint' is not defined
Drücken Sie eine beliebige Taste . . .
```

Beim Testen solltest du merken, dass dein Programm direkt nach der Eingabe nicht mehr funktionieren sollte. Die Fehlermeldung sollte wie folgt lauten:

```
Traceback (most recent call last):
  File "geisterspiel.py", line 11, in <module>
    if (Antwort == randint(1, 3)):
NameError: name 'randint' is not defined
```

Ganz unten steht dabei die eigentliche Fehlermeldung. Der Befehl `randint` ist nicht definiert. Das heißt, dass Python damit gar nichts anfangen kann. (Die Lösung dafür siehst du gleich.) Die Zeilen darüber helfen dir, den Ursprung des Fehlers herauszufinden. In der zweiten Zeile steht, wo der Fehler aufgetreten ist: In der Datei `geisterspiel.py`, in der Zeile 11. Darunter wird dir in der dritten Zeile der Quellcode der fehlerhaften Zeile angezeigt.

Fehler treten beim Programmieren immer wieder auf und sind generell auch nichts schlimmes. Wichtig ist nur, sich die Fehlermeldung genau anzuschauen und den Fehler so aufsichtig zu machen. Sollte dir das mal nicht direkt gelingen, kann es auch helfen, eine Suchmaschine zu fragen.

Schritt 4: Fehler beheben

Wie bereits erwähnt, kennt Python den Befehl `randint` noch nicht. Das lässt sich schnell ändern und ist auch nicht ungewöhnlich. Python beinhaltet nicht immer alle Befehle, da sonst die Programme zu groß wären, wenn für jede Eventualität jeder Befehl eingebunden wäre. Daher, müssen manche Befehle erst noch importiert bzw. bekannt gemacht werden.

In unserem Fall heißt das, dass wir folgenden Quellcode an den Anfang unseres Programms hinzufügen müssen:

```
from random import randint
```

Das ganze Programm sollte nun wie folgt aussehen:

```
01 from random import randint
02
03 print("Das Geisterspiel startet!")
04 mutig = 1
05 punkte = 0
06 while(mutig == 1):
07     print("Vor dir siehst du 3 Türen.")
08     Antwort = input("Durch welche Tür willst du gehen?")
09     if((int(Antwort) < 1) or (int(Antwort) > 3)):
10         print("Diese Tür gibt es nicht ...")
11     else:
12         print("Du gehst durch Tür " + Antwort + ", als plötzlich ...")
```



```

13     if((int(Antwort) == randint(1, 3))):
14         print("... ein GEIST vor dir steht!")
15         print("Du bekommst Angst und rennst schnell weg.")
16         mutig = 0
17     else:
18         print("... nichts passiert.")
19     punkte = punkte + 1
20 print("Du hast " + str(punkte) + " Punkte erreicht.")

```

Schritt 5: Spiel testen

Du kannst nun dein Spiel testen. Das Ergebniss könnte ungefähr so aussehen:

```

C:\Windows\system32\cmd.exe
Das Geisterspiel startet!
Vor dir siehst du 3 Türen.
Durch welche Tür willst du gehen?1
Du gehst durch Tür 1, als plötzlich ...
... nichts passiert.
Vor dir siehst du 3 Türen.
Durch welche Tür willst du gehen?2
Du gehst durch Tür 2, als plötzlich ...
... nichts passiert.
Vor dir siehst du 3 Türen.
Durch welche Tür willst du gehen?3
Du gehst durch Tür 3, als plötzlich ...
... nichts passiert.
Vor dir siehst du 3 Türen.
Durch welche Tür willst du gehen?1
Du gehst durch Tür 1, als plötzlich ...
... nichts passiert.
Vor dir siehst du 3 Türen.
Durch welche Tür willst du gehen?1
Du gehst durch Tür 1, als plötzlich ...
... ein GEIST vor dir steht!
Du bekommst Angst und rennst schnell weg.
Du hast 5 Punkte erreicht.
Drücken Sie eine beliebige Taste . . .

```

Aufgabe 3

Mache nun selbst kleinere Änderungen am Spiel. Füge zum Beispiel weitere Türen hinzu oder sorge dafür, dass der Text etwas leserlicher ist.

Anmerkung: Möchtest du im *print* Befehl einen Zeilenumbruch machen, dann musst du an der entsprechenden Stelle ein \n einfügen, wie im folgenden Beispiel gezeigt:
`print("Zeile 1\nZeile2")`

Aufgabe 4

In der folgenden Tabelle siehst du links, die Bausteine aus Scratch. Schreibe in die rechten Spalten den entsprechenden Befehl in Python.

Scratch	Python



Arbeitsmaterial A3.1

<code>ändere zahl um 1</code>	
<code>setze zahl auf Zufallszahl von 1 bis 10</code>	
<code>falls zahl = 5 dann</code> <code>sage Hallo Welt!</code> <code>sonst</code> <code>sage Auf Wiedersehen!</code>	



Objektorientierte Programmierung

Die objektorientierte Programmierung ist ein wichtiges Konzept in der modernen Programmierung. Doch in wie weit unterscheidet sie sich von der Programmierung in Scratch? Die Antwort ist, dass sie sich nicht wirklich davon unterscheidet. Trotzdem ist es wichtig zu wissen, was Objekte und Klassen sind und was diese mit der objektorientierten Programmierung zu tun haben.

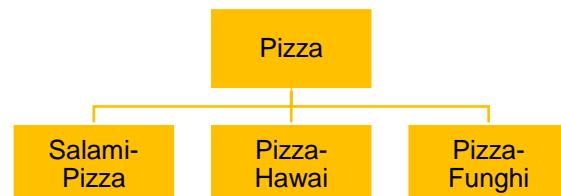
Ein Beispiel aus dem Alltag

Schauen wir uns doch mal die folgenden drei Pizzen an:



Sie haben alle einen unterschiedlichen Belag, aber bei allen handelt es sich um eine Pizza. Für eine Pizza lässt sich festlegen, dass jede Pizza einen Grundteig (z. B. normaler Hefeteig oder Vollkornteig) und verschiedene Zutaten für den Belag besitzt. Zusätzlich gibt es noch bestimmte Tätigkeiten, die mit einer Pizza möglich sind, wie zum Beispiel bestellen, backen, schneiden oder essen. Doch was hat das mit Objekten oder Klassen zu tun?

Diese Frage ist ganz einfach zu beantworten. Die verschiedenen Varianten von Pizzen sind unsere Objekte und diese lassen sich alle als Pizza klassifizieren. Oder anders gesagt: Pizza bildet eine Klasse und die tatsächliche Umsetzung einer Pizza, zum Beispiel die Salami-Pizza, ist ein Objekt dieser Klasse.



Bitte einmal Pizza mit Python ...

Doch wie sieht das ganze in Python aus und was kann man damit machen? Eine Umsetzung des Pizza-Beispiels kann wie folgt aussehen:

```

01 class Pizza:
02     def __init__(self):
03         self.teig = 'Hefeteig'
04         self.zutaten = ['Tomatensosse', 'Käse']
05         self.rand = 'ohne'
06
07     def bestellen(self):
08         pass
  
```

Klassen werden in Python mit dem Befehl `class` eingeleitet. Innerhalb von Klassen wird der Quellcode so eingerückt, wie es bei Schleifen oder Bedingungen bekannt sein sollte. Die Methode `__init__` wird ausgeführt, sobald ein Objekt dieser Klasse initialisiert (erzeugt) wird. Im obigen Beispiel wird der Teig standardmäßig auf Hefeteig, der Rand auf ohne und die Zutaten

auf Tomatensoße und Käse gesetzt. Dies soll die Vorlage für jede Pizza sein (dies kann sich aber bei einzelnen Objekten ändern). Wichtig ist die Verwendung von `self`. Bei der Definition der Methode `__init__` (z.B. Zeile 2) wird es innerhalb der Klammern eingebunden. Dies ist wichtig, da durch `self` gekennzeichnet wird, dass die daran anschließende Variable (siehe z. B. Zeile 3) zu den einzelnen Objekten gehört. Tut man dies nicht, so könnte es sich um eine beliebige globale Variable handeln. Der Befehl `pass` wird benutzt, wenn eine Methode (innerhalb von Klassen nennt man Funktionen Methoden) nicht detailliert beschreiben möchte.

Wie bereits erwähnt ist die objektorientierte Programmierung nicht viel anders als das bekannte Programmieren in Scratch. In Scratch stellten die einzelnen Sprites/Figuren die Objekte und Klassen dar, wobei Objekt und Klasse hier sehr eng verbunden waren. Das eigentliche Scratch-Programm hatte mehrere Sprites bzw. Klassen, welche miteinander interagiert haben.

Wo liegt der Nutzen?

An dieser Stelle ist die Frage nach dem Nutzen sehr berechtigt. Ein sehr großer Vorteil, den du auch in den kommenden Unterrichtsstunden noch sehen wirst, ist, dass Variablen und Methoden sinnvoll zusammengepackt werden. Schau dir dieses Beispiel einmal an:

```

01  class Pizza:
02      def __init__(self): ← Wichtig ist, auf das self zu achten.
03          self.teig = 'Hefeteig'
04          self.zutaten = ['Tomatensosse', 'Kaese']
05          self.rand = 'ohne'
06
07      def bestellen(self):
08          print('Pizza mit ' + self.teig + ' und ' + self.rand + \
09              '-Rand wird bestellt.')
10         print('Zutaten-Liste:')
11         print(self.zutaten)
12
13     def vergleiche_mit(self, andere_pizza):
14         pass ← Die Methode definieren wir nicht vollständig und nutzen daher das pass.
15
16 pizza1 = Pizza() ← Objekte werden initialisiert.
17 pizza1.zutaten.append('Salami')
18 pizza1.zutaten.append('Schinken')
19 pizza1.rand = 'Käse'
20
21 pizza2 = Pizza() ← Die Methode ist zwar nicht genau definiert, aber man erkennt, dass auch Objekte übergeben werden können.
22 pizza2.teig = 'Vollkornteig'
23 pizza2.zutaten.remove('Tomatensoße')
24 pizza2.zutaten.append('Currysoße')
25 pizza2.zutaten.append('Ananas')
26 pizza2.zutaten.append('Hähnchenfleisch')
27 pizza2.rand = 'mit'
28
29 pizza1.bestellen()
30 pizza2.bestellen()
31 pizza1.vergleiche_mit(pizza2) ← Dies hat den Vorteil, dass nicht alle Variablen des Objekt an dessen Stelle übergeben werden müssen.

```

Im obigen Beispiel sieht man, wie zwei Objekte der Klasse `Pizza` initialisiert (erzeugt) und angepasst werden. Dabei sind die Eigenschaften der beiden Pizzen voneinander getrennt und beeinflussen sich gegenseitig nicht. Gerade in größeren Programmen ist es wesentlich einfacher mit Objekten zu arbeiten, als ohne. Neben diesem Vorteil gibt es aber noch weitere Vorteile, auf die wir jedoch erst später im Unterricht eingehen werden. Übrigens kann man mit \ einen Zeilenumbruch im Quellcode anzeigen (siehe Zeile 8).

Grafische Oberflächen

Mit Python lassen sich auch recht simpel grafische Oberflächen zu den Programmen erstellen. Dabei kann man auf die Funktionen der Bibliothek (so nennt man eine größere Menge von zusammengehörenden Klassen) *Tkinter* zurückgreifen. Das folgende Beispiel zeigt dir, wie du ein einfaches Fenster mit einem *Hello Welt* - Text unter Verwendung von *Tkinter* erstellst und anzeigen lässt:

```
from Tkinter import *
fenster = Tk()
text = Label(fenster, text="Hello Welt")
text.pack()
fenster.mainloop()
```

In den folgenden Schritten wirst du dein erstes kleines Programm mit grafischer Oberfläche (GUI) programmieren und sollst die Grundlagen der GUI-Programmierung kennenlernen.

Schritt 1: Grundgerüst

Zunächst einmal musst du die Entwicklungsumgebung Geany starten und ein neues Python Programm öffnen. Anschließend kannst du den folgenden Quellcode abschreiben:

```
# Laden der Bibliothek
from Tkinter import *

# Fenster initialisieren und Titel setzen
fenster = Tk()
fenster.title("Additionsrechner")

# Fenster ausfuehren
fenster.mainloop()
```

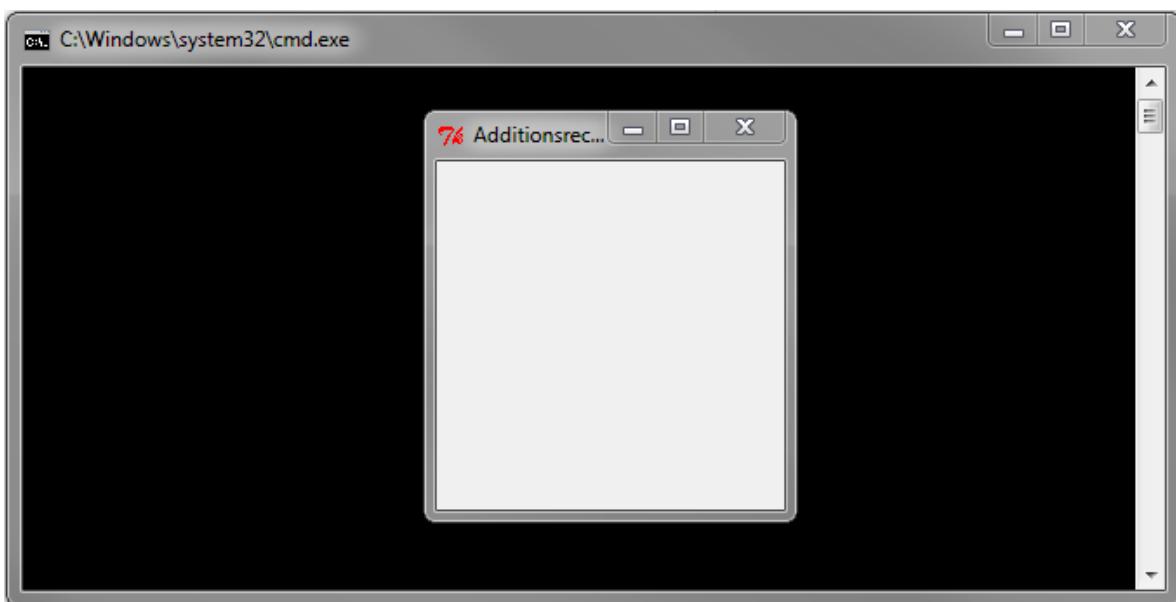
Dies soll zunächst einmal das Grundgerüst für den Additionsrechner sein. Damit grafische Oberflächen erzeugt werden können, muss die Bibliothek *Tkinter* importiert werden (siehe Zeile 2).

Hinweis: Um uns etwas Aufwand zu sparen, importieren wir direkt alles aus dieser Bibliothek durch *from ... import **. Ansonsten müssten wir vor allen Klassen aus dieser Bibliothek immer *Tkinter.* schreiben.

Mit *Tk()* wird das Fenster initialisiert und mit der Methode *title()* lässt ein ein Titel für das Fenster festlegen. Die Methode *mainloop()* sorgt dafür, dass das Fenster angezeigt wird und alle damit verbundenen Programmlogiken ausführbar sind.

Wenn du das Programm ausführst, solltest du folgendes angezeigt bekommen. (Die Console, das Fenster mit dem schwarzen Hintergrund, wird immer mitgeöffnet.)





Schritt 2: Die ersten Komponenten

Komponenten wie Labels (Textbausteine), Eingabefelder oder Buttons müssen vor dem Aufruf von `mainloop()` zum Fenster hinzugefügt werden. Zunächst wollen wir das erste Eingabefeld hinzufügen. Füge dafür folgenden Quellcode nach der Initialisierung des Fenster ein:

```
# Erstes Eingabefeld hinzufuegen
zah1_entry = Entry(fenster, width=5, justify="center")
zah1_entry.grid(row=0, column=0)
```

Die Klasse `Entry` steht für typische Eingabefelder. Bei der Initialisierung `Entry(...)` muss der erste Parameter immer die Komponente sein, in die das Eingabefeld eingebunden wird (in diesem Fall das Fenster). Anschließend können weitere Parameter frei übergeben werden (in diesem Fall die Breite (`width`) und Textausrichtung (`justify`)).

Zum Schluss muss noch angegeben werden, wo genau das Eingabefeld positioniert werden soll. Hierfür gibt es 3 verschiedene Methoden: `pack()`, `grid()` und `place()`. Mit der `pack`-Methode können Elemente grob durch Angabe einer Richtung (left, right, top, down) angeordnet werden. Mit der `grid`-Methode unterteilt man die grafische Oberfläche in eine tabellenartige Struktur. Hier lässt sich ein Element unter Angabe von Zeile (`row`) und Spalte (`column`) anordnen. Mit der `pack`-Methode können Elemente pixelgenau auf der Oberfläche angeordnet werden. Der einfachheitshalber verwenden wir hier nur die `grid`-Methode.

Würdest du das Programm nun ausführen, solltest du folgende Oberfläche angezeigt bekommen:



Schritt 3: Weitere Komponenten

Nun sollen weitere Komponenten hinzugefügt werden. Füge dazu den folgenden Quellcode nach der Platzierung des ersten Eingabefeldes hinzu:

```

# Plus-Label hinzufuegen
plus_label = Label(fenster, width=5, text="+")
plus_label.grid(row=0, column=1)

# Erstes Eingabefeld hinzufuegen
zahl1_entry = Entry(fenster, width=5, justify="center")
zahl1_entry.grid(row=0, column=2)

# Gleich-Button hinzufuegen
gleich_button = Button(fenster, width=5, text="=", command=addiere)
gleich_button.grid(row=0, column=3)

# Ergebnis-Label hinzufuegen
ergebnis_label = Label(fenster, width=5, text="")
ergebnis_label.grid(row=0, column=4)

```

Im obigen Quellcode werden nun vier weitere Komponenten hinzugefügt. Ein Label mit einem Pluszeichen als Text, ein weiteres Eingabefeld, ein Button mit einem Gleichheitszeichen als Text und ein Label ohne Text. Der Quellcode der neuen Komponenten ist ähnlich zu dem aus dem letzten Schritt. Einzige Besonderheit ist der Parameter *command* des Buttons. Dieser gibt an, welche Methode/Funktion ausgeführt wird, sobald auf diesen Button geklickt wird.

Die grafische Oberfläche würde nun wie folgt aussehen:



Ausführen lässt sich das Programm aber noch nicht, da die Methode/Funktion *addiere* noch nicht definiert wurde.

Schritt 4: Das Addieren

Wichtig ist, dass die Funktion *addiere* vor dem Button definiert werden muss, da dieser auf die Funktion verweist. Es empfiehlt sich, solche Funktionen immer am Anfang, direkt nach dem Importieren der Bibliotheken, zu definieren. Füge folgenden Quellcode vor der Initialisierung des Fenster hinzu:

```

# Methode zum Addieren definieren
def addiere():
    # Hole Eingaben aus Eingabefeldern
    zahl1 = zahl1_entry.get()
    zahl2 = zahl2_entry.get()

    # Pruefe ob Eingaben auch Zahlen sind
    if (zahl1.isdigit() and zahl2.isdigit()):
        # Wandle Eingaben in Integer um
        zahl1 = int(zahl1)
        zahl2 = int(zahl2)

        # Aendere Text des Ergebnis-Labels in berechnetes Ergebnis
        ergebnis_label.config(text = str(zahl1 + zahl2))

```

An sich sollte der obige Quellcode anhand der Kommentare selbsterklärend sein. Wichtige Stellen sind hier nur die Verwendung der *get()* Methode, um den Inhalt der Eingabefelder zu erhalten, das Prüfen auf Integer-Zahlen mittels *isdigit()* und das Ändern des *text*-Attributes des Ergebnislabels mittels *config()*.

Schritt 5: Ungültige Eingaben

Was nun jedoch noch fehlt ist, eine Nachricht oder ein Hinweis, sofern keine ordentlichen Eingaben getätigt wurden. Hierfür muss die *if*-Anweisung der *addiere*-Funktion mit einer *else*-Anweisung erweitert werden. Als kleine Besonderheit soll jedoch die Fehlermeldung in Form eines kleinen Hinweisfenster angezeigt werden. Solche Nachrichtenfenster können mit einer zusätzlichen Klasse ganz einfach angezeigt werden.

Importiere dazu die Klasse *tkMessageBox*. Füge dazu folgenden Quellcode nach dem Import von Tkinter ein:

```
import tkMessageBox
```

Anschließend kannst du folgenden Quellcode an das Ende der *addiere* Funktion hinzufügen:

```
# Wenn Eingaben keine Zahl, dann Hinweis-Fenster anzeigen
else:
    tkMessageBox.showinfo("Fehler", "Fehlerhafte Eingabe!")
```

Die Methode *showinfo* der Klasse *tkMessageBox* sorgt dafür, dass ein Nachrichtenfenster mit dem Titel *Fehler* und der Nachricht *Fehlerhafte Eingabe!* angezeigt wird.

Hiermit ist dein erstes Programm mit grafischer Oberfläche fertig.

Überblick

Der Quellcode des gesamten Programms sollte, wenn du alles richtig gemacht hast, wie folgt aussehen:

```
# Laden der Bibliothek
from Tkinter import *
import tkMessageBox

# Methode zum Addieren definieren
def addiere():
    # Hole Eingaben aus Eingabefeldern
    zahl1 = zahl1_entry.get()
    zahl2 = zahl2_entry.get()

    # Pruefe ob Eingaben auch Zahlen sind
    if (zahl1.isdigit() and zahl2.isdigit()):
        # Wandle Eingaben in Integer um
        zahl1 = int(zahl1)
        zahl2 = int(zahl2)

        # Aendere Text des Ergebnis-Labels in berechnetes Ergebnis
        ergebnis_label.config(text = str(zahl1 + zahl2))

    # Wenn Eingaben keine Zahl, dann Hinweis-Fenster anzeigen
    else:
        tkMessageBox.showinfo("Fehler", "Fehlerhafte Eingabe!")

# Fenster initialisieren und Titel setzen
fenster = Tk()
fenster.title("Additionsrechner")

# Erstes Eingabefeld hinzufuegen
zahl1_entry = Entry(fenster, width=5, justify="center")
zahl1_entry.grid(row=0, column=0)

# Plus-Label hinzufuegen
plus_label = Label(fenster, width=5, text="+")
plus_label.grid(row=0, column=1)
```

```
# Erstes Eingabefeld hinzufuegen
zahl1_entry = Entry(fenster, width=5, justify="center")
zahl1_entry.grid(row=0, column=2)

# Gleich-Button hinzufuegen
gleich_button = Button(fenster, width=5, text="=", command=addiere)
gleich_button.grid(row=0, column=3)

# Ergebnis-Label hinzufuegen
ergebnis_label = Label(fenster, width=5, text="")
ergebnis_label.grid(row=0, column=4)

# Fenster ausfuehren
fenster.mainloop()
```

Ein Berechnung auf der grafischen Oberfläche kann so aussehen:



Memory

Als nächstes Projekt mit Python kannst du dein eigenes Memory möglichst selbstständig programmieren. Hilfestellung bekommst du dabei in erster Linie durch die folgenden Beispiele und der Recherche in Online-Tutorials. Eine kleine Übersicht guter Tutorials findest du am Ende.

Aufgabenstellung

Programmiere dein eigenes Memory mit grafischer Oberfläche. Das Spielfeld sollte aus 4 Paaren von Karten bestehen. Der Spieler soll auf eine Karte klicken können, woraufhin diese aufgedeckt wird. Sollte bereits eine Karte aufgedeckt worden sein, dann muss geprüft werden ob es ein Paar ist. Wenn ja, dann bleiben beide Karten aufgedeckt und der Spieler kann fortfahren, wenn nicht, dann werden die beiden Karten wieder umgedreht. Zu Beginn wird es nur einen Spieler ohne Punktestand geben.

Ein Grundgerüst des Programms inklusive Bilder für die Karten steht dir in Form eines ZIP-Archives bereit.

Hilfestellung zum Memory

Bilder als Buttons

Um Bilder in einem Button einzubinden, muss zunächst einmal ein Objekt der Klasse `PhotoImage` (in der `Tkinter`-Bibliothek enthalten) erzeugt werden:

```
bild = PhotoImage(file='Pfad zur GIF-Datei')
```

Wichtig ist, dass es sich bei Python 2.7 um eine GIF-Datei handelt. Andere Dateiformate werden erst ab Python 3.6 unterstützt.

Anschließend kann das Bild dem Button mittels `image` Attribut zugewiesen werden:

```
Button(..., image=bild, ...)
```

Hier ein ganzes Beispiel:

```
fenster = Tk()
bild = PhotoImage(file='button.gif')
button = Button(fenster, image=bild)
button.pack()
fenster.mainloop()
```

Verzögertes Ausführen einer Funktion

Angenommen du möchtest nach 5 Sekunden etwas ausführen lassen, dafür bietet das `Tk`-Objekt (also das Fenster) die `after`-Methode an:

```
fenster.after(ZEIT, FUNKTION, PARAMETER ...)
```

Als erstes muss immer die Wartezeit in Millisekunden übergeben werden, anschließend der Name der Funktion die ausgeführt werden soll (also ohne Klammern und Parameter) und letztlich alle Parameter, die der Funktion übergeben werden sollen.

Möchtest du zum Beispiel deine eigene Funktion `karte_umdrehen(karte)` nach 5 Sekunden ausführen lassen, so müsste der Quellcode wie folgt lauten:

```
fenster.after(500, karte_umdrehen, karte)
```

Button soll auf Drücken eine Funktion mit Parametern aufrufen

Mit dem `command`-Attribut kann bei einem Button eine Methode angegeben werden, die ausgeführt wird, sobald der Button gedrückt wird. Hierbei ist es aber nicht direkt möglich, der Methode auch Parameter zu übergeben. Dafür benötigt man den Befehl `partial` aus der Bibliothek `functools`:

```
from functools import partial
```

Möchte man nun zum Beispiel beim Drücken des Buttons die Funktion `karte_aufdecken` mit den Parametern 2 und 3 ausführen lassen, dann lautet der Quellcode dafür:

```
mein_button = Button(..., command=partial(karte_aufdecken, 2, 3), ...)
```

Statt festen Werten wie 2 und 3 können natürlich auch Variablen verwendet werden.

Array (verschachtelte Listen) initialisieren

Manchmal möchte man zunächst einmal einen Array mit einer festen Größe initialisieren, bevor man ihn befüllt. Dies ist in Python etwas umständlicher als in anderen Programmiersprachen. Der folgende Quellcode erzeugt einen leeren Array mit vorgegebener Anzahl an Spalten und Zeilen:

```
mein_array = [['' for i in range(anzahl_spalten)] \\
              for j in range(anzahl_zeilen)]
```

Das Ergebnis bei 3 Spalten und 2 Zeilen würde dann so aussehen:

```
[[[], [], []],  
 [[], [], []]]
```

Anschließend kann zum Beispiel auf das Feld 1, 0 (zweite Zeile und erste Spalte; beim Programmieren zählt man von 0 an) ein neuer Wert hinterlegt werden:

```
mein_array[1][0] = 'a1'
```

Zufälliges Element aus einer Liste nehmen

Um aus einer Liste ein zufälliges Element zu nehmen, benötigt man die Funktion *choice* aus der Klasse *random*:

```
from random import choice
```

Anschließend wird mit *choice(meine_liste)* ein zufälliges Element aus der Liste *meine_liste* zurückgegeben. Das Element wird dabei nicht aus der Liste entfernt. Hier ein Beispiel:

```
moegliche_karten = ['a1', 'a2', 'b1', 'b2', 'c1', 'c2', 'd1', 'd2']  
zufaellige_karte = choice(moegliche_karten)
```

Weitere Tutorials

Hier eine kleine Übersicht nützlicher Tutorials zu GUI Programmierung in Python:

- http://www.python-kurs.eu/python_tkinter.php
Zeigt und erklärt die wichtigsten Komponenten und gibt Beispiele zur Einbindung.
- http://www.tutorialspoint.com/python/python_gui_programming.htm
Listet nicht nur wichtige Komponenten auf, sondern nennt auch deren wichtigsten Attribute.
- <http://www.tkdocs.com/tutorial/index.html>
Sehr umfangreiches Tutorial. Rechts in der Navigation, sollte Python als Programmiersprache im Dropdown-Menü ausgewählt werden.



Objektorientierte Programmierung – Teil 2

Im ersten Teil zur objektorientierten Programmierung gab es noch einige Sachen, die unerwähnt blieben und trotzdem wichtig sind. So erlaubt uns diese Programmierung noch ein paar weitere Feinheiten bei der Verwendung von Klassen sowie auch Beziehungen zwischen den Klassen zu definieren.

Kapselung von Daten

Häufig ist es sinnvoll, dass die Daten innerhalb eines Objekts (vorgegeben durch die Klasse) nicht von jeder x-beliebiger Stelle direkt verändert werden können. Angenommen wir haben ein Objekt Kunde mit dem Attribut Geburtsdatum. So bietet es sich an, dass die Änderung dieses Attributes nur durch eine interne Methode der Klasse durchgeführt wird. Dies hätte dann den Vorteil, dass zum Beispiel immer geprüft werden kann, ob es sich um ein gültiges Datum handelt. Dies wäre nicht möglich, wenn der Wert des Attributes durch alle möglichen Eingabeketten direkt verändert werden kann.

Mehr Sicherheit bietet daher die Möglichkeit, dass Attribute als *public* (öffentlich), *protected* (gesichert) oder *private* (privat) deklariert werden können:

- **public**

Diese Variablen besitzen keine Unterstriche vor dem Namen und können sowohl innerhalb des Objekts als auch von außen gesehen und verändert werden.

- **protected**

Variablen, die einen einzigen Unterstrich vor ihrem Namen haben, sind *protected*. Das bedeutet, dass man diese zwar auch von innen und außen heraus sehen und bearbeiten kann, aber es laut Entwickler nicht sollte. Der Entwickler möchte damit zeigen, dass es sich um nicht veränderbare Werte (Konstanten) handelt.

- **private**

Diese Variablen besitzen genau zwei Unterstriche vor ihrem Namen. Sie können nur innerhalb des Objekts verwendet werden und sind nicht einmal von außen zu sehen.

Hier ein kleines Beispiel:

```
class Konto:
    def __init__(self, inhaber, kontostand, kontonummer):
        self.__inhaber = inhaber
        self.__kontostand = kontostand
        self.__kontonummer = kontonummer

    def gib_kontostand(self):
        return self.__kontostand

    def auszahlen(self, betrag):
        if (betrag <= self.__kontostand):
            return True
            self.__kontostand -= betrag
        else:
            return False

    def einzahlen(self, betrag):
        self.__kontostand += betrag

konto = Konto('Max Mustermann', 500, 1234567890)
print('Ich habe ' + str(konto.gib_kontostand()) + ' Euro')
```



```

konto.__kontostand = 1000000
if konto.auszahlen(1000000):
    print('Ich bin Millionär!')
else:
    print('Ich bin immer noch kein Millionär...')


```

Ausgeführt wäre das Ergebnis dann:



Statische Attribute

Bisher besaßen zwei Objekte der selben Klasse meist zwar das gleiche Attribut nur mit unterschiedlichen Werten. Für die Programmeierung ist es manchmal auch sinnvoll, wenn es ein Attribut gibt, dass bei allen Objekten immer den gleichen Wert besitzt. Solche Attribute nennt man *static* (statisch).

In dem folgenden Beispiel gibt es das Attribut `objekt_zahler`. Dieses soll immer angeben, wie viele Objekte es von der Klasse gibt.

```

class Zaehler:
    zaehler = 0

    def __init__(self, name):
        self.name = name
        Zaehler.zaehler += 1
        print('Es gibt einen neuen Zaehler mit dem Namen ' + \
              self.name + '.')

    def nenne_anzahl(self):
        print(self.name + ' sagt, dass es ' + str(Zaehler.zaehler) + \
              ' Zaehler gibt.')

    def __del__(self):
        print('Der Zaehler mit dem Namen ' + self.name + \
              ' wurde geloescht.')
        Zaehler.zaehler -= 1

mein_zahler1 = Zaehler('Maxi')
mein_zahler1.nenne_anzahl()
print('\n')
mein_zahler2 = Zaehler('Max')
mein_zahler1.nenne_anzahl()
mein_zahler2.nenne_anzahl()
print('\n')
mein_zahler3 = Zaehler('Hugo')
mein_zahler1.nenne_anzahl()
mein_zahler2.nenne_anzahl()
mein_zahler3.nenne_anzahl()
print('\n')
del(mein_zahler1)
del(mein_zahler2)
mein_zahler3.nenne_anzahl()


```

Ausgeführt wäre das Ergebnis dann:



```
C:\Windows\system32\cmd.exe
Es gibt einen neuen Zaehler mit dem Namen Maxi.
Maxi sagt, dass es 1 Zaehler gibt.

Es gibt einen neuen Zaehler mit dem Namen Max.
Maxi sagt, dass es 2 Zaehler gibt.
Max sagt, dass es 2 Zaehler gibt.

Es gibt einen neuen Zaehler mit dem Namen Hugo.
Maxi sagt, dass es 3 Zaehler gibt.
Max sagt, dass es 3 Zaehler gibt.
Hugo sagt, dass es 3 Zaehler gibt.

Der Zaehler mit dem Namen Maxi wurde geloescht.
Der Zaehler mit dem Namen Max wurde geloescht.
Hugo sagt, dass es 1 Zaehler gibt.
```

Vererbung

Betrachten wir nun erst einmal wieder das Beispiel mit den Pizzen. Es gab eine Klasse *Pizza* mit Standardteig und -zutaten. Nun gibt es bei vielen Lieferdiensten unterschiedliche Größen von Pizzen. Da wären zum Beispiel die kleinen Pizzen, die Normalgroßen oder Familienpizzen. Bei allen handelt es sich um Pizza und auf alle können die gleichen Zutaten enthalten, aber trotzdem gibt es Unterschiede wie zum Beispiel den Preis. Angenommen jede Zutat kostet bei einer kleinen Pizza 50 Cent, bei einer Normalen 1 Euro und bei einer Familienpizza 1,50 Euro. Die Berechnung des Preises lässt sich verschieden lösen:

1. Wir erweitern die Klasse *Pizza* um ein Attribut *groesse*. Anschließend kann in der Methode *preis_berechnen()* geprüft werden, welche Größe die Pizza hat und der Preis entsprechend ermittelt werden.
2. Wir nutzen Vererbung. Hierbei werden von der Klasse *Pizza* die Unterklassen *KleinePizza*, *NormalePizza* und *FamilienPizza* erzeugt. Dies hätte den Vorteil, dass kein weiteres Attribut notwendig ist und man kann einzelne Methoden unterschiedlich definieren und programmieren.

Der folgende Quellcode zeigt die Lösung 1:

```
class Pizza:
    def __init__(self):
        self.teig = 'Hefeteig'
        self.zutaten = ['Tomatensosse', 'Käse']
        self.rand = 'ohne'
        self.groesse = 'normal'

    def bestellen(self):
        print('Pizza mit ' + self.teig + ', ' + self.rand + \
              '-Rand wird bestellt.')
        print('Zutaten-Liste:')
        print(self.zutaten)
        print('Preis in Euro:')
        print(str(self.berechne_preis()))

    def berechne_preis(self):
        if (self.groesse == 'klein'):
            return len(self.zutaten) * 0.5
        elif (self.groesse == 'normal'):
            return len(self.zutaten) * 1
```



```

        elif (self.groesse == 'familie'):
            return len(self.zutaten) * 1.5

meine_pizza = Pizza()
meine_pizza.zutaten.append('Salami')
meine_pizza.zutaten.append('Schinken')
meine_pizza.bestellen()

```

Der folgende Quellcode zeigt die Lösung 2:

```

class Pizza:
    def __init__(self):
        self.teig = 'Hefeteig'
        self.zutaten = ['Tomatensosse', 'Kaese']
        self.rand = 'ohne'

    def bestellen(self):
        print('Pizza mit ' + self.teig + ', ' + self.rand + \
              '-Rand wird bestellt.')
        print('Zutaten-Liste:')
        print(self.zutaten)
        print('Preis in Euro:')
        print(str(self.berechne_preis()))

    def berechne_preis(self):
        pass

class KleinePizza(Pizza):
    def berechne_preis(self):
        return len(self.zutaten) * 0.5

class NormalePizza(Pizza):
    def berechne_preis(self):
        return len(self.zutaten) * 1

class FamilienPizza(Pizza):
    def berechne_preis(self):
        return len(self.zutaten) * 1.5

meine_pizza = NormalePizza()
meine_pizza.zutaten.append('Salami')
meine_pizza.zutaten.append('Schinken')
meine_pizza.bestellen()

```

KleinePizza ist eine Unterklasse von Pizza.

Ähnlich zu Methoden wird die Mutterklasse bei der Definierung der Unterklassen übergeben. Eine Klasse kann auch von mehreren Klassen erben.

Methoden können in Unterklassen anders definiert bzw. programmiert werden. Alles andere bleibt wie in der Mutterklasse.

Generell lässt sich hier nun nicht sagen, welche Lösung besser ist. Wichtig für das Verständnis von Vererbung ist noch, dass die Unterklassen alle Attribute und Methoden der Mutterklasse besitzen. Der Vorteil ist aber, dass diese Unterklassen unterschiedlich voneinander erweitert oder verändert werden können.

Entwurfsmuster

Ein weiterer fundamentaler Vorteil, den die objektorientierte Programmierung mit sich bringt, ist die Verwendung von Entwurfsmustern. Hierbei handelt es sich um Lösungsansätze für wiederkehrende Entwurfsprobleme bei der Programmierung (besonders bei großen und komplexen Projekten). Es gibt sehr viele verschiedene solcher Entwurfsmuster für die unterschiedlichsten Probleme, die zum Teil auch untereinander kombiniert werden können. Eines, dass hier erwähnt werden soll, ist das *Model-View-Controller* (MVC) Entwurfsmuster.

Beim MVC werden bestimmte Bereiche eines größeren Programms voneinander getrennt, damit die Entwicklungen in größeren Teams erleichtert und mehr Übersicht über den zu entstehenden Quellcode ermöglicht wird. Programme werden beim MVC in das Model, das

View und dem Controller unterteilt ist. Dabei ist nicht ausgeschlossen, dass ein Programm nicht über mehrere Models, Views und/oder Controller verfügen kann.

- **Controller**

Der Controller beinhaltet die Hauptlogik des Programms. Er steuert das ganze Programm und gibt Anweisungen an das Model und den View.

- **Model**

Das Model dient als Datenquelle für das Programm. Das bedeutet, dass alle notwendigen Informationen in dieser Klasse enthalten sind und der Controller sich diese vom Model holt.

- **View**

Der View ist für die visuellen Rückgaben an den Benutzer zuständig. Dies kann zum Beispiel das Zusammenbauen und Anzeigen einer grafischen Oberfläche sein. Änderungen an dieser grafischen Oberfläche werden dann meist vom Controller angestoßen und durch den View ausgeführt.

Miteinander verbunden bieten diese Klassen also eine Trennung von Daten, Programmlogik und Ausgabe. Wie bereits gesagt, kann dies den Vorteil haben, dass in einem größeren Projekt sich einzelne Entwickler zum Beispiel nur auf die Bereitstellung, Validierung und Verarbeitung von Daten befassen (also nur am Model arbeiten). Anders könnte es aber auch sein, dass sich einzelne Entwickler nur mit der Gestaltung der grafischen Oberfläche befassen (also nur am View arbeiten). Im letzten Fall könnte es dann auch einzelne Entwickler geben, die sich nur mit der Programmlogik befassen (also nur am Controller arbeiten). Wichtig bei größeren Projekten ist hier, dass alle Entwickler genau miteinander kommunizieren und das Programm auch in Gänze planen. Die letztliche Umsetzung kann dann in kleinere Gruppen unterteilt werden.



Handelssimulator

Zum Abschluss folgt nun ein etwas größeres Projekt. Bei dem (historischem) Handelssimulator handelt es sich um ein recht simples Simulationsspiel. Der Spieler schlüpft in die Rolle eines Händlers und kann zwischen verschiedenen Städten hin und her reisen. Innerhalb der Städte kann der Spieler dann verschiedene Waren ein- und verkaufen.



Aufgabe 1

Als Grundlage für dieses Programm erhältst du eine Vorlage mit MVC Entwurfsmuster. Schau dir daher den Quellcode genau an, bis du den Ablauf des Programms nachvollzogen hast. Die folgenden Übersichten zeigen dir dabei, welche Attribute und Methoden die Klassen besitzen.

Hinweis: Hinter jedem Doppelpunkt steht immer der Datentyp des Rückgabewerte oder des Attributes bzw. Parameters. Void steht dabei für eine Methode ohne Rückgabewert.

Handelssimulator_Controller	
<u>__model</u> : Handelssimulator_Model	Beinhaltet ein Objekt der Klasse <i>Handelssimulator_Model</i> .
<u>__view</u> : Handelssimulator_View	Beinhaltet ein Objekt der Klasse <i>Handelssimulator_View</i> .
<u>__init__</u> () : Void	Mit dieser Methode wird der Controller initialisiert, das Programm aber noch nicht gestartet.
<u>kaufen()</u> : Void	Diese Methode wird ausgeführt, sobald man im View auf den Button <i>kaufen</i> klickt.
<u>reisen()</u> : Void	Diese Methode wird ausgeführt, sobald man im View auf den Button <i>weiterreisen</i> klickt.
<u>schiff_kaufen()</u> : Void	Diese Methode wird ausgeführt, sobald man im View auf den Button <i>Schiff kaufen</i> klickt.
<u>starte()</u> : Void	Mit dieser Methode kann das Programm gestartet werden.
<u>verkaufen()</u> : Void	Diese Methode wird ausgeführt, sobald man im View auf den Button <i>verkaufen</i> klickt.

Handelssimulator_Model	
<code>__aktuelle_stadt : String</code>	Beinhaltet den Namen einer Stadt und legt damit den aktuellen Standort des Spielers fest.
<code>__anzahl_schiffe : Integer</code>	Beinhaltet die Anzahl der aktuellen Schiffe im Besitz des Spielers.
<code>__gold : Integer</code>	Beinhaltet die Anzahl an Gold im Besitz des Spielers.
<code>__lager : Liste</code>	Beinhaltet eine Liste mit den einzelnen Mengen der Handelswaren im Besitz des Spielers
<code>__lager_pro_schiff : Integer</code>	Beinhaltet die Lagerkapazität eines Schiffes bzw. legt diese fest.
<code>__max_preis : Integer</code>	Beinhaltet den Maximalpreis einer Handelsware bzw. legt diese fest.
<code>__min_preis : Integer</code>	Beinhaltet den Minimalpreis einer Handelsware bzw. legt diese fest.
<code>__preis_pro_schiff : Integer</code>	Beinhaltet den Kaufpreis eines Schiffes bzw. legt diese fest.
<code>__preise : Liste</code>	Beinhaltet eine Liste mit den einzelnen Preisen der Handelswaren für alle Städte.
<code>__staedte : Liste</code>	Beinhaltet eine Liste mit den Namen der einzelnen Städte, welche vom Spieler bereist werden können.
<code>__waren : Liste</code>	Beinhaltet eine Liste mit den Namen der Handelswaren, welche vom Spieler ein- und verkauft werden können.
<code>berechne_lagerplatz() : Integer</code>	Mit dieser Methode kann der freie Lagerplatz berechnet werden. Rückgabewert ist der freie Lagerplatz als Integer.
<code>berechne_neue_preise() : Void</code>	Mit dieser Methode kann die Preisschwankung berechnet werden.
<code>gib_aktuelle_stadt() : String</code>	Gibt den Wert der Variable <code>__aktuelle_stadt</code> zurück.
<code>gib_anzahl_schiffe() : Integer</code>	Gibt den Wert der Variable <code>__anzahl_schiffe</code> zurück.
<code>gib_gold() : Integer</code>	Gibt den Wert der Variable <code>__gold</code> zurück.
<code>gib_lager() : Liste</code>	Gibt den Wert der Variable <code>__lager</code> zurück.
<code>gib_preise() : Liste</code>	Gibt den Wert der Variable <code>__preise</code> zurück.
<code>gib_staedte() : Liste</code>	Gibt den Wert der Variable <code>__staedte</code> zurück.
<code>gib_waren() : Liste</code>	Gibt den Wert der Variable <code>__waren</code> zurück.
<code>setze_aktuelle_stadt(stadt : String) : Void</code>	Setzt den Wert der Variable <code>__aktuelle_stadt</code> .
<code>setze_gold(gold : Integer) : Void</code>	Setzt den Wert der Variable <code>__gold</code> .
<code>setze_lager(lager : Liste) : Void</code>	Setzt den Wert der Variable <code>__lager</code> .

Handelssimulator_View	
<code>__fenster : Tk</code>	Beinhaltet ein Objekt der Klasse Tk. Ist damit ein Fenster der grafischen Oberfläche.
<code>__gold_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Verfügbares Gold:“.
<code>__handelswaren_auswahl : Listbox</code>	Beinhaltet ein Auswahlfeld und ist ein Element auf dem Fenster. Besitzt als Auswahlelemente die einzelnen Handelswaren inklusiver ihres Preises.



Arbeitsmaterial A3.7

<code>__handelswaren_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Handelswaren“.
<code>__kaufen_button : Button</code>	Beinhaltet einen Button und ist ein Element auf dem Fenster. Beim Betätigen des Buttons soll eine Handelsware gekauft werden.
<code>__kaufen_methode : Function</code>	Beinhaltet einen Verweis auf eine Methode der Klasse <i>Handelssimulator_Controller</i> .
<code>__lager_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Verfügbarer Lagerplatz:“.
<code>__lagerraum_auswahl : Listbox</code>	Beinhaltet ein Auswahlfeld und ist ein Element auf dem Fenster. Besitzt als Auswahlelemente die einzelnen Handelswaren inklusiver ihrer Anzahl im Lager.
<code>__lagerraum_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Lagerraum“.
<code>__menge : Entry</code>	Beinhaltet ein Eingabefeld und ist ein Element auf dem Fenster. Auf den Wert des Eingabefeldes wird beim Ein- und Verkauf zugegriffen.
<code>__ort_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Aktueller Ort:“.
<code>__platzhalter : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Das Label besitzt einen Inhalt und dient nur als Platzhalter.
<code>__platzhalter2 : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Das Label besitzt einen Inhalt und dient nur als Platzhalter.
<code>__reisen_button : Button</code>	Beinhaltet einen Button und ist ein Element auf dem Fenster. Beim Betätigen des Buttons soll zu einer neuen Handelsstadt gereist werden.
<code>__reisen_methode : Function</code>	Beinhaltet einen Verweis auf eine Methode der Klasse <i>Handelssimulator_Controller</i> .
<code>__schiff_button : Button</code>	Beinhaltet einen Button und ist ein Element auf dem Fenster. Beim Betätigen des Buttons soll ein weiteres Schiff gekauft werden.
<code>__schiff_kaufen_methode : Function</code>	Beinhaltet einen Verweis auf eine Methode der Klasse <i>Handelssimulator_Controller</i> .
<code>__schiffe_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Schiffe im Besitz:“.
<code>__staedte_auswahl : Listbox</code>	Beinhaltet ein Auswahlfeld und ist ein Element auf dem Fenster. Besitzt als Auswahlelemente die einzelnen Handelsstädte.
<code>__staedte_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Handelsstädte“.
<code>__status_label : Label</code>	Beinhaltet ein Label und ist ein Element auf dem Fenster. Besitzt den Text „Händlerstatus“.
<code>__verkaufen_button : Button</code>	Beinhaltet einen Button und ist ein Element auf dem Fenster. Beim Betätigen des Buttons soll eine Handelsware verkauft werden.
<code>__verkaufen_methode : Function</code>	Beinhaltet einen Verweis auf eine Methode der Klasse <i>Handelssimulator_Controller</i> .
<code>__init__() : Void</code>	In dieser Methode werden Variablen initialisiert und die GUI des Programms zusammengebaut.
<code>aktualisiere_handelswaren_auswahl(waren : Liste, preise : Liste) : Void</code>	Mittels dieser Methode soll die Auswahl der Handelswaren in der GUI aktualisiert werden.



<code>aktualisiere_lagerraum_auswahl(waren : Liste, lager : Liste) : void</code>	Mittels dieser Methode soll die Auswahl des Lagerraumes in der GUI aktualisiert werden.
<code>aktualisiere_staedte_auswahl(staedte : Liste) : Void</code>	Mittels dieser Methode soll die Auswahl der Städte in der GUI aktualisiert werden.
<code>aktualisiere_status(aktuelle_stadt : String, gold : Integer, lagerplatz : Integer, anzahl_schiffe : Integer) : void</code>	Mittels dieser Methode soll der Status-Bereich der GUI aktualisiert werden.
<code>gib_ausgewahlte_handelsware() : Liste</code>	Gibt das ausgewählte Element in __handelswaren_auswahl zurück.
<code>gib_ausgewahlte_lagerware() : Liste</code>	Gibt das ausgewählte Element in __lagerraum_auswahl zurück.
<code>gib_ausgewahlte_stadt() : Liste</code>	Gibt das ausgewählte Element in __staedte_auswahl zurück.
<code>gib_eingabe_menge : String</code>	Gibt die Eingabe in __menge zurück.
<code>kaufen() : Void</code>	Diese Methode wird ausgeführt, sobald man auf den Button <i>kaufen</i> klickt.
<code>reisen() : Void</code>	Diese Methode wird ausgeführt, sobald man auf den Button <i>weiterreisen</i> klickt.
<code>schiff_kaufen() : Void</code>	Diese Methode wird ausgeführt, sobald man auf den Button <i>Schiff kaufen</i> klickt.
<code>setze_kaufen_methode(methode : Function) : Void</code>	Setzt den Wert der Variable __kaufen_methode.
<code>setze_schiff_kaufen_methode(methode : Function) : Void</code>	Setzt den Wert der Variable __schiff_kaufen_methode.
<code>setze_reisen_methode(methode : Function) : Void</code>	Setzt den Wert der Variable __reisen_methode.
<code>setze_verkaufen_methode(methode : Function) : Void</code>	Setzt den Wert der Variable __verkaufen_methode.
<code>starte() : Void</code>	Mit dieser Methode kann die GUI gestartet werden.
<code>verkaufen() : Void</code>	Diese Methode wird ausgeführt, sobald man auf den Button <i>verkaufen</i> klickt.
<code>zeige_nachricht(title : String, nachricht : String) : Void</code>	Mit dieser Methode kann eine Nachrichtenbox (für Fehler, Informationen etc.) angezeigt werden.

Aufgabe 2

Beschreibe den Ablauf beim ...

- a) Starten des Spiel bis zu Anzeige der grafischen Oberfläche.
- b) Kauf einer Handelsware. (Beginn bei Drücken des Buttons.)

Erläutere dabei, auf welche Klassen und jeweilige Methoden zugegriffen wird. Nutze im Idealfall ein Sequenzdiagramm zur Beschreibung des Ablaufs.

Aufgabe 3

In der Vorlage sind noch nicht alle Funktionen (Waren aus dem Lagerraum verkaufen; ein weiteres Schiff kaufen) implementiert worden. Füge diese selber in die Vorlage ein und teste sie anschließend. (Du musst dafür lediglich in der Controller-Klasse arbeiten.)

Starten kannst du das Programm von der Datei *Handelssimulator.py* aus.



Aufgabe 4

Nimm dir aus der folgenden Liste weitere Features (Schwierigkeit nimmt absteigend zu) für das Spiel heraus und implementiere diese in das Programm:

- Ein Spielende nach einer bestimmten Anzahl an Spielrunden mit der Berechnung einer Punktzahl.
- Möglichkeit während der Reise von Piraten, Unwettern etc. Überrascht zu werden und einen Teil der Schiffsladung zu verlieren.
- Preise schwanken nicht global gleichmäßig, sondern für jede Stadt werden die Preise je Ware einzeln bestimmt. (Jede Stadt besitzt also eine eigene Preisliste für die Waren, welche sich bei jedem Weitereisen verändert.)
- Möglichkeit des Eintritts eines besonderen Ereignisses (z.B. Naturkatastrophen, Kriege etc.), die den Preis einer Ware in allen Städten für mehrere Spielrunden stark an- oder absteigen lassen.
- Einen Computergegner, der ebenfalls verschiedene Städte bereist und Waren ein- und verkauft. Am Ende sollten dessen Punktzahlen mit der des Spielers verglichen werden.
- Gestalte eine neue grafische Oberfläche mit mehr Bildern und Fenstern.



Befehlsübersicht

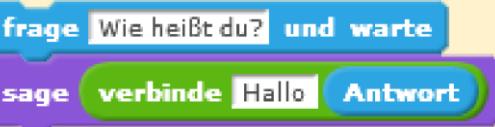
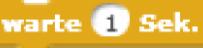
In der folgenden Tabelle sind einzelne Befehle aus Python dem Bausteinen aus Scratch gegenübergestellt.

Beschreibung	in Scratch	in Python
Text auf Bildschirm ausgeben		<code>print ("Hallo Welt!")</code>
Einer Variable eine Zahl zuordnen		<code>zahl = 42</code>
Einer Variable einen String zuordnen		<code>wort = "Fisch"</code>
Erhöhe den Wert der Variable <i>zahl</i> um 1		<code>zahl = zahl + 1</code> oder alternativ <code>zahl += 1</code>
Addieren		<code>zahl + 1</code>
Subtrahieren		<code>zahl - 2</code>
Multiplizieren		<code>zahl * 3</code>
Dividieren		<code>zahl / 4</code>
Setzt die Variable <i>zahl</i> auf einen zufälligen Wert zwischen 1 und 10		<code>zahl = randint(1, 10)</code> Hinweis: Der Befehl <code>randint()</code> muss importiert werden: <code>from random import randint</code>



Endlosschleife		<pre>while True: print ("Durchlaufe Schleife ...")</pre>
Schleife, die 10 mal durchlaufen wird		<pre>while i in range(10): print ("Durchlaufe Schleife ...")</pre>
Bedingung: Wenn <i>wort</i> gleich <i>Tannenbaum</i> ist, dann <i>Hello Welt!</i>		<pre>if (wort == "Tannenbaum"): print ("Hello Welt!")</pre>
<i>wort</i> gleich <i>Tannenbaum</i> ?		<pre>wort == "Tannenbaum"</pre>
Bedingung: Wenn <i>zahl</i> gleich 5 ist, dann <i>Hello Welt!</i> , sonst <i>Auf Wiedersehen!</i>		<pre>if (zahl == 5): print ("Hello Welt!") else: print ("Auf Wiedersehen!")</pre>
<i>zahl</i> gleich 5?		<pre>zahl == 5</pre>
<i>zahl</i> kleiner 10?		<pre>zahl < 10</pre>
<i>zahl</i> größer 20?		<pre>zahl > 20</pre>
Länge eines Strings		<pre>len(wort)</pre>



Eingabeaufforderung und Ausgabe der Antwort		antwort = input("Wie heißt du?") print ("Hallo " + antwort)
Verbinden von Strings		print ("Bernd hat " + zahl + " Äpfel.")
Definieren von Funktionen		def spring ():
Befehl ausführen		spring()
Bedingte Schleife mit Ungleichung		while (zahl != 5): spring()
Warten		sleep(1) Hinweis: Der Befehl sleep() muss importiert werden: from time import sleep
Definition einer Funktion mit Parameter		def gruesse(name): print ("Hallo" + name)
Befehl mit Parameter ausführen		gruesse("Hugo")



Liste anlegen	Neue Liste	<code>tiere = list()</code>
Füge <i>Tiger</i> zur Liste <i>tiere</i> hinzu		<code>tiere.append("Tiger")</code>
Lösche Eintrag aus Liste		<code>del tiere[1]</code>
Ausgabe der Anzahl der Listeneinträge		<code>print ("In der Liste sind " + len(tiere) + " Tiere.")</code>
Prüfe ob Element in der Liste ist		<code>if ("Katzenbaby" in tiere): print ("Wie süß!")</code>
Turtle-Grafik		<code>clear() pendown() forward(10) right(90) penup()</code> <p>Hinweis: Die obigen Befehle müssen importiert werden: <code>from turtle import *</code></p>

Python

Python ist eine **Programmiersprache**. Es liest einen Text ein, den du geschrieben hast (auch **Code** genannt), wandelt ihn in Befehle um, die für den Computer verständlich sind und führt diese dann aus. Du wirst lernen selbst Code zu schreiben, der coole und nützliche Dinge tut. Du wirst in der Lage sein, Programme selber zu schreiben, um mit Deinem Computer Dinge zu tun und nicht mehr nur Software von anderen Programmierern verwenden müssen.

Zuerst wirst du lernen Python zu benutzen und mit ihm zu kommunizieren. Es gibt mehrere Wege das zu tun; der erste ist, mit dem Python Interpreter über die Textkonsole deines **Betriebssystems** in Kontakt zu treten.

Eine **Konsole** (auch ‘Terminal’ oder ‘Prompt’) ist ein *schriftlicher* Weg deinem Betriebssystem Befehle zu erteilen, so wie der ‘Desktop’ zusammen mit deiner Maus ein *grafischer* Weg zu deinem System ist.

Wie öffne ich eine Konsole?

Die Windows-Konsole wird *Eingabeaufforderung* genannt, oder *cmd*. Die Tastenkombination Windows+R (Windows meint die Taste mit dem Windows-Logo) öffnet dieses als neues Fenster mit schwarzem Hintergrund. Gib dann *cmd* ein und klick Enter. Die erste Zeile sollte in etwa so aussehen:

```
C:\Users\myusername>
```

Du kannst den Python Interpreter (siehe unten) auch direkt ausführen, oder das IDLE-Programm das Python mitliefert. Diese findest Du im Startmenü.

Python verwenden

Python verhält sich standardmäßig als etwas das sich *Interpreter* nennt. Ein *Interpreter* nimmt Befehle entgegen und führt sie dann aus – super praktisch um Dinge auszuprobieren!

Tippe in deine Konsole *python* ein und drücke Enter um den Python-Interpreter zu starten. Sobald Python startet, zeigt es Dir einige Kontextinformationen ähnlich zu diesen:

```
Python 3.3.2 (default, January 21 2014, 15:40:45)
[GCC 4.8.0 20130502 (prerelease)] on linux
Type "help", "copyright", "credits" or "license" for more
information.

>>>
```

Tipp: Der Prompt *>>>* in der letzten Zeile signalisiert, dass du dich nun in der interaktiven *Python Konsole* befindest, auch *Python Shell* genannt. **Diese Konsole ist etwas anderes als das normale Kommandozeilenterminal!**

Theoretisch ist es also möglich in dieser *Python Shell* zu programmieren, jedoch dient es in den meisten Fällen nur für sehr kurze Anweisungen oder Tests. Für die Programmierung von richtigen Programmen verwendet man auch hier eine Entwicklungsumgebung. Als Entwicklungsumgebung reicht für Python theoretisch schon der ganz normale *Editor* unter Windows (nicht zu verwechseln mit Wordpad oder Word). Damit die Handhabung für dich aber

leichter ist, verwenden wir in dieser Einheit die Entwicklungsumgebung Geany, mit welcher sich unter anderem auch Python Programme schreiben, testen und ausführen lassen.

Python Installation

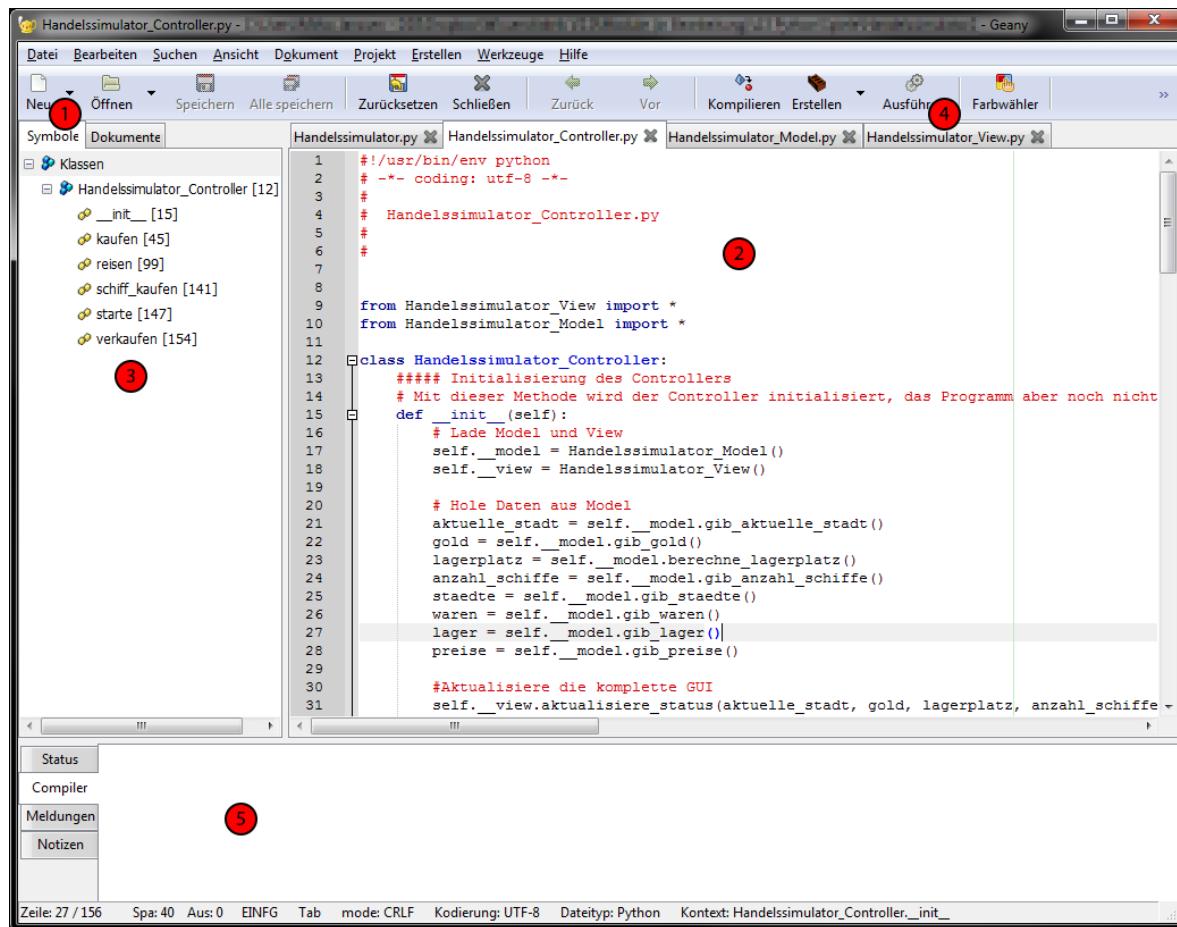
Wenn Python noch nicht auf deinem Computer installiert ist, findest du die aktuellen offiziellen Installationspakete hier:

<http://python.org/download/>

Wir verwenden Python 2.7 in dieser Unterrichtseinheit, prinzipiell sind beide Versionen nutzbar. Python 3 ist die eigentliche neueste Version, Python 2 wird jedoch parallel noch weiter mit entwickelt. Unter <https://wiki.python.org/moin/Python2orPython3> findest du die Hauptunterschiede und Gemeinsamkeiten beider Versionen.

Hinweis: Unter Windows ist es nötig, dass du Python zu deinem Systempfad hinzufügst, sodass anderen Programmen in der Lage sind Python zu finden. Eigentlich sollte das in dem Setup-Programm bereits ausgewählt und damit automatisch gemacht werden. Für den Fall, dass du beim späteren Programmieren merkst, dass dies nicht der Fall ist, kannst du in das Installationsverzeichnis (C:\Python33) navigieren, den Tools-Ordner und anschließend den Scripts-Ordner öffnen, durch ausführen der Datei win_add2path.py wird die Umgebungsvariable dann gesetzt.

Geany



1. Hier kann eine neue Datei angelegt werden. Wichtig zu erwähnen ist hier, dass verschiedene Formate (.py für Python) als Vorlagen ausgewählt werden können.
2. In diesem Bereich befindet sich der Quellcode des Programms.
3. Hier werden die Klassen und deren Methoden aus der aktuellen Datei aufgelistet.
4. Mit diesem Button kann das Programm ausgeführt werden.
5. In diesem Bereich werden alle Benachrichtigungen in Bezug auf das Kompilieren bzw. Ausführen des Programmes angezeigt. Fehlermeldungen werden grundsätzlich im Terminal des ausführten Programmes angezeigt.



IT2School

Gemeinsam IT entdecken



Modul E1 – IT kinderleicht

IT und Informatik spielend entdecken

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	IT von Anfang an.....	3
2	Warum gibt es das Modul?	4
3	Ziele	4
4	Inhalte des Moduls.....	4
4.1	Littlebits	4
4.2	Bee-Bot	5
4.3	Ozobot.....	6
5	Unterrichtliche Umsetzung.....	6
6	Literatur und Links	6

1 IT und Informatik spielend entdecken

Dieses Erweiterungsmodul stellt Ihnen verschiedene Möglichkeiten vor, wie auch schon jüngere Kinder auf spielerische Weise IT entdecken können. Hierfür bieten sich beispielsweise die Littlebits an. Die kleinen, elektronischen Bauelemente lassen sich magnetisch miteinander verbinden, wodurch einfache Schaltkreise entstehen. Die kleinen Bauteile regen zum Experimentieren und Ausprobieren an und es können vielfältige kleine Projekte damit umgesetzt werden. Hierdurch wird ein spielerischer Zugang zu den Grundlagen der Elektronik und Programmierung ermöglicht.

Lernfeld/Cluster:	IT spielend entdecken
Zielgruppe/Klassenstufe:	X Kita oder Vorschule
	X 1. bis 2. Klasse
	X 3. bis 4. Klasse
	X 5. bis 6. Klasse
Geschätzter Zeitaufwand:	2 – 4 Stunden
Lernziele:	<ul style="list-style-type: none">• Hintergründe von Elektronik verstehen• Einblick in die Funktionsweise vieler digitaler Konstruktionen• Einfache Anwendungen selbst gestalten• Kreative Denkweise schulen• Eigene Ideen umsetzen
Vorkenntnisse der Schülerinnen und Schüler:	Keine
Vorkenntnisse der/des Lehrenden:	Keine
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters:	Keine
Sonstige Voraussetzungen:	Keine

2 Warum gibt es das Modul?

In unserem Alltag gibt es viele verschiedene Formen von Informatiksystemen. Dies kann zum Beispiel der heimische Computer, das Smartphone, eine LED-Anzeige an der Bushaltestelle oder der Leergutautomat im Supermarkt sein. All diese Systeme haben gemeinsam, dass sie Informationen in Form von Daten aufnehmen, verarbeiten und ggf. ausgeben. Das EVA-Prinzip (Eingabe-Verarbeitung-Ausgabe) ist also ein wichtiger Bestandteil dieser Systeme. Um gerade Kindern dieses Prinzip und die Bedeutung von Informationssystemen näher zu bringen, können sie einfache Systeme kennenlernen, die sie spielerisch anwenden, analysieren und vergleichen können.

3 Ziele

- Hintergründe von Elektronik verstehen
- Einblick in die Funktionsweise vieler digitaler Konstruktionen
- Einfache Anwendungen selbst gestalten
- Kreative Denkweise schulen
- Eigene Ideen umsetzen

4 Inhalte des Moduls

4.1 Littlebits

Die Idee der Littlebits stammt von der Ingenieurin und Künstlerin Ayah Bdeir, die die Bausteine durch ihr Start-Up seit 2011 vertreibt. Die einzelnen Bausteine (Licht, Ton, Batterie, Taster, Motor, Verbindungen ...) sind in verschiedenen Bausätzen für Anfänger und Fortgeschrittene zusammengefasst. Jedes Bauteil, Bit genannt, hat eine bestimmte Funktion, die auch farblich gekennzeichnet ist (pink ist für *Input*, grün für *Output*, blau für *Power* usw.). Die Bausteine werden mit Magneten verbunden, wodurch ein falsches Zusammenstecken nicht möglich ist.

Über die „Power-Bits“ können Schaltungen per Batterie, USB oder Netzteil betrieben werden. Sie werden immer zu Beginn einer Schaltung gesetzt. Für einen Eingang oder Ausgang stehen mittlerweile über 60 verschiedene Bits zur Verfügung (z.B. Lichtsensor, Drucksensor, Temperaturfühler, Motoren, LEDs,...). Um „Und/Oder-Schaltungen“ zu stecken gibt es so genannte „Verbindungs-Bits“, die als Leitungssystem fungieren.

Durch die kinderleichte Anwendung können auch schon Projekte mit Kindern ohne Vorkenntnisse im Elektronikbereich umgesetzt werden. Aber auch Jugendliche können komplexere Gestaltungen und Erfindungen mit den Bauteilen umsetzen.

Die Grundidee von Ayah Bdeir war es, Kindern, Jugendlichen und interessierten Erwachsenen einen Einblick in die digitale Welt von heute zu geben, ohne löten und programmieren. Ihrer Ansicht nach, sollte heute jeder in der Lage sein, mit elektronischen Bauelementen zu konstruieren, Ideen zu entwickeln und in Form von Prototypen in die Tat umzusetzen. Sie inspirierte sich dabei von anderen Bausätzen, wie beispielsweise Lego.



Die Littlebits sind einfach in der Handhabung und Versuche und Schaltungen können im Gegensatz zu anderen Materialien viel schneller aufgebaut und durchgeführt werden. Dabei sind sie wiederverwendbar und ermöglichen Kindern und Jugendlichen ein spannendes und handlungsorientiertes Lernen.

Alle Modulentwicklungen und Schaltpläne werden von littleBits unter Creative-Commons-Lizenzen veröffentlicht. Es gibt vielfältige Inspirationen für den Einsatz in Schule und Unterricht (<http://littlebits.cc/education>).

4.2 Bee-Bot

Der Bee-Bot ist ein kleiner Roboter, der an eine Biene erinnert und der mit Tasten auf dem Rücken programmiert werden kann. Zur Programmierung stehen 7 Tasten zur Verfügung: *Vorwärts, Rückwärts, Linksdrehung, Rechtsdrehung, Programmierung löschen, Programmierung ausführen und Ausführung pausieren*.



Mit diesen Tasten lässt sich die Bewegung des Bee-Bots programmieren. Zur Unterstützung (und um die Motivation und Einsatzmöglichkeiten zu steigern) können verschiedenen Spielpläne genutzt werden. Dies sind Matten auf denen der Bee-Bot fahren kann und auf denen sich verschiedene Figuren, Buchstaben, Zahlen usw. befinden können. Mit Hilfe der Buchstabenmatte können beispielsweise Wörter buchstabiert werden oder der Beebot kann so programmiert werden, dass er den eigenen Namen abfährt. Auch können eigene Labyrinthe gebaut werden. Der kleine BeeBot kann dann so programmiert werden, dass er ohne Probleme durch das Labyrinth findet (siehe: <https://www.youtube.com/watch?v=9KNfgmDPVM>).

4.3 Ozobot

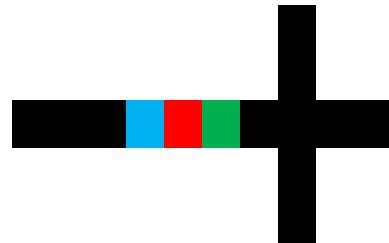
Beim Ozobot handelt es sich um einen kleinen Roboter der mit einem integrierten Farbsensor und Motor ausgestattet ist.

Dadurch kann er vorgefertigten oder auch selbst gemalten Linien folgen. In den Farben und Formen ist der Code für den kleinen Roboter enthalten z.B. für fahren, abbiegen, drehen, anhalten oder auch Geschwindigkeitsänderungen.



Nimmt man sich beispielsweise folgende Abbildung, so würde der Ozobot (ausgehend von links) der schwarzen Linie folgen, dann den Farbcode blau-rot-grün einlesen und an der Kreuzung rechts abbiegen.

Besonders schön ist, dass die Linien auch selbst gezeichnet werden können. Dadurch erfahren Kinder, dass der kleine Roboter genau das ausführt was programmiert wurde und man sein Verhalten direkt beeinflussen kann.



Auch eine App für den Tablet-PC ist für die Programmierung verfügbar. Der Ozobot folgt dann denn Linien direkt auf dem Tablet.

5 Unterrichtliche Umsetzung

Ein Variante, um alle Elemente kennen zu lernen, könnte die Stationenarbeit sein. Alle Kinder durchlaufen dabei die drei Stationen

- LittleBits
- Beebots
- Ozobots

Am Ende können die verschiedenen Mini-Roboter verglichen und ausgewertet werden.

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Vorstellen der drei Stationen inkl. Arbeitsauftrag
Vertiefung	30 – 45 min. pro Station,
Abschluss	Auswertung, wo liegen die Unterschiede, wo die Gemeinsamkeiten

Im nächsten Kapitel finden Sie viele weitere Ideen und Einsatzmöglichkeiten.

6 Literatur und Links

- Herstellerseite der **Littlebits**: <http://littlebits.cc/> Hier findet man auch zahlreiche Einsatzmöglichkeiten für Projekte
- **LittleBits Workshop Guide 2014:**
<https://d2q6sbo7w75ef4.cloudfront.net/Workshop+Guide+2014.pdf>

- **Littlebits bringen das Internet der Dinge ins Klassenzimmer:**
<https://www.yaez.com/blog/littlebits-bringt-das-internet-der-dinge-ins-klassenzimmer-2985/>
- **Littlebits und Papierbastelbögen** (z.B. R2D2) <http://littlebits.cc/projects/r2d2-bitrobot>
- **BeeBots:** <https://www.bee-bot.us/>
- **BeeBot.** Denken lernen - Probleme lösen: <http://beebot.ibach.at/>
- Medienkindergarten: **Der BeeBot – Ideen für den Einsatz im Kindergarten:**
<http://medienkindergarten.wien/medienpraxis/roboer-coding/der-bienenroboter-bee-bot/>
- **Herstellerseite Ozobot:** <http://ozobot.com>
- **Ozobots im Mathematik-Unterricht:** <http://www.projektschule-goldau.ch/permalink/33461>
- **Programmieraufträge für den Ozobot:** <https://blogs.phsg.ch/making/modul-werken2-0/kw13/ozobot/>

IT2School

Gemeinsam IT entdecken



Modul E2 – Wearable

Smarte Kleidung selbst gestalten

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



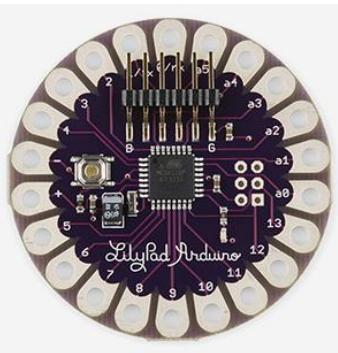
Inhalt

1	Wearables.....	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Inhalte des Moduls.....	4
5	Unterrichtliche Umsetzung.....	5
6	Literatur und Links	6

1 Wearables

In diesem Modul befassen sich die Schülerinnen und Schüler mit tragbaren und interaktiven Systemen und Controllern, wie sie beispielsweise in smarten Kleidungsstücken und Accessoires oder Smart Watches und FitnessTrackern Anwendung finden. Sie haben die Möglichkeit in eigenen Projekten selbst smarte Kleidung zu designen und zu erstellen.

Dieses Modul knüpft an die Lebenswelt der Jugendlichen und deren Interesse an Mode an. Darüberhinaus eröffnet es ihnen einen Blick auf die vielfältigen, teilweise sogar lebensrettenden Einsatzmöglichkeiten von IT beispielsweise in der Medizin. In diesem Fall werden Smart Textiles genutzt, die unter anderem die Temperatur sowie Atem- und Herzfrequenz von Patientinnen und Patienten überwachen.



Lernfeld/Cluster:	IT selber machen
Zielgruppe/Klassenstufe:	4. bis 5. Klasse
	6. bis 7. Klasse
	X 8. bis 10. Klasse
	X 11. bis 12. Klasse
Geschätzter Zeitaufwand:	12 – 15 Stunden
Lernziele:	<ul style="list-style-type: none">• Gestalterischen und kreativen Zugang zu IT und Informatik ermöglichen• Handlungsorientiert mit digitalen Technologien umgehen• Innovative bis hin zu lebensrettenden Einsatzmöglichkeiten von IT kennenlernen (in der Medizin, im Sport, in der Mode,...)
Vorkenntnisse der Schülerinnen und Schüler:	Empfohlen <ul style="list-style-type: none">• Erfahrung in der Programmierung (ggf. durch die Module B5 und B7)• Erfahrung im Nähen
Vorkenntnisse der/des Lehrenden:	Empfohlen <ul style="list-style-type: none">• Erfahrung in der Programmierung mit Arduino
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters	
Sonstige Voraussetzungen:	Voraussetzung <ul style="list-style-type: none">• Laptops/PCs für die Programmierung

2 Warum gibt es das Modul?

Kleidung und Mode spielt für viele Jugendliche eine besondere Rolle - sie ist Ausdruck von Gruppenidentität, Selbstinszenierung oder auch politisches Statement. Die Ästhetisierung des Alltags spiegelt sich aber nicht nur in der Mode und den Accessoires wider, sondern auch in der IT-Ausstattung. Diese beiden Interessen, Mode und Technologie, werden in diesem Modul zusammengebracht und ermöglichen Jugendlichen einen kreativen Zugang zu den Grundlagen der Informatik.

Zu Beginn befassen sich die Schülerinnen und Schüler mit smarten Textilien und entwickeln eigenständige Projekte: T-Shirts, Taschen, Hüte oder Armbänder können designt und programmiert werden, dazu werden sie mit LEDs, Sensoren und Aktoren sowie einem Microcontroller bestückt.

Dabei verbindet das Modul handwerklich-gestalterische Tätigkeiten mit Informatik und ermöglicht fächerübergreifendes Lernen, insbesondere zwischen Kunst, Informatik und Physik. Im Ende können die verschiedenen Einsatzmöglichkeiten von It näher beleuchtet werden. So bekommen die Schülerinnen und Schüler einen Eindrück der Relevanz der neuen Technik in verschiedenen Lebensbereichen, wie beispielsweise die lebensrettenden Funktionen in der medizinischen Anwendung.

3 Ziele des Moduls

- Gestalterischen und kreativen Zugang zu IT und Informatik ermöglichen
- Handlungsorientiert mit digitalen Technologien umgehen
- Innovative Einsatzmöglichkeiten von IT kennenlernen (in der Medizin, im Sport, in der Mode,...)
- Selbstwirksamkeit erfahren

4 Inhalte des Moduls

Wearable Computing, kurz Wearables bezeichnet alle Arten von Technologien, die am Körper getragen werden. Es gibt unterschiedlichste Ausführungen und verschiedenste Anwendungen tragbarer Technik. Bekannt sind vor allem Smart Watches - Uhren, die in der Regel mit einem Smartphone per Bluetooth verbunden sind und über aktuelle Messanger-Nachrichten oder E-Mails durch beispielsweise Vibrationsalarm informieren oder an Termine erinnern. Ein weiteres prominentes Beispiel sind die sogenannten Fitness-Tracker. Sie zählen Schritte, messen den Puls, machen Angaben zum Kalorienverbrauch und überwachen ggf. auch den Schlaf.

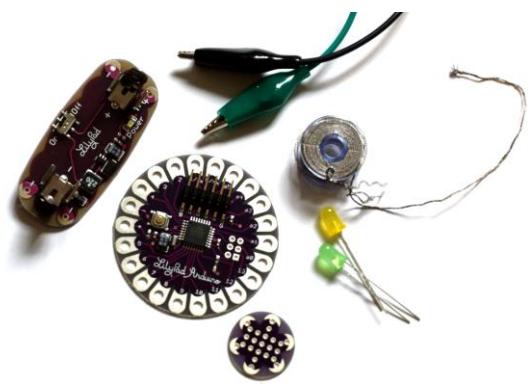
Neben diesen bekannten Anwendungen gibt es aber auch die Möglichkeit, Microcontroller, Sensoren und Aktoren in Kleidung einzunähen und so genannte Smart Fashion oder E-Clothes herzustellen. Die programmierbare Kleidung ist nicht nur eine Modespielerei, sondern findet in vielen Bereichen Anwendung. In der Medizin geben beispielsweise Temperatursensoren Auskunft darüber, ob jemand Fieber hat oder nicht. Die Temperatur kann gemessen werden, ohne die entsprechende Person zum Fiebermessen aufzuwecken. Bei zu hoher Temperatur wird via Smartphone Alarm geschlagen. Dies ist nicht nur für Krankenhäuser interessant sondern auch für Eltern. Genauso wie ein sensorischer Babybody, der Vitalfunktionen misst und so vor dem plötzlichen Kindstod schützen soll.

Für Wanderer und Bergsteiger gibt es Funktionskleidung mit eingenähten GPS-Sendern, so dass sie bei Unfällen direkt lokalisiert werden können. Ein Sport-T-Shirt misst die Herz- und Atemfrequenz, verbrannte Kalorien und zurückgelegte Schritte werden gezählt. Auch eine smarte Feuerwehruniform ist in Entwicklung. Kritische Situationen sollen über die Messung von beispielsweise Herzschlag und Atmung erkannt werden.

Dies sind nur einige Anwendungsbeispiele, die derzeit entwickelt werden oder die es heute schon auf dem Markt zu kaufen gibt.

Um „smarte“ Kleidung zu erstellen, benötigt man verschiedenes Material. Das Herstück ist ein Mikrocontroller, durch den die Kleidung intelligent wird. Auf dem Mikrocontroller kann die Programmierung gespeichert und LEDs oder kleine Motoren angesteuert werden.

Das Arduino LillyPad ist beispielsweise ein Mikrocontroller für Wearables, genauso wie Flora oder Gemma von AdaFruit. Alle Controller sind Open Source und lassen sich daher frei programmieren. Als Programmierumgebung kann Arduino IDE (textuelle Programmierung) genutzt werden oder auch grafische Programmierumgebungen wie Scratch for Arduino oder Ardublocks (siehe 6. Literatur und Links).



Neben einem Mikrocontroller benötigt man außerdem leitfähiges Garn, verschiedene Sensoren (Lichtsensor, Temperatursensor,...), Aktoren (Lautsprecher, Vibrationsboard,...), LEDs, Batterie/ Knopfzelle, Druckknöpfe, ggf. leitfähiger Klettverschluss oder Stoff.

Als Grundlage kann ein einfaches T-Shirt, eine Mütze oder ein Jutebeutel dienen. Jedes Kleidungsstück kann interaktiv und smart gestaltet werden.

5 Unterrichtliche Umsetzung

Zu Beginn der Unterrichtseinheit können verschiedenen Einsatzmöglichkeiten vorgestellt werden, z.B. in der Mode, der Medizin zur Patientenüberwachung oder zur Sicherheit.

Im zweiten Schritt befasst man sich mit den Grundlagen, wie Spannung, Widerstand, LED, Batterie, Reihenschaltung und Parallelschaltung. Hierfür können erste Schaltkizzen angefertigt werden, die dann beispielsweise mit einfachen Krokodilklemmen oder Draht, einer Batterie und LEDs umgesetzt werden.

Im nächsten Schritt kommt der Mikrokontroller dazu, dadurch wird das Kleidungsstück zu Smart Fashion. Um die Programmierung kennenzulernen kann ein einfaches Projekt nach Anleitung umgesetzt werden (z.B. ein Fieberteddy – siehe Links). Viele Projektbeispiele sind im Internet und bei Youtube zu finden.

Im Anschluss daran, kann ein eigenes Projekt umgesetzt werden. Hier kann man den Schülerinnen und Schülern ggf. auch eine Thematik vorgeben, wie beispielsweise „Sicherheit im Straßenverkehr“.

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Einsatzmöglichkeiten kennenlernen, erste Schaltskizzen zeichnen
Einstieg	Wearables selbst gestalten nach Anleitung z.B. Tasche, T-Shirt, Kennenlernen von Arduino-Programmierung/ oder Amici, Edublocks
Vertiefung	Entwicklung einer eigenen Idee (dies kann auch mit Hilfe von M1 - Design Thinking umgesetzt werden)
Vertiefung	Umsetzung eines eigenen Projekts (M2 – Projektmethode)

6 Literatur und Links

- **Arduino** Software zum Programmieren: <https://www.arduino.cc/en/main/software>
- **Grafische Programmierumgebungen für Arduino:**
<https://www.lxrobotics.com/graphische-programmierumgebungen-ide-fuer-arduino>
- **Anleitung für einen Fieber-Teddy:** http://dimeb.informatik.uni-bremen.de/eduwear/wp-content/uploads/manual/anleitung_teddy.pdf
- In die Zukunft mit Klamotten von gestern. **Projektbeschreibung:** <http://pixel-medienwerkstatt.de/Media/Images/Program/Eduwear/Eduwear.pdf>
- Video: **Intelligente Kleidung** - Wie Elektronik in Textilien Einzug hält.
<https://www.youtube.com/watch?v=U5G8EeaMawk>
- Video: **Intelligente Kleidung**. FUTUREMAG – ARTE:
<https://www.youtube.com/watch?v=pzZqz-94ch0>
- Video: **Neue Materialien - was High-Tech-Stoffe alles können:**
<https://www.youtube.com/watch?v=SWKn6WxcIC0>
- Modebeispiel: **Sneaker mit Sensor und LED:**
<https://www.youtube.com/watch?v=gWZi71JkPAA>
- Bohne, René (2012): **Making Things Wearable. Intelligente Kleidung selber schneidern.** O'Reilly Verlag, Köln
- Soft Circuits & Wearables – Gestaltung mit smarten Stoffen. **Projektbeschreibung für die Schule:** <http://www.medien-in-die-schule.de/werkzeugkaesten/werkzeugkasten-diy-und-making/werkzeugportraets-beispiele-aus-der-praxis/soft-circuits-wearables-gestaltung-mit-smarten-stoffen/>

IT2School

Gemeinsam IT entdecken



Modul E3 – Robotik

BB8-Roboter selber bauen

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	Robotik	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Inhalte des Moduls.....	4
5	Unterrichtliche Umsetzung.....	4
6	Literatur und Links	5
7	Arbeitsmaterialien	5

1 Robotik

Star Wars erfreut sich großer Beliebtheit bei Klein und Groß und auch der neue Droide BB8 aus der letzten Star Wars Episode VII hat für große Begeisterung unter den Fans gesorgt.

In diesem Modul erhalten die Schülerinnen und Schüler eine Anleitung, wie der kleine Roboter selbst gebaut werden kann.

Mit Hilfe von verschiedensten Materialien wie Styroporkugeln und Pappmaché, sowie einem BluCoLight-Controller oder einem Arduino-Board kann der Roboter zum Leben erweckt werden. Mit Hilfe einer programmierten App kann der kleine BB8 über das Smartphone gesteuert werden.



Lernfeld/Cluster:	IT selber machen
Zielgruppe/Klassenstufe:	4. bis 5. Klasse
	<input checked="" type="checkbox"/> 6. bis 7. Klasse
	<input checked="" type="checkbox"/> 8. bis 10. Klasse
	<input checked="" type="checkbox"/> 11. bis 12. Klasse
Geschätzter Zeitaufwand:	10 Stunden
Lernziele:	<ul style="list-style-type: none">• Gestalterischen und kreativen Zugang zu IT und Informatik ermöglichen• Handlungsorientiert mit digitalen Technologien umgehen• Vertiefung der Programmierung (mit Arduino)
Vorkenntnisse der Schülerinnen und Schüler:	Empfohlen <ul style="list-style-type: none">• Erfahrung in der Programmierung (ggf. durch die Module B5 und B7)
Vorkenntnisse der/des Lehrenden:	Empfohlen <ul style="list-style-type: none">• Erfahrung in der Programmierung mit Arduino
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters	
Sonstige Voraussetzungen:	Voraussetzung <ul style="list-style-type: none">• Laptops/PCs für die Programmierung

2 Warum gibt es das Modul?

In erster Linie geht es in diesem Erweiterungsmodul um einen kreativen Zugang zur Informatik. Die Schülerinnen und Schüler entdecken einen künstlerisch-kreativen Zugang bei dem sie Spaß haben und sich ausprobieren können. Durch das Basteln mit verschiedenen Materialien werden auch Schülerinnen und Schüler angesprochen, die sonst eher wenig Interesse an Informatik haben. Dabei lernen sie etwas über Elektronik und über die Programmierung mit Arduino.

Darüber hinaus kann das Modul als Einstieg in die Welt der Robotik und die Thematik der Mensch-Maschine-Interaktion genutzt werden.

Das Modul kann besonders gut als Projektwoche angeboten werden. Im Rahmen des gemeinsamen Projektes arbeiten die Schülerinnen und Schüler im Team, erfahren Selbstwirksamkeit und erschaffen ein öffentlichwirksames Produkt, dass bei Schulfesten o.Ä. präsentiert werden kann.

3 Ziele des Moduls

- Gestalterischen und kreativen Zugang zu IT und Informatik ermöglichen
- Vertiefung in die Programmierung (mit Arduino)
- Handlungsorientiert mit digitalen Technologien umgehen
- Selbstwirksamkeit erfahren
- Teamarbeit

4 Inhalte des Moduls

Dieses Modul bietet eine Anleitung zum Bau eines Roboters. Mit Hilfe eines Mikrocontrollers kann der Star Wars-Roboter BB8 nachgebaut werden.

Hierfür kann das BluCoLight aus Modul B7 noch einmal eingesetzt werden, alternativ kann aber auch ein Arduino UNO verwendet werden.

5 Unterrichtliche Umsetzung

Zu Beginn der Unterrichtseinheit wird das vorhaben vorgestellt. Hierfür kann das Video des 17jährigen Angelo Casimiro (<https://www.youtube.com/watch?v=-QbFvDlyy1k>) gezeigt werden.

Im Anschluss daran können sich die Schülerinnen und Schüler in Kleingruppen aufteilen. Für jede Gruppen sollten die benötigten Materialien (E3.1) bereit stehen.

Zum Einstieg wird der Körper des BB8 gebaut. Haben die Schülerinnen und Schüler eigene kreative Ideen, kann der Körper selbstverständlich auch anders angemalt werden, beispielsweise als Schneemann oder Biene etc.

Im zweiten Schritt gibt es eine Einführung in die Technik. Der BluCo oder das Arduino-Board werden vorgestellt und die Programmierung eingeführt. Im Anschluss daran wird das Innenleben zusammengebaut und programmiert.

Am Ende werden alle Komponenten zusammengefügt - hierfür wird das Innenleben in die Styroporkugeln gesetzt.

Der Bau des Roboters ist besonders gut für eine Projektwoche geeignet. Während das Pappmaché oder die Farben für den Körper trocknen, kann man eine Einführung in Arduino geben und sich um die Programmierung kümmern. Sollte das Projekt im regulären Unterricht umgesetzt werden, müssen die Trocknungszeiten mit eingeplant werden.

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Planung des Vorhabens: Was wird benötigt, Gruppeneinteilung, Arbeitsplan erstellen
Einstieg	Körper des BB8 bauen
Vertiefung	Die Technik: Wie funktioniert der BluCo/ Arduino? Wie sieht die Programmierung aus?
Vertiefung	Zusammensetzung aller Komponenten
Abschluss	Präsentation

6 Literatur und Links

- **Eine weitere Anleitung für den BB8:** <http://www.bb8-selber-bauen.net>
- **YouTube Anleitung zum BB8:** <https://www.youtube.com/watch?v=-QbFvDlly1k>
- **Arduino-Tutorial:** <http://www.arduino-tutorial.de>
- **Arduino Programmier-Handbuch:** <http://www.ov-meschede.de/workshop/Arduino%20Programmierhandbuch.pdf>

7 Arbeitsmaterialien

Nr.	Titel	Beschreibung
😊 E3.1	Materialliste	Auflistung aller benötigter Materialien
😊 E3.2	Bauanleitung	Schritt-für-Schritt-Anleitung
😊 E3.3	BB8 Arduino Code	Programmiercode für den Arduino

Legende

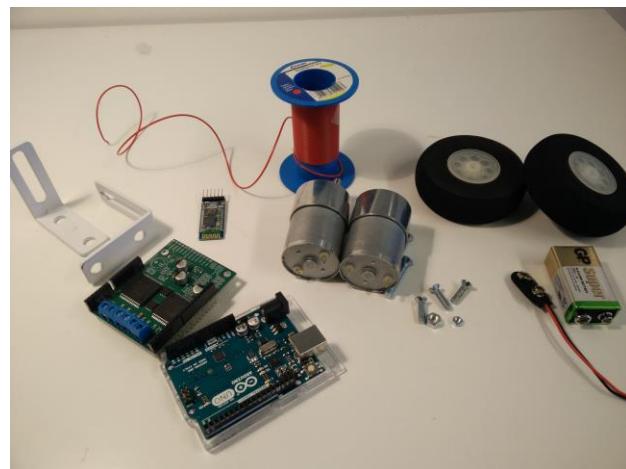
- 😊 Material für Schülerinnen und Schüler
- 😊 Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
- 😊 Zusatzmaterial

Materialliste

Um den BB8 zu bauen, werden einige Dinge benötigt. Manche Materialien können von zu Hause mitgebracht andere müssen zuvor gekauft werden.

Bastelladen:

- 1 x Styroporkugel 50 cm
- 1 x Styroporkugel 30 cm
- 2 Magnetscheiben (Haftkraft mindestens 7 Kg) (z.B. <https://www.magnet-shop.net>)



Aus dem Baumarkt:

- Tapetenkleister
- Malerkrepp
- 9V Batterie/ 12V Batterie
- Farben (Spraydosen oder Abtönfarbe in weiß, orange, grau und schwarz)
- 1 x Hartfaserplatte oder dünne Holzplatte (viele Baumärkte bieten in der Holzabteilung Schnittreste günstig an)
- Rundholz (Durchmesser ca. 5 cm.)
- Holzleiste (30 cm lang, 2 cm breit)
- 2 Winkel
- Schrauben und Muttern (passend zu den Winkeln und der Holzplatte)

Modellbauladen/ Elektroladen (beispielsweise Conrad oder <http://www.exp-tech.de> oder <https://nodna.de>)

- Metallgetriebemotor (z.B. <http://www.exp-tech.de/19-1-metal-gearmotor-37dx52l-mm>; <https://www.conrad.de/de/getriebemotor-12-v-modelcraft-rb350100-0a101r-1100-227560.html?sc.ref=Product%20Details>)
- Halterung/ Winkel für Motor (<http://www.exp-tech.de/pololu-stamped-aluminum-l-bracket-pair-for-37d-mm-metal-gearmotors>)
- Modell-Flugzeugräder ca. 7 cm Durchmesser

Tipp: Lassen Sie sich am Besten im Laden alles drei zusammenstellen, dann können Sie sicher sein, dass es zusammen passt.

- BluCoLight (Modul B7)

Alternativ zum BluCoLight kann auch ein Arduino verwendet werden:

- Arduino Uno Rev 3
- Arduino Netzteil-Batterie-Adapter
- HC-05 Bluetooth-Modul für Arduino (es gibt verschiedene günstige Anbieter, am besten eines mit Kabeln bestellen)
- Pololu Dual VNH5019 Motor Driver Shield (Motortreiber zur Verbindung von Arduino und Motoren)

Werkzeug:

- Pinsel
- Feile
- Schleifpapier
- Stichsäge oder Lasercutter
- Akkuschrauber, Schraubenzieher
- Zirkel
- Lineal
- Bleistift
- Heißklebepistole

Verbrauchsmaterialien:

- Alte Zeitungen
- Leere Deoroller aus Kunststoff

Bauanleitung

Der BB8 besteht außen aus einer großen und einer kleinen Styroporkugel, die mit Pappmaché ummantelt werden. Im Inneren befindet sich ein Microcontroller mit Getriebemotoren und ein Bluetoothmodul für die Fernsteuerung. Die benötigten Materialien sind in E3.1 aufgelistet. Es folgt eine Schritt-für-Schritt-Anleitung für die einzelnen Elemente.

Zur Einführung empfehlen wir darüber hinaus folgendes Video:

<https://www.youtube.com/watch?v=-QbFvDIyy1k>

Schritt 1: Der Körper des BB

Zu Beginn nimmt man die große Styroporkugel (ca. 50 cm) und feilt auf der Innenseite die Erhebungen ab, damit später die Räder des Innenlebens nirgends hängen bleiben.



Im Anschluss daran, wird die Innenseite als auch die Außenseite mit Zeitungspapier beklebt. Dadurch bekommt die Kugel eine größere Stabilität. Hierfür werden die Zeitungen in Streifen gerissen und Schicht für Schicht mit Kleister aufgetragen. Man sollte dabei darauf achten, dass die Zeitung immer gut angedrückt wird, damit später keine Luftpolster entstehen.



Nach dem Trocknen wird die Kugel grundiert und weiß bemalt. Auf die weiße Kugel können nun die Details aufgemalt werden. Hierfür sollte man einen Zirkel verwenden. Mit einem dünnen Pinsel und Abtönfarbe können dann die Flächen ausgemalt werden. Möchte man die Farbe mit Hilfe einer Spraydose aufbringen, müssen die Flächen sauber mit Malerkrepp abgeklebt werden.



Genauso verfahren wir auch mit dem Kopf, allerdings benötigen wir hier nur die halbe Kugel. Selbstverständlich kann man den Körper des Roboters auch ganz anders bemalen z.B. als Biene oder Käfer.

Schritt 2: Die Technik

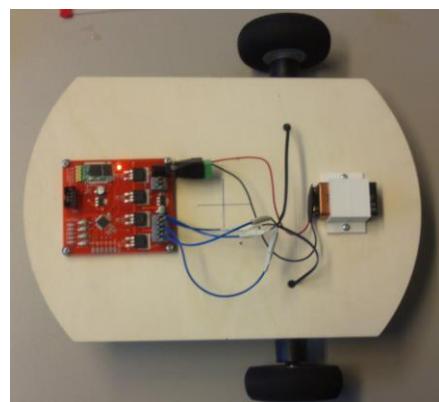
Für die Technik gibt es zwei Möglichkeiten. Man kann für den Roboter das BluCoLight aus dem Modul B7 verwenden oder einen Arduino-Controller. Beide Varianten werden im Folgenden beschrieben.

Möglichkeit 1: BluCoLight-Anleitung

An den BluCo werden die beiden Getriebemotoren angeschlossen sowie eine 9V-Batterie.

Achte dabei auf folgende Verbindung:

- Beide Motoren müssen in die 12V-Steckverbindung (am Besten die beiden Kabelenden etwas gemeinsam eindrehen, dann lassen sie sich leichter in die entsprechende Steckverbindung stecken)
- jeweils ein Motor direkt daneben in die beiden Farben Grün und Rot.



Im Anschluss daran können die Getriebemotoren mit Hilfe passender Halterungen an einer Holzplatte befestigt werden.

Die Holzplatte sollte zuvor zurechtgeschnitten werden (**ca. 30x20cm**)

Die Platte kann mit Hilfe einer Stichsäge oder eines Lasercutters ausgesägt werden. Die Ränder können anschließend mit etwas Schleifpapier sauber geschliffen werden.

Motor ansteuern

Um den Motor anzusteuern, kann nun eine App für das Smartphone verwendet werden. Öffne dazu den AppInventor und suche in der Gallery nach der App: **BB8_Control_B**

Alternativer Link zur App: ai2.appinventor.mit.edu/?galleryId=5695318742794240

Nun kann man sich mit Hilfe des AI Companion (Siehe nähere Hinweise Modul B7) die App auf das eigene Smartphone bringen und via Bluetooth die Motoren ansteuern. Das kleine Holzgefäß sollte nun fahren und mit dem Smartphone gesteuert werden können.

Zum Testen kann das Gefährt auch in die große Styroporkugel gesetzt werden. Diese sollte sich nun mit Hilfe der App steuern lassen.

Je nach Zeit und Vorerfahrung können die Schülerinnen und Schüler das Programm mit Hilfe des AppInventors auch selbst schreiben oder anpassen.

Möglichkeit 2: Arduino-Anleitung (Fortgeschritten)

Zu Beginn muss die Arduino-Software heruntergeladen werden, damit das Programm für den BB8 auf den Mikrocontroller hochgeladen werden kann.

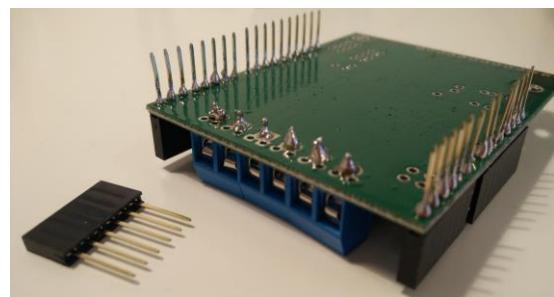
Download: <https://www.arduino.cc/en/Main/Software>

Nach dem Download kann die Programmierung für den BB8 (E3.3 - *BB8s_Arduino_Code*) geöffnet werden.

Je nach Zeit und Vorerfahrung können die Schülerinnen und Schüler das Programm auch selbst mit Anleitung schreiben.

Der Arduino-Controller wird dann mit dem PC verbunden, um die Programmierung hochzuladen.

Im Anschluss daran wird der Motorshild mit dem Arduino verbunden. Dazu müssen allerdings erst Steckverbindungen an den Motorshild gelötet werden. Die Steckverbindungen sind beim Kauf dabei. Dies ist eine Aufgabe für Fortgeschrittene, die schon etwas Erfahrung im Löten haben.



Danach kann das Motorshild auf den Arduino Uno draufgesetzt werden.



Im Anschluss daran wird das Bluetooth-Modul angebracht. Achte dabei auf folgende Verbindung:

1. Arduino RC mit Bluetooth TX
2. Arduino TX mit Bluetooth RX
3. Arduino 5V an Bluetooth 5V
4. Arduino GND mit Bluetooth GND

Im nächsten Schritt werden die Motoren mit dem Motorshild verbunden.

1. Motor 1 (Plus) mit **M1A** des Motorshield verbinden
2. Motor 1 (Minus) mit **M1B** des Motorshield verbinden
3. Motor 2 (Plus) mit **M2A** des Motorshield verbinden
4. Motor 2 (Minus) mit **M2B** des Motorshield verbinden

Die Technik testen



Wenn alles miteinander verbunden ist, kann nun eine Stromquelle an den Arduino angeschlossen und getestet werden, ob sich die Motoren steuern lassen. Dazu sollte eine App auf das Smartphone geladen werden.

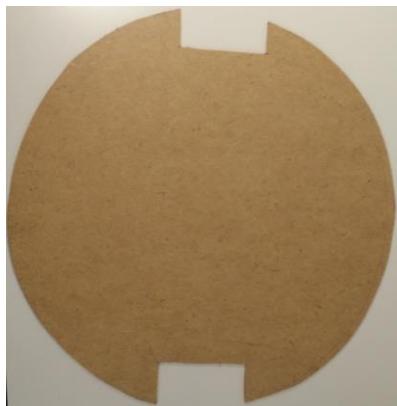
Download: Arduino Bluetooth RC Car:

<https://play.google.com/store/apps/details?id=braulio.calle.bluetoothRCcontroller&hl=de> (App im Google Play Store)

Wenn der Arduino am Strom angeschlossen ist sollte das Bluetooth-Modul blinken.

Nun kann via App eine Bluetooth-Verbindung aufgebaut werden. Wenn alles richtig funktioniert, können sich die Motoren mit dem Smartphone bedienen lassen. Falls das nicht der Fall ist, sollten alle Verbindungen noch einmal überprüft werden.

Das Innenleben zusammenbauen

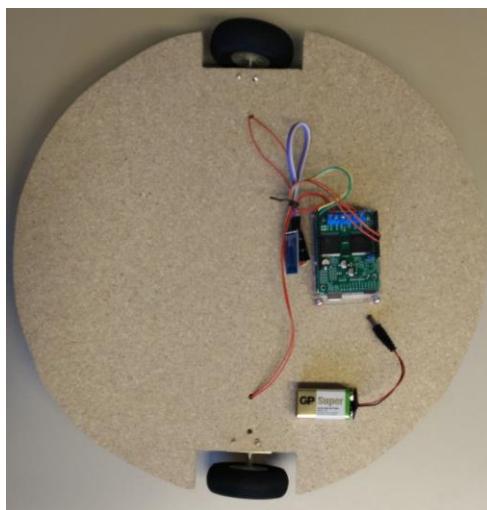
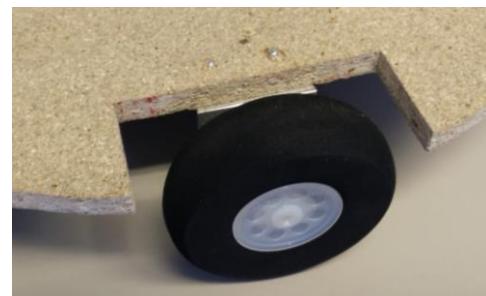


Auch für den Arduino wird eine Holzplatte benötigt: Für das Innenleben wird ein Kreis mit ca. 42 cm. Durchmesser ausgeschnitten. An zwei Seiten werden Aussparungen für die Räder aufgezeichnet. Die Platte kann mit Hilfe einer Stichsäge oder eines Lasercutters ausgesägt werden. Die Ränder können anschließend mit etwas Schleifpapier sauber geschliffen werden.

An die Aussparungen werden die Halterungen für die Motoren montiert.

Im Anschluss daran können die Motoren sowie die Räder an der Halterung befestigt werden.

Im nächsten Schritt wird das Arduino-Board in der Mitte der Holzplatte mit Platinen-Abstandshaltern festgeschraubt und der Motortreiber daraufgesetzt.



Der Motortreiber wird nun mit den Motoren via Kabel verbunden. Der Arduino benötigt noch eine Stromquelle, dafür wird eine 9V- Batterie angeschlossen.

Dieses Holzgefährt kann nun schon getestet werden. Mit Hilfe der App aus dem Google PlayStore sollte der Roboter sich steuern lassen. Auch kann der Roboter schon in die große Holzkugel gesetzt werden. Diese kann sich nun mit dem Smartphone bedienen lassen.



Um der Holzplatte in der Kugel eine bessere Stabilität zu geben, benötigen wir nun die leeren Deoroller. Diese werden an den Seiten der Holzplatte mit Schrauben angebracht.

Schritt 3: Der Kopf (BluCoLight und Arduino)

Der Kopf des BB8 wird mit Magneten gehalten. Hierfür wird ein Holzstab auf ca. 25 cm Länge geschnitten. Das Holz wird dann mit Winkeln an die Holzplatte des Mikrocontrollers befestigt. Am anderen Ende wird ein sehr starker Magnet (oder mehrere Kleine) mit Hilfe einer Heißklebepistole festgeklebt.



Achte darauf, dass das Holz bei geschlossener Styropor-Kugel nicht am Styropor schleift. Der Abstand darf aber auch nicht zu groß sein, da sonst die Magneten nicht mehr haften.



Das Gegenstück befindet sich im Kopf des BB8. Im Kopfteil wird ebenfalls eine Holzleiste mit Magneten befestigt (Heißklebepistole). Schließt man nun die große Kugel, kann man die halbe Kopf-Styroporkugel darauf setzen und sie wird durch die Magneten oben gehalten.



IT2School

Gemeinsam IT entdecken



Modul E4 – Webseiten

Erstellung von Webseiten

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	Webseiten	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Inhalte des Moduls.....	4
4.1	Hypertext-Markup-Language.....	4
4.2	Cascading Style Sheets.....	5
4.3	Beispielhaftes HTML/CSS-Projekt.....	6
4.4	JavaScript	9
4.5	Vorstellung verschiedener Editoren.....	10
5	Unterrichtliche Umsetzung.....	13
6	Literatur und Links	13

1 Webseiten

In diesem Modul lernen die Schülerinnen und Schüler *Markup-Languages* (ML) – im Deutschen auch als Auszeichnungs- oder Beschreibungssprachen¹ bezeichnet – exemplarisch an der *Hypertext Markup Language* (HTML) kennen, die das Gerüst von Webseiten ist. Dazu nutzen sie Online-Editoren, um in projektorientiertem Unterricht eigene einfache Projekte zu entwickeln.

Während der Nutzung der in diesem Erweiterungsmodul vorgestellten Online-Editoren² (Abs. 4.5) steht den Schülerinnen und Schülern eine Live-Vorschau der von ihnen erarbeiteten HTML-Seiten zur Verfügung. Dies bietet den Vorteil, dass sie nicht nur unter Anleitung, sondern auch experimentativ vorgehen können.

Lernfeld/Cluster:	IT spielend entdecken
Zielgruppe/Klassenstufe:	4. bis 5. Klasse
	X 6. bis 7. Klasse
	X 8. bis 10. Klasse
	X 11. bis 12. Klasse
Geschätzter Zeitaufwand:	8 – 10 Stunden
Lernziele:	<ul style="list-style-type: none">• Kennenlernen von Markup-Languages• Darstellen von Informationen mittels Markup-Languages• Kennenlernen und Verwendung von Cascading Style Sheets (CSS)• Kennenlernen von JavaScript als Scriptsprache zur clientseitigen Programmierung• Programmierung einfacher Programme mittels JavaScript
Vorkenntnisse der Schülerinnen und Schüler:	Keine
Vorkenntnisse der/des Lehrenden:	Keine
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters:	Keine
Sonstige Voraussetzungen:	Keine

¹ Auszeichnungs- bzw. Beschreibungssprachen unterscheiden sich von Programmiersprachen dahingehend, als dass sich mit letzteren Scripts oder Programme erzeugen lassen, mit ersteren hingegen Dokumente.

² Namensgebend für dieses Erweiterungsmodul war bei der Einführung von IT2School der einfach zu bedienende Online-Editor zur Erstellung von Webseiten – *Thimble* (engl. für „Fingerhut“). Zwar wird das Open-Source-Projekt seit 2019 von Mozilla leider nicht fortgesetzt, das Git-Repository ist aber noch aufrufbar: <https://github.com.mozilla/thimble.mozilla.org>

2 Warum gibt es das Modul?

In diesem Modul können sich Schülerinnen und Schüler kreativ entfalten und letztlich ein Produkt entstehen lassen, das starken Alltagsbezug hat und ihre Lebenswelt aufgreift. Die Nutzung des World Wide Webs und die Möglichkeit, sich online zu präsentieren - auf z.B. verschiedenen sozialen Netzwerken wie Facebook, Instagramm oder Ähnlichem - löst großes Interesse für die Gestaltung eigener Webseiten aus.

In diesem Modul können die Schülerinnen und Schüler hinter die Kulissen der Entstehung von Websites schauen. Dabei entwickeln sie ein Verständnis, wie Webseiten aufgebaut sind. Sie erfahren mehr über die Grundlagen und erlernen die Basiskenntnisse von HTML. Durch die Erstellung eigener Webseiten, erleben sie das Internet als etwas, dass sie aktiv mitgestalten können.

3 Ziele des Moduls

- Kennenlernen von Markup-Languages (im speziellen HTML)
- Darstellen von Informationen mittels Markup-Languages (im speziellen mit HTML)
- Kennenlernen und Verwendung von Cascading Style Sheets (CSS)
- Kennenlernen von JavaScript als Scriptsprache zur clientseitigen Programmierung
- Programmierung einfacher Programme mittels JavaScript

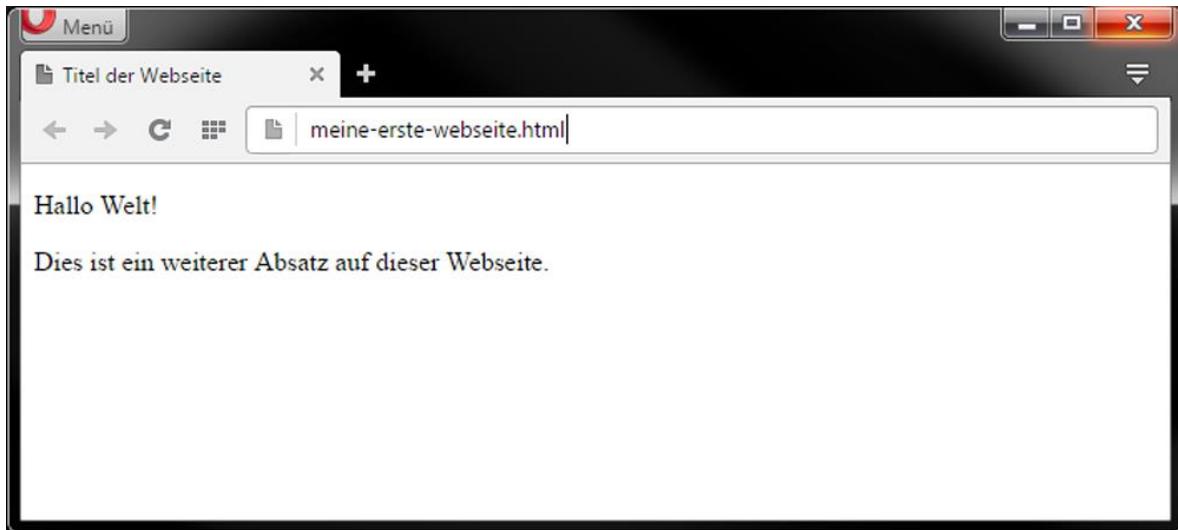
4 Inhalte des Moduls

4.1 Hypertext-Markup-Language

Die *Hypertext-Markup-Language* (kurz HTML) ist eine besondere Form der textbasierten Auszeichnungssprachen zur Strukturierung von Texten inklusive *Hyperlinks*, Bildern sowie anderen Inhalten. Grundlage des *World Wide Webs* bilden die Webseiten in Form von HTML-Dokumenten, die auf einen festen Standard des *World Wide Web Consortiums* (W3C) basieren. Ähnlich zu anderen Auszeichnungssprachen (Markup-Languages) wie *XML* gliedert sich ein HTML-Dokument durch verschiedene Tags. Dem folgenden Beispiel ist die Grundstruktur einer Webseite zu entnehmen. Sie gliedert sich in einen *Head-* und *Body-*Bereich.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titel der Webseite</title>
  </head>
  <body>
    <p>Hallo Welt!</p>
    <p>Dies ist ein weiterer Absatz auf dieser
       Webseite.</p>
  </body>
</html>
```

Erkennbar sollte sein, dass der Head-Bereich der Webseite durch das `<head> ... </head>` eingegrenzt wird sowie der Body-Bereich durch das `<body> ... </body>`. Im Head-Bereich stehen die wichtigen Meta-Informationen, wie beispielsweise der Titel der Webseite, der auch im Browser steht oder Angaben zum Autor. Im Body-Bereich steht dann der eigentliche Text der Webseite. Wie das obige HTML-Dokument im Browser dargestellt werden würde, zeigt folgendes Bild.



Für die Schülerinnen und Schüler kann es auch von Interesse sein, den Quellcode einer bekannten Webseite wie Google, Facebook oder Youtube zu öffnen. Dies ist mit jedem Browser mittels eines simplen Rechtsklicks möglich und zeigt, wie komplex und für den Menschen unleserlich dieser Quellcode werden kann.

Weitere Anlaufstellen, um sich tiefergehend mit HTML zu befassen und einzuarbeiten, finden Sie im Abschnitt „Literatur und Links“.

4.2 Cascading Style Sheets

Die Cascading Style Sheets (CSS-Dokumente), die in die HTML-Dokumente eingebunden werden können, dienen zur Formatierung einzelner und zur Gruppierung von Tags. Mit ihnen können zum Beispiel Schrift, Farbe, Position usw. beschrieben werden.

Der folgende Auszug zeigt ein Beispiel zur Formatierung von Absätzen (p Tags), bei dem die Schriftgröße auf 14px und Schriftfarbe auf rot gesetzt wird.

```
P {  
    font-color: red;  
    font-size: 14px;  
}
```

Für eine weitere Vertiefung mit diesem Thema sei ebenfalls auf den Abschnitt *Literatur und Links* verwiesen.

4.3 Beispielhaftes HTML/CSS-Projekt

Als Minimalbeispiel schauen wir uns im Folgenden den Quelltext³ der fiktiven Seite „We <3 IT2School“ (siehe rechter Screenshot) etwas genauer an:



³ Um besser Bezug nehmen zu können, sind hier Zeilennummern vorangestellt (grau).

```

1 <!doctype html>
2 <html lang="de-DE">
3
4   <head>
5     <meta charset="utf-8">
6     <title>We love IT2School!</title>
7     <link rel="stylesheet" href="style.css">
8   </head>
9
10  <body>
11
12    <div id="box">
13      <h1>We ❤ IT2School!</h1>
14      
16      <h2>Würde diese Person existieren, dann würde sie IT2School
17        sicher genauso lieben wie wir.</h2>
18      <p>Moment mal: Wieso "würde"? Die oben abgebildete Person
19        existiert nämlich gar nicht. Sie entstammt einem
20        Algorithmus, der gelernt hat, aus Unmengen vorhandener
21        Bilddateien künstlich Bilder von Personen zu erzeugen. Das
22        Tool <a href="https://thispersondoesnotexist.com">
23          thispersondoesnotexist</a> und noch viel mehr lernst du in
24        den KI-Modulen von <a href="https://www.wissensfabrik.de/
25          it2school/">IT2School</a> kennen!</p>
26    </div>
27
28    
30
31  </body>
32 </html>

```

Im Head-Bereich (Z. 4-9) wird zunächst die verwendete Zeichencodierung (UTF-8) angegeben. Dadurch wird sichergestellt, dass die Zeichen im Dokument richtig interpretiert werden⁴. Anschließend folgt die Angabe des Titels der Seite, der bspw. von manchen Browsern in der Fensterleiste übernommen wird (Hinweis: Das meint hier nicht die Überschrift der eigentlichen Webseite, die in Z. 14 angegeben ist!).

Im Body-Bereich (Z. 11-39) folgt dann zunächst eine Aufteilung (engl. „division“) mittels eines `<div>`-Tags. Solche „Tags“ (engl. für „Etikett“) sind Behälter für HTML-Elemente, die sich wiederum mit CSS gestalten (vgl. Abs. 4.2) oder mit JS manipulieren (vgl. folgender Abs. 4.4) lassen. Eigenschaften zu `box`, `h1`, `h2`, `p` und `a`⁵ werden im CSS spezifiziert.

Erwähnenswert ist die Einbettung eines zufällig generierten Bildes einer nicht-realnen Person in Z. 14 von der Seite thispersondoesnotexist.com sowie das Logo der Wissensfabrik, das über Z. 19 eingebettet wird.

Hinweis: Im Basismodul KI-B1 lernt ihr weitere Beispiele von Künstlicher Intelligenz (KI) kennen!

⁴ Warum es wichtig ist, sich mit der Zeichencodierung zu beschäftigen, wird bspw. unter <https://www.w3.org/International/questions/qa-what-is-encoding> anschaulich erklärt. Relevanz erfährt die Zeichencodierung im vorliegenden Dokument bspw. bei den Umlauten oder auch dem ❤ in Z. 14.

⁵ Mit `<a>`-Tags werden üblicherweise Hyperlinks bezeichnet, `<p>` wird für Absätze (engl. „paragraph“) und `<h>` für Überschriften (engl. „heading“) verwendet.

Im digitalen Archiv findet sich das HTML-Dokument, das entweder zur Ansicht in einem Browser (bspw. Firefox) oder zur Bearbeitung in einem Texteditor (Bsp. werden im folgenden Abs. 4.5 gegeben, grundsätzlich eignet sich jedoch auch ein systemseitig vorinstallierter Texteditor wie Wordpad) geöffnet werden kann.

Dort wird das CSS über direkt in den Header eingebaut:

```
1 <!doctype html>
2 <html lang="de-DE">
3
4     <head>
5         <meta charset="utf-8">
6         <title>We love IT2School!</title>
7         <style type="text/css">
8             * { font-family: Arial, sans-serif; }
9             html, body { min-height:90%; min-width:90%; }
10            body { background:rgb(237, 244, 250); text-align:center; }
11            #box { padding:20px; margin:30px auto; max-width:70%; text-
12                align:center; border-radius:20px; background-
13                    color:#fff; color:rgb(111,111,111); }
14            a { color:#f5a700; }
15            p { margin-top:10px; }
16            h1 { color:#f5a700; font-size: 60px; }
17            h2 { color:rgb(111, 111, 111); font-size: 30px; }
18        </style>
19     </head>
20
21     <body>
22
23         [...]
24
25     </body>
26 </html>
```

Insbesondere bei größeren Webseiten-Projekten ist es üblich, die CSS-Datei(en) stattdessen separat anzulegen (Suffix „.css“) und im HTML-Dokument zu verlinken. Das funktioniert wie folgt:

```
1 <!doctype html>
2 <html lang="de-DE">
3
4     <head>
5         <meta charset="utf-8">
6         <title>We love IT2School!</title>
7         <link rel="stylesheet" href="style.css">
8     </head>
9
10    <body>
11
12    [...]
13
14    </body>
15 </html>
```

Das eigentliche HTML-Dokument lässt sich so übersichtlicher gestalten. Außerdem ermöglicht dieses Vorgehen, dass dasselbe CSS redundanzfrei auch in anderen HTML-Dokumenten verwendet werden kann.

Die Lesbarkeit von CSS wird durch Zeilenumbrüche und Tabulatoren massiv erleichtert. Online stehen frei zugängliche Tools zur Verfügung, bspw. unter <https://www.freeformatter.com/css-beautifier.html> (für HTML: <https://www.freeformatter.com/html-formatter.html>). Das verwendete CSS lässt sich so wie folgt strukturieren.

```
* {  
    font-family: Arial, sans-serif;  
}  
html, body {  
    min-height: 90%;  
    min-width: 90%;  
}  
body {  
    background: rgb(237, 244, 250);  
    text-align: center;  
}  
#box {  
    padding: 20px;  
    margin: 30px auto;  
    max-width: 70%;  
    text-align: center;  
    border-radius: 20px;  
    background-color: #fff;  
    color: rgb(111, 111, 111);  
}  
a {  
    color: #f5a700;  
}  
p {  
    margin-top: 10px;  
}  
h1 {  
    color: #f5a700;  
    font-size: 60px;  
}  
h2 {  
    color: rgb(111, 111, 111);  
    font-size: 30px;  
}
```

4.4 JavaScript

JavaScript ist von der Programmiersprache Java abgrenzen. Es bestehen nur wenige Gemeinsamkeiten. Bei JavaScript handelt es sich um eine Scriptsprache, das bedeutet, dass die Programme/Scripte und Befehle nicht (wie bei Java, C++ und anderen) bereits vorher in Maschinencode vorhanden sind. Bei einer Scriptsprache werden diese beim Lesen/Aufruf mittels Browser von einem Interpreten ausgeführt. Dies hat den Vorteil, dass die Scripte clientseitig ausgeführt werden.

Mit JavaScript lassen sich dynamische HTML-Dokumente erzeugen. So lassen sich durch einen Knopfdruck auf einer Webseite bestimmte Formatierung verändern, Berechnungen ausführen oder bestehendes HTML und CSS erweitern. Verwendet werden können viele bekannte Elemente aus anderen Programmiersprachen wie Kontrollstrukturen (if-else), Schleifen, Variablen usw.

Das folgende Beispiel zeigt wie der String "Hallo Welt" in einer Variablen gespeichert und mittels alert-Befehl als Dialogbox im Browser ausgegeben wird.

```
var variable = "Hallo Welt";  
alert(variable);
```

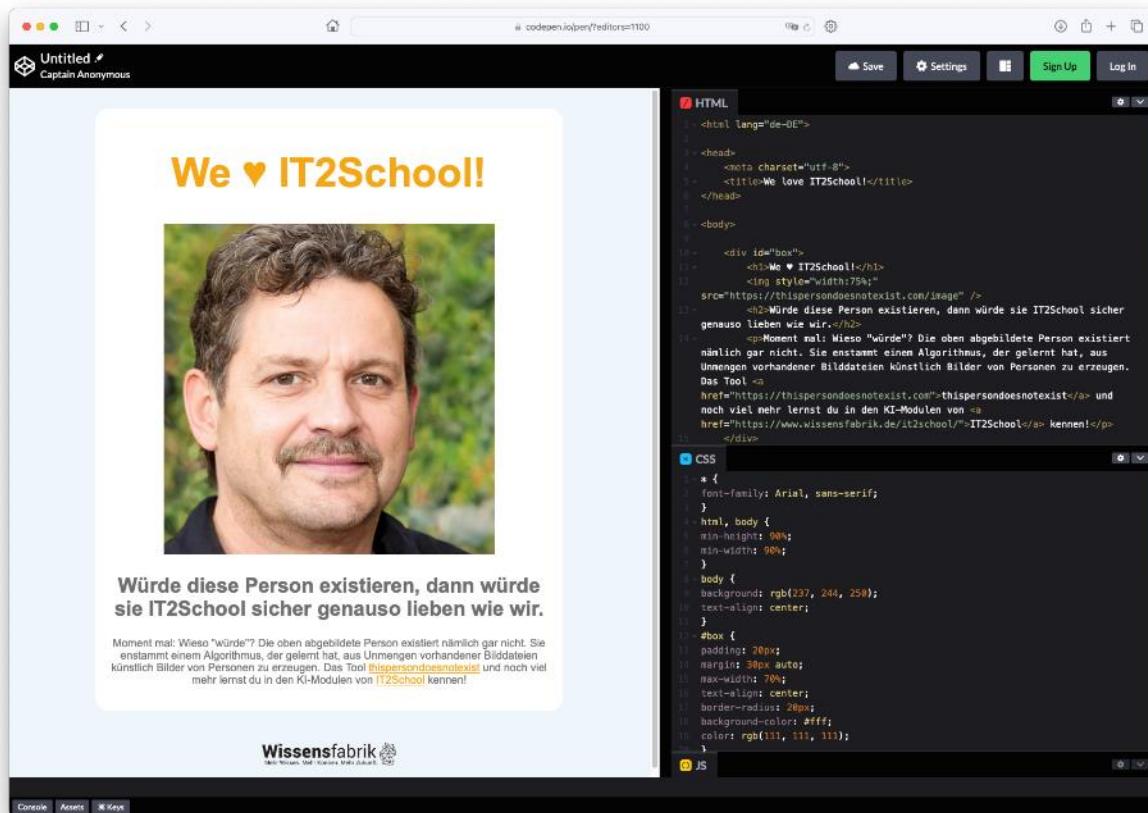
Für eine weitere Vertiefung in dieses Thema sei ebenfalls auf den Abschnitt *Literatur und Links* verwiesen.

4.5 Vorstellung verschiedener Editoren

Kern dieses Erweiterungsmoduls ist die Arbeit mit Online-Quelltext-Editoren zur Entwicklung einfacher, eigener Webseiten. Um eine Vorschau in Echtzeit zu ermöglichen, hat man bei all den drei im Folgenden vorgestellten Editoren sowohl den Quelltext als auch die Webseite gleichzeitig im Blick. Wenn man also etwas im Quelltext ändert, sieht man direkt die Veränderung.

4.5.1 Codepen.io

Der Editor **Codepen.io** ist unter dem Link <https://codepen.io/pen/> zu finden und in einem beliebigen Browser einsetzbar. Auf der Startseite findet sich die Möglichkeit, sich mit Nutzerdaten anzumelden und Projekte online abzulegen. Alternativ kann man den Dienst auch ohne Konto nutzen, indem man auf „Start Coding“ klickt oder direkt <https://codepen.io/pen/> aufruft.



Startet man den Editor so ist er wie folgt aufgebaut: Im oberen Bereich finden sich drei Textfelder: Eines für den HTML-Code, einen für das CSS und einen für JS. Mit einem Klick auf die entsprechende Schaltfläche in der oberen rechten Ecke lassen sich Editoren und Vorschaufenster auch nebeneinander anordnen (wie im Screenshot).

Unser Beispielprojekt kommt ohne JS aus, daher können wir den dritten Editor zur besseren Übersichtlichkeit minimieren (über einen Klick auf den entsprechenden Pfeil und eine Auswahl des „minimize“-Befehls). So entsteht ein insgesamt **sehr übersichtlicher Editor**.

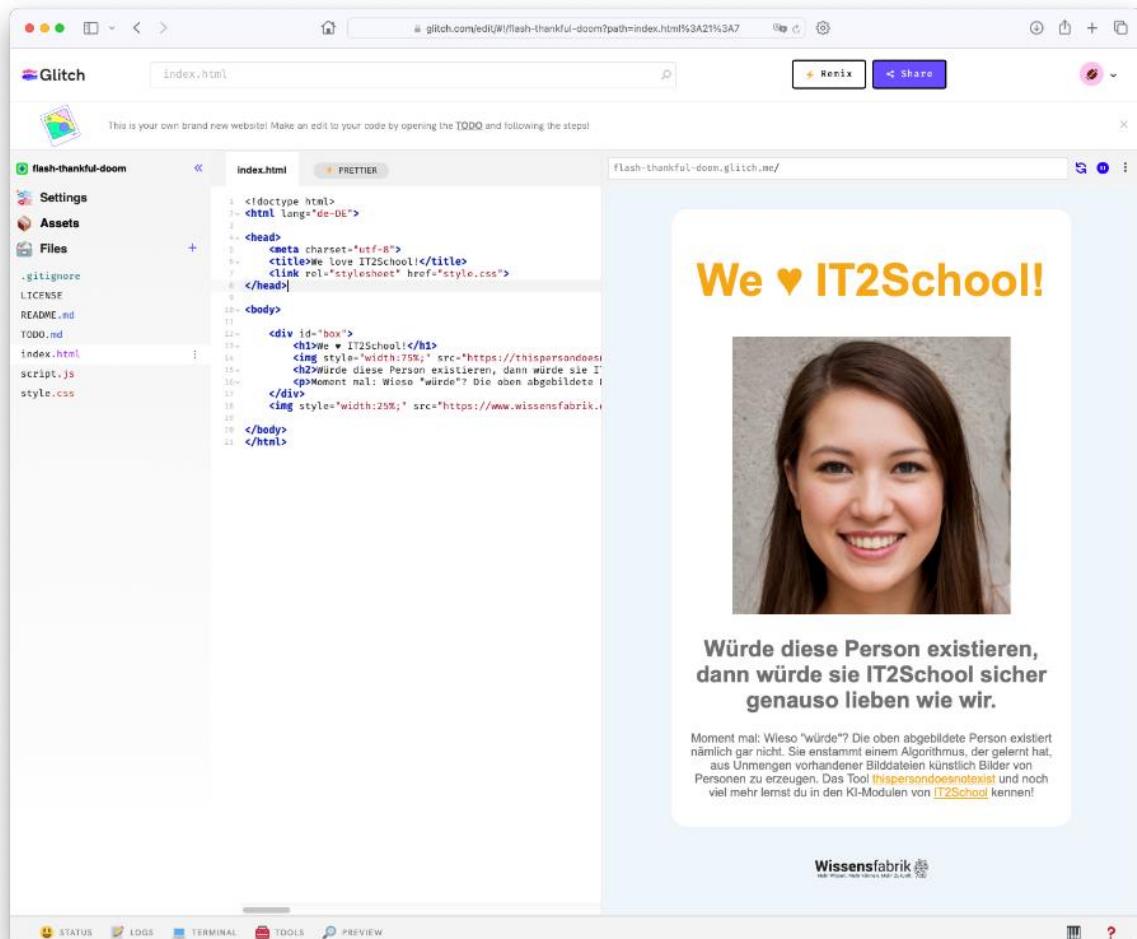
Ein weiterer Vorteil von Codepen.io ist, dass eine **Registrierung nicht zwingend notwendig** ist. Jedoch ist es empfehlenswert, einen eigenen Account anlegen. Dies ermöglicht es,

Projekte online zu speichern und später an ihnen weiterzuarbeiten, ohne die Dokumente lokal zwischenspeichern zu müssen.

Ein Nachteil kann sich aus datenschutzrechtlicher Sicht daraus ergeben, dass der Dienst **nicht in Deutschland gehostet** wird und das Tool **nur mit vorhandener Internetverbindung nutzbar** ist.

4.5.2 Glitch

Nachdem Thimble 2019 eingestellt wurde, wurde **Glitch** von Mozilla als Möglichkeit zur Migrierung von Thimble-Projekten angegeben. Wie auch Codepen.io ist Glitch online in den gängigen Browsern einsetzbar: <https://glitch.com/> Im Gegensatz zu Codepen.io ist bei Glitch jedoch die Anlegung eines Nutzerkontos oder der Login über einen vorhandenen Facebook-, GitHub- oder Google-Account unumgänglich. Zu empfehlen ist das Anlegen eines Kontos über die Verknüpfung mit einer E-Mail-Adresse, an die zur Verifikation ein Code geschickt wird. Ggf. eignen sich hierfür Wegwerf-E-Mail-Adressen, deren Nutzung mit den Schülerinnen und Schülern in diesem Zusammenhang diskutiert werden könnte.



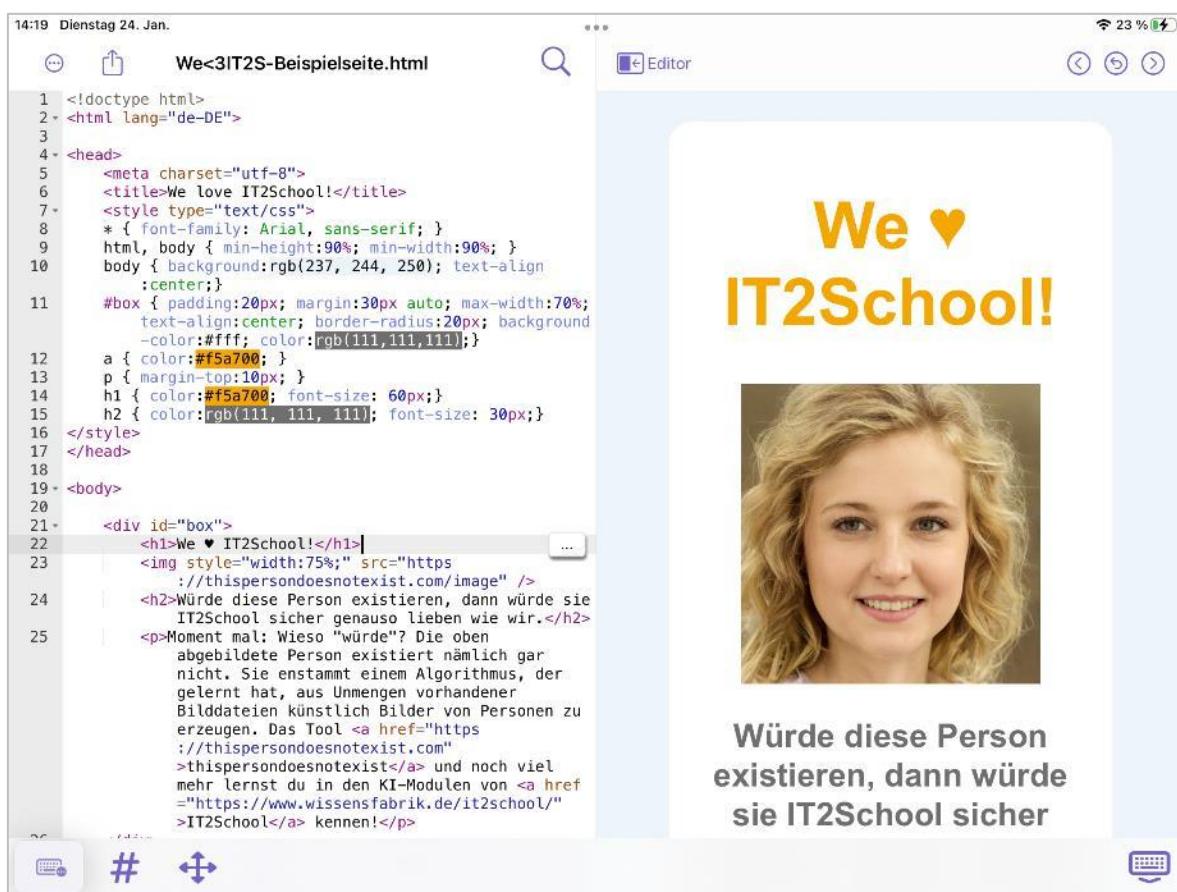
Nachdem der Prozess des Anlegens eines Nutzerkontos und der Login gemeistert sind, lässt sich in Glitch ein neues Webseiten-Projekt anlegen. Neben den bekannten HTML-, CSS- und JS-Dateien werden auch zwei MD-Dateien angelegt, die zu vernachlässigen sind. Das Projekt ist initial mit Beispielcode gefüllt, der durch eigene Projekte ersetzt werden kann. Über einen

Klick auf „Preview“ lässt sich das Vorschaufenster entweder als Panel im aktuellen Fenster ergänzen oder in einem separaten Fenster öffnen.

Ein Vorteil von Glitch ist die **integrierte „Prettier“-Funktion** (siehe Abs. 4.3). Ein Nachteil kann sich wie bei Codepen.io aus der Tatsache ergeben, dass der Dienst nicht in Deutschland gehostet wird. Anders als bei Codepen.io ist bei Glitch das **Anlegen eines Nutzerkontos verpflichtend**.

4.5.3 HTML Editor (iPad & Mac App)

iPads finden in Schulen zunehmend Verbreitung. Exemplarisch für die zahlreichen Möglichkeiten soll hier der **HTML Editor** (kostenlos als [iPad App](#) sowie auch als [Mac App](#) verfügbar) vorgestellt werden.



Ein Vorteil dieser und vergleichbarer Apps ist, dass sie **ohne Anlegen eines Kontos** funktionieren und **nativ auf den Geräten** laufen, d. h. die Inhalte der Webseiten-Projekte bleiben bei vielen dieser Apps auf dem Gerät.

5 Unterrichtliche Umsetzung

Unterrichtsszenarien	Kurze Zusammenfassung
Einstieg	Präsentation einer Webseite per Beamer, Betrachtung des Aufbaus und des Quellcodes, Tags (<html> </html>)
Vertiefung	Umsetzung kleinerer Aufgaben und Übungen (z.B. von Mozilla Teaching Activities oder Lehrer-Online.de)
Vertiefung	Umsetzung einer eigenen Webseite
Präsentation	Präsentation und Veröffentlichung im Netz der entstandenen Webseiten

6 Literatur und Links

- HTMLing.net – **Online-Kurs für Kinder**: <http://www.htmling.net/>
- Informatik-Zentrale - Erste HTML-Befehle:<http://www.informatikzentrale.de/html-erste-befehle.html>
- **Inf-Schule – das elektronische Schulbuch**: <http://www.inf-schule.de/information/informationsdarstellunginternet>
- Self-HTML – **HTML-Wiki** zum Nachschlagen, Austauschen und Lernen
<https://wiki.selfhtml.org/>

IT2School

Gemeinsam IT entdecken



Modul M1 – Design Thinking

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

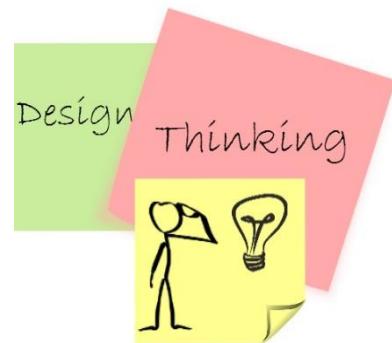
1	Design Thinking	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters.....	4
5	Inhalte des Moduls.....	5
5.1	Der Design-Thinking-Prozess	5
5.2	Warum Design Thinking?.....	7
6	Unterrichtliche Umsetzung.....	7
6.1	Allgemeine Tipps.....	8
6.2	Grober Unterrichtsplan.....	8
6.3	Stundenverlaufsskizzen	10
6.3.1	Smarte Technologien – Einführung.....	10
6.3.2	Smart School – Design Thinking	10
6.3.3	Unterrichtseinheit – Bau eines digitalen Prototyps	12
7	Einbettung in verschiedene Fächer und Themen	13
8	Anschlussthemen.....	13
9	Literatur und Links	13
10	Arbeitsmaterialien	14

1 Design Thinking

Dieses Modul führt in die Innovationsmethode Design Thinking ein. Um ihre Möglichkeiten bei der Entwicklung neuer, kreativer Ideen auszutesten, wenden wir sie auf das Thema „Smart School“ an. Aber auch der thematische Fokus „Smart Home“ ist denkbar.

Die Schülerinnen und Schüler haben die Möglichkeit, neue Produktideen für eine „Smart School“ zu entwickeln, bekannte Produkte zu verbessern und weiterzuentwickeln.

Die Methode des Design Thinking ist im Rahmen verschiedener Module von IT2School einsetzbar.



Lernfeld/Cluster:	IT selber machen
Zielgruppe/Klassenstufe:	X 4. bis 5. Klasse
	X 6. bis 7. Klasse
	X 8. bis 10. Klasse
	X 11. bis 12. Klasse
Geschätzter Zeitaufwand:	5 bis 12 Stunden, auch als Projektwoche umsetzbar
Lernziele:	<ul style="list-style-type: none">• Eine Methode zum kreativen Problemlösen kennenlernen• Probleme, Sachverhalte und Lösungen definieren und beschreiben und reflektieren• Entwicklung von Projektideen
Vorkenntnisse der Schülerinnen und Schüler:	Keine
Vorkenntnisse der/des Lehrenden:	Keine
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters:	Keine
Sonstige Voraussetzungen:	Erforderlich: <ul style="list-style-type: none">• Räume, die an die nötigen Bedürfnisse angepasst werden können• Tische, ggf. Pinnwände• Flipchart• Stifte, Papier, Post-its• Bastelmaterialien

2 Warum gibt es das Modul?

In diesem Modul lernen die Schülerinnen und Schüler die Methode Design Thinking kennen. Viele Unternehmen und Organisationen nutzen Design Thinking, um neue Produkte oder Dienstleistungen zu entwickeln oder zu verbessern – zum Beispiel SAP, Siemens, Volkswagen oder die Deutsche Bahn. Aber auch der Zoo in Hannover hat mithilfe von Design Thinking ein neues Konzept entwickelt, das den Tierpark wieder auf Erfolgskurs brachte. Das Beispiel zeigt, dass Design Thinking in vielen Bereichen eingesetzt werden kann.

Im Mittelpunkt dieser Methode steht der Anwender mit seinen Bedürfnissen. Kaum ein Unternehmen kann es sich heute leisten, an neuen Produkten und Innovationen zu forschen und diese zu entwickeln, ohne sich ein umfassendes Bild von dem erwarteten Nutzer, insbesondere seinen Wünschen und Bedürfnissen zu machen. Die Methode ist an die Arbeit von Designern angelehnt und in sechs Phasen unterteilt: Verstehen, Beobachten, Synthese, Ideen sammeln, Prototyp erstellen und Testen.

Ziel dieses Moduls ist es, dass Schülerinnen und Schüler Design Thinking selbst als kreativen Entwicklungsprozess erleben und so erfahren, wie in der Wirtschaft, vor allem auch in der IT-Branche, mit neuen innovativen Methoden gearbeitet wird. Neben Handlungs- und Problemlösekompetenzen werden in diesem Modul auch Sozialkompetenzen geschult. Durch die gemeinsame Entwicklungsarbeit und das konstruktive und produktive Zusammenarbeiten wird die Teamfähigkeit gestärkt. Kommunikative Kompetenzen werden durch das Führen von Interviews, den Austausch im Team sowie das Präsentieren der Ergebnisse gefördert. Im Rahmen des Entwicklungsprozesses müssen sich die Schülerinnen und Schüler in andere Personen hineinversetzen, sich auf neue Erfahrungen einlassen und andere Sichtweisen zulassen und einnehmen können, wodurch auch die Fähigkeit zur Empathie gestärkt wird.

3 Ziele des Moduls

- eine Methode zum kreativen Problemlösen kennen lernen
- Entwicklung von Projektideen
- erfahren, wie in der Wirtschaft neue, innovative Produkte und Dienstleistungen entwickelt werden
- Förderung von Handlungs- und Problemlösekompetenzen
- Förderung von Sozialkompetenzen: Teamfähigkeit, Kommunikative Kompetenzen, Fähigkeit zur Empathie und Kreativität

4 Die Rolle der Unternehmensvertreterin/des Unternehmensvertreters

Im *Modul B4 – Design Thinking* hat die Unternehmensvertreterin/der Unternehmensvertreter mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

- Sie oder er kann als Special-Guest eingeladen werden, um den Design Thinking-Prozess zu begleiten.
- Sie oder er kann zur Präsentation der Prototypen eingeladen werden.

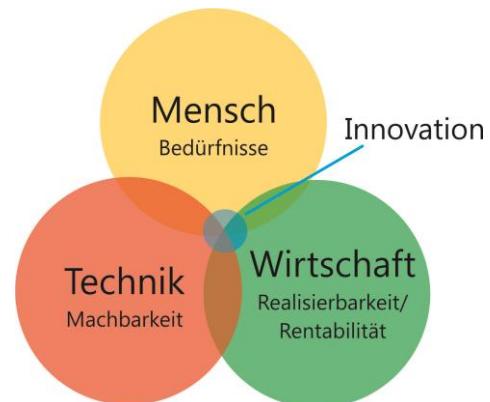
- Sie oder er kann die Schülerinnen und Schüler bei der digitalen Umsetzung der Prototypen unterstützen..

5 Inhalte des Moduls

Design Thinking beschreibt eine Methode zum kreativen Problemlösen und erfinderischen Denken. Insbesondere geht es darum, neue und innovative Produkte und Dienstleistungen zu entwickeln. Was aber genau ist unter Innovation zu verstehen?

Innovation bezeichnet im Design Thinking die Schnittmenge aus dem, was sich Menschen wünschen, was ökonomisch realisierbar und technisch machbar ist. Technische und wirtschaftliche Machbarkeit sind wichtig, bilden aber nicht den Ausgangspunkt.

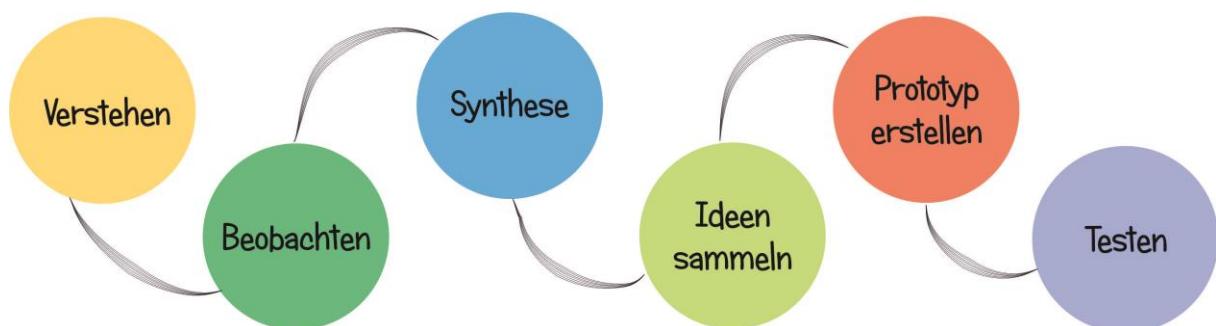
Alle drei Aspekte sind zu beachten, aber im Vordergrund steht der Mensch mit seinen Bedürfnissen. Ein interdisziplinäres Team fragt sich, was ein prototypischer Nutzer braucht, versetzt sich in ihn hinein und versucht, Bedürfnisse zu identifizieren und zu verstehen. Aus diesen Einsichten ergeben sich neue Ideen und Lösungsvorschläge, die in einem Prototyp konkretisiert werden.



Dieser Prozess wird mittlerweile in vielen großen und mittelständischen Unternehmen eingesetzt, um innovative Produkte zu entwickeln. Aber auch in der Schule kann Design Thinking seine Potenziale entfalten.

5.1 Der Design-Thinking-Prozess

Der Design-Thinking-Prozess besteht aus sechs Phasen, die allerdings nicht geradlinig verlaufen. Die folgenden Schritte sind eine idealtypische Darstellung, das Vor- und Zurückspringen, etwa bei neuen Erkenntnissen oder aufgrund nicht zufriedenstellender Ergebnisse, ist manchmal notwendig und auch erwünscht.



1. Verstehen

Zu Beginn wird das Arbeitsfeld definiert und eine geeignete Fragestellung muss gefunden werden. Das Team versucht, ein gemeinsames Verständnis für die Problemlage zu entwickeln. Die Bedürfnisse und Herausforderungen, die in dem Projekt eine Rolle spielen, werden definiert. Es ist darauf zu achten, dass der zu bearbeitende Bereich nicht zu groß oder gar zu klein ist und die Eingangsfrage nicht schon Lösungen impliziert. Geeignete Fragen wären

beispielsweise: Wie kann Schule attraktiver für Schülerinnen und Schüler gestaltet werden? Oder wie soll die Schule von morgen aussehen? Es besteht die Möglichkeit, dass die Klasse eine eigene Frage entwickelt oder aber diese von der Lehrkraft vorgegeben wird. Neben der richtigen Frage spielt auch der Projektplan eine wichtige Rolle. Es sollte zu Beginn festgelegt werden, wie viel Zeit für die einzelnen Phasen zur Verfügung steht. Die Design-Thinking-Phasen können in wenigen Unterrichtsstunden oder im Rahmen einer ganzen Projektwoche durchlaufen werden.

2. Beobachten

Zur Erkenntnisgewinnung stehen verschiedene Möglichkeiten offen: Recherche in Literatur und Internet, Interviews sowie Beobachtung (von Menschen und Alltagssituationen). Auch eigene Erfahrungen können einfließen, daher ist auch das Ausprobieren ein wichtiger Bestandteil der Erkenntnisgewinnung.

3. Synthese

Die gewonnenen Erkenntnisse werden nun zusammengeführt. Dazu werden die Beobachtungen in Form von Geschichten geteilt und visualisiert. Verschiedene Sichtweisen können mit Hilfe von Personas (fiktive, prototypische Nutzerinnen und Nutzer) definiert und Muster identifiziert werden.

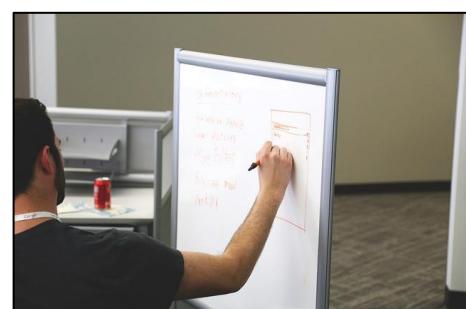
4. Ideen sammeln

Jetzt gilt: Alle Teilnehmerinnen und Teilnehmer können ihren Ideen freien Lauf lassen. Es gibt keine richtige oder falsche Lösung – freies Assoziieren und „Rumspinnen“ sind erlaubt. Möglichst viele Ideen sollen in kurzer Zeit entwickelt und auf Zetteln oder Post-its visualisiert werden. Anschließend werden die Ideen sortiert und strukturiert. Ähnliches wird zusammengefasst. Aus dieser Ideensammlung werden eine oder mehrere ausgewählt, um daran weiterzuarbeiten.



5. Prototyp erstellen

Die ausgewählte(n) Idee(n) werden in einem Prototypen umgesetzt. Damit ist kein „fertiges“ Produkt gemeint – das Ergebnis kann ein Pappmodell oder ein Rollenspiel sein, eine kunstvolle Collage oder technische Konstruktion.



6. Testen

Um den Prototypen zu testen, wird ein Dialog mit den Nutzerinnen und Nutzer geführt; diese werden zu Beteiligten.

Wichtig für Design Thinking ist ein Umfeld, das genügend Platz bietet und die Kreativität anregt. Daher wird nicht nur mit Stift und Papier gearbeitet, sondern es liegen auch Post-its bereit, Malutensilien, Schere und Kleber, Luftballons, Schnur, Pfeifenreiniger, Lego etc.

5.2 Warum Design Thinking?

Design Thinking bietet für Schülerinnen und Schüler eine motivierende und kreative Lernerfahrung, bei der die Phantasie ebenso angeregt wird wie das analytische und konstruktive Denken. Eine neue Form des projektorientierten Unterrichts wird ermöglicht – die Schülerinnen und Schüler entwickeln eigene Schwerpunkte und Ziele, strukturieren Informationen, bewerten ihre eigenen Fortschritte, reflektieren sie und generieren neues Wissen. Dazu arbeiten sie im Team, erleben sich als selbstwirksam und werden dazu ermutigt, sich zu äußern, ihre Ideen zu erklären, zu interpretieren und zu präsentieren, wodurch auch sprachliche Kompetenzen gefördert werden. Darüber hinaus kann sich jeder mit seinen unterschiedlichen Denkweisen, Interessenenschwerpunkten und Erfahrungen an dem Projekt beteiligen.

Design Thinking im Unterricht ist sehr schüleraktivierend und handlungsorientiert und bietet einen realistischen Lebensweltbezug. Das wirkt sich auf die Motivation der Schülerinnen und Schüler aus. Der Sinn und Zweck des Unterrichts für den Alltag, auch für den eigenen beruflichen Werdegang, werden deutlich.

Seit den 80er Jahren des letzten Jahrhunderts befasst sich Dr. Charles Burnette (University of Philadelphia) mit Design Thinking als Unterrichtsmethode. Er geht davon aus, dass sich damit jedes beliebige Thema bearbeiten lässt und dadurch eine umfassende, projektorientierte Lernerfahrung möglich ist. Nähere Informationen zu Design Thinking als Lehrmethode finden Sie in Kapitel 9 (Literatur und Links).

6 Unterrichtliche Umsetzung

Es gibt vielfältige Möglichkeiten, IT im Klassenzimmer einzusetzen. Viele denken als erstes an Tablets, interaktive Whiteboards oder sogar 3D-Drucker. Was aber an der Schule künftig auch vermehrt eine Rolle spielen könnte, um das Lernen und die Organisation zu verbessern, ist die intelligente Vernetzung verschiedener digitaler Geräte.

In Privathaushalten kennt man das unter dem Begriff „Smart Home“. Schon heute ist es möglich, morgens vom Bett aus die Kaffeemaschine per Smartphone zu starten oder auf dem Weg nach Hause schon einmal die Heizung aufzudrehen. Neben den Einsatzmöglichkeiten, die unseren Komfort betreffen, gibt es auch einige, die der Sicherheit sowie dem Energiesparen dienen.

Einige intelligente Systeme können auf die Schule übertragen werden. Zum Beispiel die Funktion des selbstgesteuerten Lüftens. Das Lüften spielt in der Schule eine wichtige Rolle, denn es hat sich gezeigt, dass durch richtiges Lüften die Leistungsfähigkeit und Aufmerksamkeit von Schülerinnen und Schülern sowie Lehrkräften gesteigert werden können. Nicht ausreichendes Lüften führt zu einer vermehrten Bakterien- und Virenanzahl im Klassenzimmer, wodurch die Fehlquote steigt. Eine Idee für die „Schule von morgen“ könnten daher Sensoren sein, die sowohl die Temperatur als auch den Kohlenstoffdioxidgehalt im Klassenzimmer messen und bei Bedarf automatisch eine Öffnung der Fenster veranlassen.

Im folgenden Unterrichtsplan befassen sich die Schülerinnen und Schülern mit den Möglichkeiten einer „Schule von morgen“ und entwickeln mithilfe von Design Thinking eigene Ideen, wie „smarte Technologien“ in der Schule eingesetzt werden können.

6.1 Allgemeine Tipps

1. Design Thinking wird in interdisziplinären Teams umgesetzt, daher achten Sie auch bei Ihren Schülerinnen und Schülern auf eine untypische Mischung der Gruppen. Jede Gruppe sollte 4 bis 6 Personen umfassen.
2. Geben Sie den Schülerinnen und Schülern (Frei-)Raum. Die einzelnen Gruppen sollten für sich genügend Platz und ausreichend Materialien wie Papier, Post-its, Whiteboards oder Flipcharts haben. Auch Bewegung kann die Kreativität fördern.
3. Lassen Sie Fehler zu – an jeder Stelle im Design-Thinking-Prozess ist es möglich, Schritte zurückzugehen, etwa wenn wichtige Einsichten über Personas zu einem späteren Zeitpunkt entstehen.
4. Geben Sie für alle Phasen feste Zeitfenster vor. Dadurch wird verhindert, dass sich die Gruppenmitglieder in einzelnen Phasen „verzetteln“. Sorgen Sie dafür, dass einzelne Schülerinnen und Schüler innerhalb der Gruppen für das Zeitmanagement zuständig sind und benutzen Sie gegebenenfalls eine Stoppuhr oder eine Eieruhr, um die nächste Phase einzuläuten.
5. Es gibt verschiedene Möglichkeiten, ein umfassendes Verständnis eines Sachverhaltes zu gewinnen: Internet-Recherche, aktiv ausprobieren, Interviews führen, Experten befragen.
6. Auch für die Ideenphasen können verschiedene Methoden angewandt werden: klassisches Brainstorming/Braindrawing, bei dem so viele Ideen wie möglich aufgeschrieben oder gemalt werden oder die 6-3-5-Methode (6 Teilnehmer, je 3 Ideen, 5-mal weitergeben). Hierbei schreibt jeder Teilnehmer drei Ideen auf ein Arbeitsblatt, nach fünf Minuten wird das Arbeitsblatt weitergegeben, die Ideen können nun ergänzt oder weiterentwickelt werden.
7. Ideen sollen nicht kritisiert werden. Während des Prozesses der Ideenfindung ist erst einmal alles erlaubt.
8. Beim Prototyping geht es darum, Ideen begreifbar zu machen. Je nachdem, um was es sich handelt (neues Produkt, Dienstleistung etc.), können die Prototypen aus Papier, Pappe, Knete, Lego oder einem anderen geeigneten Material gebaut werden. Es können aber auch Rollenspiele oder Bildergeschichten entwickelt werden, um die Umsetzung einer Dienstleistung darzustellen.

6.2 Grober Unterrichtsplan

Unterrichtsszenarien	Kurze Zusammenfassung
Einführung	Die Schülerinnen und Schüler befassen sich mit vernetzten Geräten (Smart Home). Den Schülerinnen und Schülern wird der Prozess des Design Thinking erklärt.
Verstehen	Es wird ein Standpunkt definiert und eine konkrete Frage festgelegt, z. B.: Was brauchen Lehrkräfte oder Schülerinnen und

	Schüler in der Schule von morgen? Oder bezogen auf das Smart Home: Was brauchen Kinder, Eltern und Großeltern in einem intelligent vernetzten Haus? Aufteilung in Gruppen.
Beobachten	Die Schülerinnen und Schüler schreiben eigene Erfahrungen und Erlebnisse auf, sie führen Interviews mit Mitschülerinnen und Mitschülern sowie mit Lehrkräften oder auch Mitarbeitern der Haustechnik zu möglichen vorhandenen Problemen.
Synthese	Die Ergebnisse werden gesammelt und in kleine Geschichten verpackt. Auch Personas können dazu entwickelt werden. Die Geschichten werden sortiert und analysiert.
Ideen sammeln	Ideen werden passend zu den Personas entwickelt. Mit Klebepunkten wird über die Ideen abgestimmt, um herauszufinden, an welcher weitergearbeitet werden soll.
Prototyp erstellen	Mit Bastelmanmaterial wird ganz analog ein Prototyp entwickelt.
Testen, Ideenaustausch	Interner Austausch der Ideen zweier Gruppen.
Überarbeitung	Überarbeitung, das Feedback kann eingearbeitet werden.
Präsentation im Plenum	Die Schülerinnen und Schüler präsentieren ihren Prototypen im Plenum.
Prototyp mit digitalen Hilfsmitteln	Umsetzung der Idee auch prototypisch mit digitalen Mitteln.

6.3 Stundenverlaufsskizzen

Abkürzungen/Legende

AB = Arbeitsblatt/Arbeitsblätter; L = Lehrkraft; MuM = Mitschülerinnen und Mitschüler; SuS = Schülerinnen und Schüler;
UV = Unternehmensvertreterin/Unternehmensvertreter

6.3.1 Smarte Technologien – Einführung

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
10-15 Min.	Einstieg	Lehrervortrag, Plenum	Begrüßung SuS, ggf. Vorstellung UV, Erläuterungen zur Stunde. Es wird der Frage nachgegangen, was smarte Technologien sind und was sich dahinter verbirgt – Ideen und Beispiele werden im Plenum als Mindmap gesammelt. <ul style="list-style-type: none">• Welche Informations- und Kommunikationstechnologien kennst du im Wohnbereich?• Welche Smart-Home-Anwendungen nutzt deine Familie?• Welche Smart-Home-Anwendungen wünschst du dir zu Hause?	Tafel, Kreide oder Whiteboard, Smartboard
40 Min.	Hinführung	Gruppenarbeit	SuS recherchieren im Internet zu smarten Technologien, danach erstellen sie ein Wandplakat mit ihrem Traumhaus oder ihrer Traumwohnung. Ziel ist es, intelligente Systeme kennenzulernen und diese im nächsten Schritt auf die Schule zu übertragen.	B4.1, Internet, PC, Flipchart-Papier, Stifte
20 Min.	Präsentation	Plenum	Präsentation der Wandplakate	

6.3.2 Smart School – Design Thinking

Der folgende Abschnitt kann in mehreren Unterrichtsstunden oder auch als Projekttag/-woche durchgeführt werden.

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
------	-------	-----------------------------	-----------------------------	----------



15 Min.	Hinführung	Plenum	Vorstellung der Design-Thinking-Methode	Film B4.2, AB B4.3, B4.4
30–60 Min.	Vertiefung	Plenum	<p>Verstehensphase: SuS entwickeln eine geeignete Frage, ggf. kann auch eine Frage vorgegeben werden, z. B.:</p> <ul style="list-style-type: none"> • Wie können smarte Technologien Lehrkräfte und SuS beim Lernen unterstützen? • Wie können smarte Technologien die Kommunikation zwischen Schülern, Lehrkräften und Eltern verbessern? • Wie können intelligente Systeme dazu genutzt werden, um die Schule umweltfreundlicher zu machen? <p>Standpunkte werden definiert und Gruppen aufgeteilt; Gruppeneinteilung der Klasse: 6 Gruppen à etwa 5 Pers.</p>	AB B4.5
60–180 Min.	Vertiefung	Gruppenarbeit	Beobachten: SuS recherchieren zum Thema im Internet und überlegen, welche Erfahrungen und Beobachtungen sie selbst auf diesem Feld gemacht haben. Darüber hinaus führen sie Interviews mit MuM, LuL oder Mitarbeiterinnen und Mitarbeitern der Haustechnik.	
30–60 Min.	Vertiefung	Gruppenarbeit	Synthese: Die Gruppe trägt ihre Ergebnisse zusammen, prototypische Nutzerinnen und Nutzer (Personas) werden definiert und die Ergebnisse und Eindrücke sortiert und analysiert: Gibt es etwas, das häufig genannt wurde? ...	B4.6
30–60 Min.	Praxisphase	Gruppenarbeit	<p>Ideen sammeln:</p> <ol style="list-style-type: none"> 1. Schritt: Ideenfindung – jeder in der Gruppe skizziert so viele Ideen wie möglich (10 min.). 2. Schritt: Vorstellung der Ideen in der Gruppe, Ideen sortieren (20-45 Min., je nach Anzahl). 3. Schritt: Abstimmen mit Klebepunkten, an der besten Idee wird weitergearbeitet (5 Min.). 	Post-its, Papier, Klebeband, Stifte
20–30	Praxisphase	Gruppenarbeit	Prototyp bauen: Mit Bastelmaterialien wird ein erster Prototyp gebaut, um	Papier, Schere,

Min.			die Idee <i>begreifbar</i> zu machen.	Pappe, Lego, Knete, Pfeifenreiniger, Klebeband, Stifte etc.
20–30 Min.	Testen	Gruppenarbeit	Testen: Zwei Gruppen stellen sich gegenseitig ihren Prototypen vor, Feedback wird gegeben.	
10 Min.	Überarbeiten	Gruppenarbeit	Das Feedback und Änderungsvorschläge werden eingearbeitet, der Prototyp wird noch einmal überarbeitet.	
30 Min.	Präsentation	Plenum	Alle Gruppenergebnisse werden im Plenum präsentiert.	

6.3.3 Unterrichtseinheit – Bau eines digitalen Prototyps

Zeit	Phase	Sozialform/ Lehrerimpuls	Inhalt/Unterrichtsgeschehen	Material
10 Min.	Einstieg	Plenum	Begrüßung SuS, Erläuterungen zur Stunde, Anknüpfung an die Ergebnisse der Design-Thinking-Einheit, Aufteilung in Gruppen.	
60–120 Min.	Praxisphase	Gruppenarbeit	Der analoge Prototyp kann mit digitalen Mitteln umgesetzt werden, etwa mit MocoMoco, Scratch, ifttt ¹ oder auch, falls vorhanden, Littlebits. Zusätzliche Informationen erhalten Sie in unseren Modulen „B5 – Leichter Programmereinstieg“ und „B6 – Mein besonderer Anschluss“.	PCs, geeignete Anwendung
20–30 Min.	Sicherung	Plenum	Präsentation der Ergebnisse	

¹ Abkürzung „If this then that“ – dieser Dienst erlaubt es Benutzern, verschiedene Webanwendungen miteinander zu verknüpfen, z. B. Facebook, Twitter, Dropbox, GMail, Evernote.

7 Einbettung in verschiedene Fächer und Themen

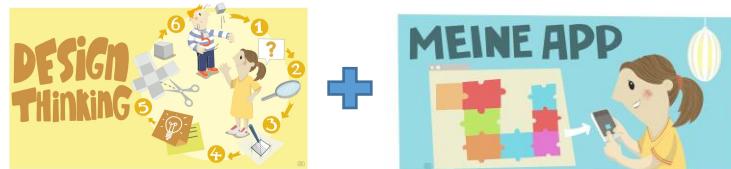
Da es sich beim Design Thinking in erster Linie um eine Methode handelt, lässt sich diese in jedes Unterrichtsfach einbinden.

8 Anschlussthemen

Das Modul M1 Design Thinking ist im Rahmen verschiedener Module einsetzbar. Beispielsweise können neue innovative Produkte mit Scratch (B5) und dem MocoMoco (B6) umgesetzt werden.



Auch für die Entwicklung einer neuen App kann Design Thinking verwendet werden.



9 Literatur und Links

- Burnette, Charles (2005): **IDESIGN – Design Thinking als Unterrichtsmethode**. Ein Modell für Pädagoginnen und Pädagogen mit Anleitungen, Arbeitsblättern und vielem mehr. Online: <http://www.idesignthinking.com>
- **Toolkit für Pädagoginnen und Pädagogen**, mit zusätzlichen Informationen, zahlreichen Videos etc. Online: <http://www.designthinkingforeducators.com>
- Interview mit Michael Busch zum Thema **Design Thinking für die Schule der Zukunft**. Online: <https://blog.lernox.de/2014/12/design-thinking-in-der-schule-der-zukunft/>
- Beschreibung eines **Design-Thinking-Projekts** an einer Schule. Online: <http://www.futureatschool.wordpress.com/2011/10/18/john-dewey-und-design-thinking-ein-experiment/>

10 Arbeitsmaterialien

Nr.	Titel	Beschreibung
😊 B4.1	Smarte Wohnwelt	Arbeitsblatt zur Einführung in die Thematik. SuS gestalten Wandplakat zum „Smart Home“.
😊 B4.2	Design Thinking	Erklärvideo zur Methode Design Thinking.
😊 B4.3	Smart School – Fahrplan	Dokument mit einer Übersicht der einzelnen Phasen von Design Thinking für Lehrkräfte.
😊 B4.4	Smart School – Fahrplan	Dokument mit einer Übersicht der einzelnen Phasen von Design Thinking für Schülerinnen und Schüler.
😊 B4.5	Die Schule von morgen: Die richtige Frage finden	Arbeitsblatt zur Unterstützung bei der Ermittlung der „richtigen“ Frage.
😊 B4.6	Personas	Arbeitsblatt zur Entwicklung von Personas.

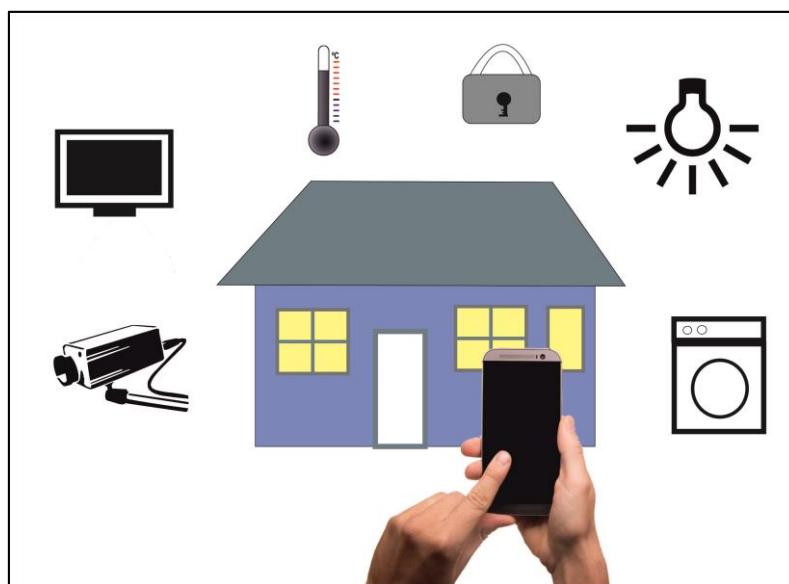
Legende

- 😊 Material für Schülerinnen und Schüler
- 😊 Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
- 😊 Zusatzmaterial

Smarte Wohnwelt

Morgens noch länger im Bett liegen bleiben, aber mit dem Smartphone schon mal die Kaffeemaschine starten? Auf dem Weg nach Hause die Heizung „aufdrehen“, damit es schön warm ist, wenn man ankommt? Das alles ist heute schon möglich, genauso wie der Fernseher, der automatisch startet, sobald man den Raum betritt, oder das Flurlicht, das angeschaltet wird, wenn man die Haustür aufschließt. Unter dem Begriff „Smart Home“ wird die Vernetzung von Haushaltsgeräten, Haustechnik und Unterhaltungselektronik bezeichnet. Smart-Home-Technologie unterteilt sich in drei Bereiche:

- **Energie** – alles, was zum Energiesparen beiträgt, Licht- und Heizungssteuerung
- **Sicherheit** – Brand- und Einbruchschutz
- **Komfort** – Geräte, die automatisch an, oder abgeschaltet werden, Bedienung von Geräten etc.



Euer Traumhaus

Wie sieht euer „Smart Home“ der Zukunft aus? Recherchiert im Internet und informiert euch, welche Möglichkeiten es gibt und wie intelligente Systeme in Häusern eingesetzt werden können.

Erstellt ein Wandplakat oder eine Präsentation mit den digitalen Möglichkeiten, die euer Traumhaus in Zukunft bieten sollte. Welche Schwerpunkte sind euch wichtig? Und warum wünscht ihr euch das?

Präsentiert am Ende der Stunde euer Traumhaus.

Smart School – Fahrplan

Im Folgenden finden Sie einen möglichen Verlauf zur Fragestellung: „Wie können intelligente Systeme dazu genutzt werden, um die Schule umweltfreundlicher zu gestalten?“ In den kurzen Abschnitten erklären wir exemplarisch, wie die einzelnen Phasen von Design Thinking ablaufen und wie viel Zeit ungefähr dafür eingeplant werden sollte.

Design Thinking kann in lediglich 5 bis 6 Unterrichtsstunden, aber auch in Form einer Projektwoche umgesetzt werden. Überlegen Sie sich als Lehrkraft, wie viel Zeit Sie für einen Durchgang aufwenden wollen. Schaffen Sie auch für Ihre Schülerinnen und Schüler Transparenz und geben Sie Hinweise dazu, wie viel Zeit in den einzelnen Phasen zur Verfügung steht. Um die Zeit nicht aus den Augen zu verlieren, bietet es sich an, mit einer Stoppuhr oder Eieruhr die nächste Runde einzuläuten.

Schritt 1: Aufgabe ist es, eine konkrete Frage zu dem Themenbereich „Smart School – Die Schule von Morgen“ zu entwickeln. Häufig kommen erste Ideen für Fragen durch die Formulierung eigener Probleme oder Wünsche. Die Schülerinnen und Schüler können in Kleingruppen AB B4.5 bearbeiten und so zu verschiedenen Fragestellungen gelangen. Jede Gruppe entscheidet sich dann für eine Frage, der sie weiter nachgehen möchte. Sie können als Lehrkraft aber auch eine konkrete Frage vorgeben. Der weitere Verlauf des Projekts sollte geplant werden. Gehen Sie mit ihren Schülerinnen und Schüler gemeinsam insbesondere die Zeiten für die einzelnen Phasen durch.



Beispiel: Wie können intelligente Systeme dazu genutzt werden, um die Schule umweltfreundlicher zu gestalten?

Zeit: 30 bis 60 Min.

Schritt 2: Die thematische Auseinandersetzung startet. Die Schülerinnen und Schüler haben verschiedene Möglichkeiten, sich der Thematik zu nähern und mögliche Wünsche, Bedürfnisse und Probleme der Nutzer zu ermitteln. Die Gruppe kann beispielsweise im Internet zum Thema „Energiesparen in der Schule“ recherchieren oder selbst einen Rundgang durch die Schule machen, um Problembereiche zu entdecken und ggf. mit Fotos zu dokumentieren. Außerdem können Interviews mit Mitschülerinnen und Mitschülern, Lehrkräften oder auch Mitarbeitern der Haustechnik geführt werden. Im Rahmen von Interviews erfahren die Schülerinnen und Schüler vielleicht, wo an ihrer Schule besonders viel Energie verschwendet wird oder welche Bereiche besonders problembehaftet sind. Möglicherweise zeigt sich hier auch, dass Kinder und Erwachsene unterschiedliche Dinge als störend empfinden.



Beispiel: Wenn gelüftet wird, steht die Heizung oft noch auf 5; in der Toilette brennt immer das Licht, obwohl keiner drin ist, Müll liegt häufig neben den Müllheimern...

Zeit: 1 bis 3 Stunden

Schritt 3: Nun werden die gewonnenen Erkenntnisse zusammengeführt und ausgewertet. Hierzu bietet es sich an, die Beispiele aus eigenen Erfahrungen oder Interviews zu visualisieren. Auf diese Weise können zum Beispiel Muster schneller abgeleitet und sogenannte Personas entwickelt werden. Dies sind prototypische Nutzer, die das Problem veranschaulichen sollen.

Synthese

Beispiel: Ältere Schülerinnen und Schüler und Lehrkräfte haben beobachtet, dass besonders jüngere Schülerinnen und Schüler das Licht in der Toilette vergessen auszuschalten. Daraus kann sich folgende Persona ergeben: Paul, 11 Jahre, ist noch neu an der Schule, hat aber schon viele Freunde gefunden. In der Pause spielt er sehr gerne mit ihnen Fußball, da möchte er keine Sekunde verpassen. Oder: Jüngere Schülerinnen und Schüler haben häufiger beobachtet, dass die älteren, cooleren Schülerinnen und Schüler ihren Müll nicht in den Papierkorb werfen...

Zeit: 30 Min. bis 1 Stunde

Schritt 4: Nun werden kreative Ideen und Lösungen in Bezug auf die Personas gesucht. Es werden alle Ideen und Lösungen aufgeschrieben und aufgemalt, um die Schule umweltfreundlicher zu gestalten. Auch verrückte Ideen sind ausdrücklich erlaubt. Danach besprechen die Schülerinnen und Schüler ihre Ideen und stimmen mit Klebepunkten darüber ab, welche Idee weiter verfolgt wird.

Ideen
sammeln

Beispiel: Das Licht in der Schule wird ausschließlich von Bewegungssensoren gesteuert oder der Lichtschalter in der Toilette ist gleichzeitig der Türöffner. Öffnet man die Fenster, werden die Heizungen automatisch runtergeregelt. Über den Mülleimern werden Basketballkörbe befestigt, die applaudieren, wenn man etwas hinein wirft, so können Schülerinnen und Schüler motiviert werden, ihren Müll direkt in den Mülleimer und nicht daneben zu werfen.

Zeit: 40 bis 60 Min.

Schritt 5: Ein erster Prototyp wird gebastelt. Er dient dazu, die Lösungsidee in der Praxis zu überprüfen. Der Prototyp kann mit verschiedenen Materialien (Papier, Knete, Luftballons, Styropor etc.), als Rollenspiel oder auch als Wandplakat umgesetzt werden.

Prototyp
erstellen

Beispiel: Die Idee mit dem Basketballkorb wurde mit Klebepunkten ausgewählt. Dieser Korb wird dann mit Hilfe von Draht an einem Papierkorb umgesetzt.

Zeit: 30 Min. bis 1,5 Stunden

Schritt 6: Im letzten Schritt wird der Prototypen einer anderen Gruppe präsentiert. Diese Gruppe hat nun die Möglichkeit Feedback zu geben. Das Feedback zeigt, ob die Idee angenommen wird, weiterentwickelt oder verworfen werden muss. Wird die Idee verworfen, setzt der Ablauf wieder am Punkt „Ideen sammeln“ ein, die Schülerinnen und Schüler schauen sich ihre Ideen an und wählen eine neue aus.

Testen

Beispiel: Die Gruppe präsentiert ihre Idee der Nachbargruppe und umgekehrt.
Es wird demonstriert, wie der Mülleimer Beifall gibt, wenn etwas in ihn hinein geworfen wird.

Zeit: 30 bis 60 Min.

Smart School – Fahrplan



Verstehen

Schritt 1: Überlegt euch eine geeignete Frage zum Thema „Smart School – die Schule von morgen“. Häufig kommen erste Ideen für Fragen durch die Formulierung eigener Probleme oder Wünsche. Das Arbeitsblatt B4.5 wird euch dabei helfen. Legt im Anschluss im Plenum fest, wie viel Zeit für die einzelnen Phasen aufgewendet werden sollen und schreibt das in die entsprechenden Felder.

Eure Frage:



Beobachten

Zeit: _____

Schritt 2: Jetzt geht es darum, die gewählte Problemstellung besser zu verstehen. Ihr könnt zum Thema im Internet recherchieren, selbst einen Rundgang durch die Schule machen oder aber Interviews führen. Überlegt euch, wer die richtigen Ansprechpersonen für eure Frage sind: Das können Mitschülerinnen und Mitschüler sein, Lehrkräfte oder aber vielleicht auch jemand aus der Verwaltung oder der Hausmeister.

Zeit: _____



Synthese

Schritt 3: Führt die gewonnenen Erkenntnisse zusammen und wertet sie aus. Hierzu bietet es sich an, die Beispiele aus eigenen Erfahrungen oder den Interviews zu visualisieren. Malt die Ergebnisse als kleine Geschichten/Bilder auf und sortiert sie. Vielleicht wurden manche Aspekte besonders häufig in Interviews genannt oder andere Gemeinsamkeiten fallen euch auf. Hierdurch können zum Beispiel Muster schneller abgeleitet und sogenannte Personas entwickelt werden. Personas sind erfundene Personen, die das Problem veranschaulichen sollen. Hier hilft euch das Arbeitsblatt B4.6 weiter.

Zeit: _____



Ideen sammeln

Schritt 4: Nun werden kreative Ideen und Lösungen gesucht. Schreibt und malt alle Ideen und Lösungen auf, die euch einfallen – auch verrückte Sachen sind erlaubt. Innerhalb von 10 Minuten solltet ihr so viele Ideen wie möglich skizziert haben. Danach besprecht ihr die Ergebnisse und stimmt mit Klebepunkten darüber ab, welche Idee ihr weiter verfolgen wollt.

Zeit: _____

Prototyp
erstellen

Schritt 5: Für die ausgewählte Idee bastelt ihr einen Prototyp. Das ist ein vereinfachtes Modell und dient dazu, die Lösungsidee in der Praxis zu überprüfen. Der Prototyp kann mit verschiedensten Materialien (Papier, Knete, Luftballons, Styropor etc.), als Rollenspiel oder auch als Wandplakat umgesetzt werden.

Zeit: _____

Testen

Schritt 6: Ihr präsentiert euren Prototypen einer anderen Gruppe und holt euch Feedback ein. Die Rückmeldungen der anderen Gruppe zeigen euch, ob eure Idee angenommen wird, weiterentwickelt oder sogar verworfen werden muss. Wird die Idee verworfen, setzt ihr wieder am Punkt „Ideen sammeln“ ein, schaut eure Ideen an und wählt eine neue aus.

Zeit: _____

Die Schule von morgen: Die richtige Frage finden

Um für das Design-Thinking-Projekt die richtige Frage zu finden, ist es hilfreich, zunächst die eigenen Wünsche oder Ärgernisse zu formulieren. Überlegt in eurer Gruppe, was ihr euch wünscht und was euch ärgert.

Nachdem ihr eure Vorstellungen aufgeschrieben habt, formuliert daraus eine Frage.



Zum Beispiel

- Es ärgert mich, dass ich manchmal zur ersten Stunde in die Schule komme, die erste Stunde dann aber ausfällt.
 - Wie kann die Kommunikation zwischen Schülern, Lehrern und auch Eltern verbessert werden?
- Ich wünsche mir, dass die Schule mehr auf die Umwelt achtet und Energie spart.
 - Wie können intelligente Systeme dazu genutzt werden, um die Schule umweltfreundlicher zu gestalten?

Ich wünsche mir ...

Es ärgert mich, dass ...

Fragen?

Persona – Prototypischer Nutzer

Name: _____

Alter: _____

Geschlecht: _____



Foto – zum selber malen

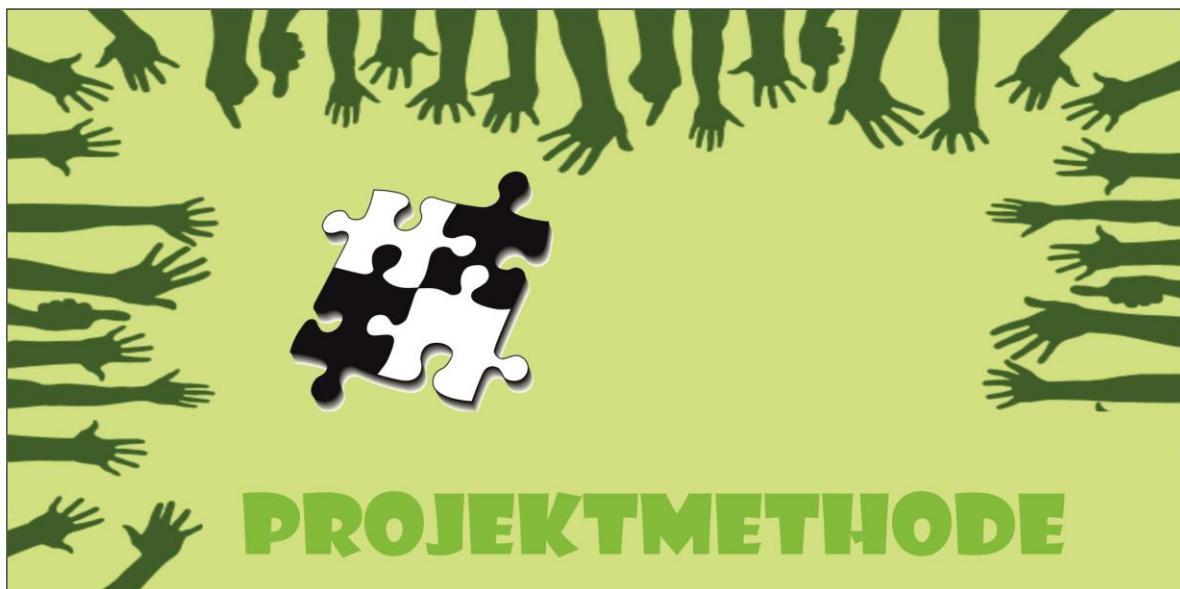
Wer ist die Person? Was macht die Person? Ist es eine Lehrkraft, eine Schülerin oder ein Schüler oder vielleicht ein Mitarbeiter?

Welche Eigenschaften hat die Person? Was macht die Person aus? Welche Wünsche hat die Person? Ärgert sie etwas? Hat sie Probleme?

Wo hält sich die Person auf, etwa während des Unterrichts, in den Pausen?

IT2School

Gemeinsam IT entdecken



Modul M2 – Projektorientierter Unterricht

Eine Entwicklung von



In Kooperation mit



Im Auftrag der



Inhalt

1	Projektorientierter Unterricht.....	3
2	Warum gibt es das Modul?	4
3	Ziele des Moduls.....	4
4	Die Rolle des Unternehmensvertreterin/des Unternehmensvertreters.....	4
5	Inhalte des Moduls.....	4
5.1	Merkmale von Projekten	5
5.2	Projektphasen	6
5.3	Allgemeine Tipps.....	8
6	Unterrichtliche Umsetzung.....	8
7	Einbettung in verschiedene Fächer und Themen	9
8	Literatur und Links	9
9	Arbeitsmaterialien	9



1 Projektorientierter Unterricht

Dieses Modul führt in den projektorientierten Unterricht ein. Dies ist eine Form des handlungsorientierten Unterrichts bei dem Schülerinnen und Schüler eine Aufgabe oder eine Problemstellung selbstständig – von der Planung über die Durchführung bis zur Präsentation – innerhalb einer Gruppe bearbeiten.



In einigen Modulen von IT2School werden offene Aufgabenstellungen unterbreitet, die in Form eines Projektes umgesetzt werden können. Beispielsweise haben die Schülerinnen und Schüler die Möglichkeit eigene Spiele oder Apps zu programmieren. Dieses Methodenmodul gibt ganz konkrete Hinweise zur Umsetzung von Projekten.

Lernfeld/Cluster:	IT selber machen		
Zielgruppe/Klassenstufe:	X	4. bis 5. Klasse	
	X	6. bis 7. Klasse	
	X	8. bis 10. Klasse	
	X	11. bis 12. Klasse	
Geschätzter Zeitaufwand:	5 bis 12 Stunden, auch als Projektwoche umsetzbar		
Lernziele:	<ul style="list-style-type: none">Entwicklung von ProjektideenFörderung von Handlungs- und ProblemlösekompetenzenFörderung von Schlüsselkompetenzen: Teamfähigkeit, Kommunikative Kompetenzen, Fähigkeit zur Empathie und Kreativität		
Vorkenntnisse der Schülerinnen und Schüler:	Keine		
Vorkenntnisse der/des Lehrenden:	Keine		
Vorkenntnisse der Unternehmensvertreterin/des Unternehmensvertreters:	Keine		
Sonstige Voraussetzungen:	keine		

2 Warum gibt es das Modul?

Guter Unterricht lebt von Methodenvielfalt, wenn Sozialformen variieren und unterschiedliche Grundformen des Unterrichts (Frontalunterricht, Projektarbeit, Planarbeit, Freiarbeit etc.) umgesetzt werden. Auch die Module von IT2School sind durch Methodenvielfalt geprägt. Einige Themen im Rahmen der Module sind stark geführt, mit konkreten Arbeitsblättern und andere Module bieten Freiraum, um die Inhalte handlungsorientiert in Projekten zu behandeln und umzusetzen.

Der projektorientierte Unterricht ist eine Möglichkeit, um einen offenen Lernraum zu eröffnen und gleichzeitig Lernen als Gemeinschaftserlebnis zu strukturieren. Lehrkräfte werden zu Lernbegleitern, aber auch Projektleitern, denn Projekte bewegen sich in einem bestimmten Rahmen, den es zu strukturieren gilt.

Gerade der Umgang mit digitalen Medien, IT und Informatik ermöglicht eine kreative und produktive Auseinandersetzung im Rahmen von Projekten. Dieses Modul gibt Hilfestellung, wie Schülerinnen und Schüler sich selbstgesteuert mit digitalen Inhalten auseinandersetzen können. Die Theorie wird mit der Praxis verknüpft und Informatik und IT wird begreifbar.

3 Ziele des Moduls

- Entwicklung von eigenen Projektideen
- Förderung von Handlungs- und Problemlösekompetenzen
- Förderung von Schlüsselkompetenzen: Teamfähigkeit, Kommunikative Kompetenzen, Fähigkeit zur Empathie und Kreativität
- Förderung von Planungs- und Handlungskompetenz

4 Die Rolle des Unternehmensvertreterin/des Unternehmensvertreters

Im *Modul M2 – Projektorientierter Unterricht* hat die Unternehmensvertreterin/ der Unternehmensvertreter mehrere Möglichkeiten aktiv mitzuwirken. Hier einige Anregungen:

- Sie oder er kann als Special-Guest eingeladen werden, um Projekte zu begleiten.
- Sie oder er kann zur Präsentation von Ergebnissen eingeladen werden.
- Sie oder er kann die Schülerinnen und Schüler bei der digitalen Umsetzung der Projekte unterstützen.

In den einzelnen Modulen werden weitere Hinweise gegeben, wie man sich als Unternehmensvertreterin/ Vertreter aktiv im jeweiligen Modul beteiligen kann.

5 Inhalte des Moduls

Die Projektmethode hat in der außerschulischen Kinder- und Jugendarbeit sowie der Reformpädagogischen Bildung schon lange Tradition und auch in vielen Unternehmen und Branchen wird Projektarbeit praktiziert. Schon im 18. Jahrhundert haben angehende

Architekten an der Academie Royale d'Architecture Projekte durchgeführt und selbstständig und kooperativ Gebäude im Rahmen ihrer Ausbildung entworfen. Aber eine erste systematische Konzeption der Projektmethode für den Unterricht entwickelten die Pädagogen Dewey und Kilpatrick. Damit legten sie den Grundstein für eine partizipative, kooperative, demokratische und handlungsorientierte Pädagogik.

Geprägt von der Philosophie des Pragmatismus gingen Dewey und Kilpatrick davon aus, dass ein Gramm Erfahrung besser ist, als eine Tonne Theorie (Dewey in Demokratie und Erziehung, 1916). Die praktischen Erfahrungen stehen im Mittelpunkt, um daraus Theorien zu entwickeln oder Theorien zu überprüfen, das reine theoretische Wissen ohne praktische Erfahrung kann nie vollkommen erfasst werden. Verkürzt wird daher die Theorie von Dewey auch häufig als „Learning by doing“ bezeichnet.

Eingebettet ist ihre Idee von Projekten in ein grundlegendes Verständnis von Demokratie (Gudjons, 2014). Dewey versteht Demokratie nicht nur als Herrschaftsform, sondern als eine Form des Zusammenlebens. Bildungsprozesse sollten daher in Form eines demokratischen Miteinanders gestaltet sein mit dem Ziel, Kindern und Jugendlichen Erfahrungen zu ermöglichen, in denen sie sich mit der sächlichen als auch gesellschaftlichen Umwelt auseinander setzen können und durch Kommunikation und Kooperation das demokratischen Zusammenleben einüben. Für den Unterricht bedeutet dies, dass die Schülerinnen und Schüler frei und selbstbestimmt und ohne Hierarchie ihren Lernweg selbst in die Hand nehmen.

Im Rahmen von Projekten wird eine Aufgaben über einen längeren Zeitraum selbstständig innerhalb einer Gruppe bearbeitet. Das Thema entspricht dem Interesse der Schülerinnen und Schüler oder wird selbst von ihnen ausgewählt und auch der Projektverlauf und Arbeitsplan wird selbstständig von ihnen konzipiert. Die Schülerinnen und Schüler planen ihre Inhalte und Ziele und setzen diese demokratisch in der Gruppe um. Am Ende eines Projekts steht ein verwertbares Produkt, dass einer Öffentlichkeit zugänglich gemacht wird. Die Präsentation ist wesentlicher Bestandteil von Projekten.

Häufig ist dies in der Schule, bedingt durch Lehrpläne und 45 min. Rhythmus nur bedingt umsetzbar, daher wird im Rahmen von Schule von *projektorientiertem Unterricht* gesprochen. An einigen Stellen findet eine Reduktion statt, so dass es einem freien Projekt nach der Projektmethode noch nahe ist, aber dem auch nicht mehr ganz entspricht.

Die Themen werden beispielsweise von der Lehrkraft festegelegt. Dabei sollte aber darauf geachtet werden, dass sie im Interessensbereich der Schülerinnen und Schülern liegen und sie auch einen lebensweltlichen Bezug haben. Klassen, die es nicht gewohnt sind, in Projekten zu arbeiten, können sich überfordert fühlen, daher sollte das Lernen in Projekten geübt werden. Je häufiger Schülerinnen und Schüler in Projekten lernen, desto mehr Elemente der Projektmethode kann man implementieren. Die Schülerinnen und Schüler können so zunehmend selbstständiger und eigenverantwortlicher arbeiten.

5.1 Merkmale von Projekten

Projektorientiertes Lernen und Arbeiten zeichnet sich durch bestimmte Merkmale aus.

- **Projekte orientieren sich an den Interessen der Beteiligten –** Die Interessen von Schülerinnen und Schülern werden beachtet und mit einbezogen. Es kann aber auch Aufgabe der Lehrkraft sein, Interesse erst einmal zu wecken, durch erstes praktisches Ausprobieren, Videos etc.

- **Gesellschaftliche Relevanz** - Die Aufgaben- und Problemstellung ist Lebensweltorientiert
- **Produktorientierung** - Es wird an einem Produkt gearbeitet, über das am Ende kommuniziert werden kann, daher wird das Produkt auch öffentlich präsentiert.
- **Kooperation und Reflexion** - Es wird Raum für offene Kommunikation und Kooperation gegeben und Reflexionsphasen mit eingeplant.
- **Selbstorganisation** - Die Schülerinnen und Schüler sind im Rahmen der Möglichkeiten weitestgehend aktiv am Planungsprozess beteiligt. Die Lehrkraft wird zum Lernbegleiter.
- **Zielgerichtete Projektplanung** – Das Planvolle Handeln ist wesentlich. Die Abfolge von einzelnen Schritten, Aufgabenverteilung, Zeiteinteilung etc. wird von den Schülerinnen und Schülern, unterstützt durch die Lehrkraft, geplant.
- **Einbezug vieler Sinne** – Während des handlungsorientierten Lernens sollen so viele Sinne, wie möglich mit einbezogen werden. Pestalozzi bezeichnete dies als Lernen mit „Herz, Kopf und Hand“. Kognitive und praktische Tätigkeiten werden miteinander verbunden.

5.2 Projektphasen



1. Einführung/ Initiierung

Zu Beginn wird das Projektvorhaben durch die Lehrkraft oder die Schülerinnen und Schüler initiiert. Die Schülerinnen und Schüler sollten motiviert und neugierig gemacht werden, beispielsweise durch Videos oder praktisches Ausprobieren.

Das Thema wird festgelegt und Gruppen eingeteilt. Die Projektleitung, also die Lehrkraft, legt die Rahmenbedingungen fest (Ziel, Zeit, Präsentation, Arbeitsort, Beurteilung etc.) und macht diese transparent, ggf. können sie im Plenum noch mal diskutiert und angepasst werden. Durch eine klare Absteckung der Rahmenbedingungen wird den Schülerinnen und Schülern hält gegeben und Fehlplanungen kann entgegengewirkt werden.

2. Projektplanung

In den Kleingruppen befassen sich die Schülerinnen und Schüler im nächsten Schritt mit der Projektplanung. Sie sammeln und strukturieren Material sowie Aufgaben, legen einen Zeitplan fest und verankern alles in einem Projektplan.

Die Projektleitung kann hierfür Material zu Verfügung stellen sowie Strukturhilfen und Beratung anbieten.

3. Durchführung

Die Schülerinnen und Schüler führen nun ihr Projekt praktisch aus und erstellen ihr Produkt.

Die Projektleitung hilft bei auftretenden Problemen, unterstützt die Schülerinnen und Schüler und organisiert die gemeinsamen Fixpunkte, zu denen alle wieder im Plenum zusammenkommen.

4. Präsentation

Die Projektergebnisse werden präsentiert. Die Gruppen können sich gegenseitig ihre Produkte vorstellen, es sind aber auch größere Präsentationen denkbar, wie beispielsweise Ausstellungen in der Aula oder beim Tag der offenen Tür. Auch Eltern und UnternehmensvertreterInnen können eingeladen werden.

Ergebnisse können auch im Internet veröffentlicht werden, beispielsweise auf der Schulhomepage oder als Video auf YouTube.

Die Projektleitung organisiert die Präsentation und schafft Raum für wertschätzendes Feedback für die Schülerinnen und Schüler.

5. Reflexion

Am Ende wird das Projekt reflektiert. Dabei sollen das Produkt, der Prozess, die Organisation sowie der Gruppenprozess beachtet werden. Die Projektleitung strukturiert die Reflexion. Es hilft, klare Fragen zu formulieren wie beispielsweise:

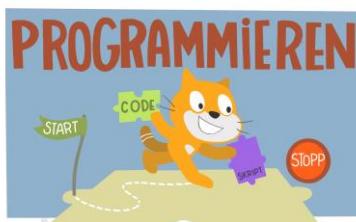
- Was haben wir gemacht? Wie sind wir Vorgegangen?
- Wie sind wir miteinander umgegangen?
- Konnte sich jeder einbringen?
- Was haben wir erreicht?
- Etc.

5.3 Allgemeine Tipps

1. Während des Prozesses sollten Fixpunkte eingeplant werden, Zeiten zu denen die Gruppen immer wieder zusammenkommen, um über den aktuellen Stand der Arbeit zu berichten und ggf. um sich über Probleme auszutauschen.
2. Die Struktur von Projekten sollte Schülerinnen und Schülern zu Beginn klar vermittelt werden, damit sie verinnerlicht werden kann. Es sollte nicht so ausssehen, als ob man einfach Freiarbeit hat. Die Projektmethode folgt einem Plan und man durchläuft verschiedene Phasen, die den Schülerinnen und Schülern transparent gemacht werden sollten.
3. Die Lernumgebung hat wesentlichen Einfluss auf das Verhalten und das Lernen. Es ist eine Sitzordnung zu bevorzugen, die es den Gruppen ermöglicht, im gemeinsamen Austausch an ihrem Produkt zu arbeiten. Für die Reflexion, Fixpunkte oder auch die Präsentation ist ein Sitzkreis angemessen.
4. Die Lehrkraft wird in Projekten maßgeblich zum Lernbegleiter. Sie greifen ein, wenn Schülerinnen und Schüler Hilfe benötigen. Dies sollte aber nicht zu früh geschehen, damit Schülerinnen und Schüler eigene Ideen entwickeln können, um Probleme zu lösen. Für manche Lehrkräfte ist es schwierig Fehlläufe oder Stillstand in Projekten auszuhalten.

6 Unterrichtliche Umsetzung

Innerhalb verschiedenen Module von IT2School kann die Projektmethode angewendet werden. Hier einige Beispiele:



B5 – Programmieren

Im Rahmen des Modules B5 können die Schülerinnen und Schüler selbstständig ein eigenes Projekt umsetzen. Um einen einfachen Projekteinstieg zu ermöglichen wurde ein offener Programmierauftrag konzipiert (AB 5.4). Die Schülerinnen und Schüler können mit Hilfe des Arbeitsblattes ihr eigenes Projekt strukturieren und planen.



B6 – Mein Anschluss

Auch im Modul B6 wurde für Scratch in Zusammenhang mit dem MocoMoco ein Programmierauftrag konzipiert, den die Schülerinnen und Schüler selbstständig im Rahmen eines Projekts umsetzen können.

B7 – App Inventor



Im Modul B7 gibt es keinen konkreten Auftrag. Hier besteht die Möglichkeit, tatsächlich ein freies Projekt bezüglich der Themenwahl, Dauer etc. durchzuführen. Als Lehrkraft geben Sie den Rahmen für das Projekt vor, ggf. können Sie auch eine thematische Vorgabe machen.

7 Einbettung in verschiedene Fächer und Themen

Da es sich beim Projektorientierten Unterricht um eine Methode handelt, lässt sich diese in jedes Unterrichtsfach zu verschiedensten Themen einbinden. Die Ziele und Kompetenzen können daher auch stark variieren. Allen Projekten ist gegeben, dass folgende Sozial- und Selbstkompetenzen bzw. so genannte Schlüsselqualifikationen gefördert werden:

Die Schülerinnen und Schüler können ...

- Aufgaben entsprechend ihrer Stärken und Schwächen übernehmen.
- Verantwortung für ihre Aufgaben übernehmen.
- ihre Sichtweisen und Standpunkte formulieren und vertreten.
- Selbstwirksamkeit erleben, dadurch, dass sie ein umfangreiches Projekt durchführen und abschließen.
- Teamfähigkeit zeigen, indem sie sich auf gemeinsame Ziele und Herangehensweisen festlegen und sich gegenseitig in der Gruppe unterstützen.
- Arbeitsergebnisse reflektieren und kritisch beurteilen.
- neue Inhalte erarbeiten und strukturieren und so Planungskompetenz erwerben.

8 Literatur und Links

- Bruggmann, Guido (2006): Einführung in die Projektmethode. Online: http://www.guteschule.ch/Schriften/Bruggmann_Projektmethode.pdf
- Gudjons, Herbert (2008): Handlungsorientiert lehren und lernen. Schüleraktivierung – Selbständigkeit – Projektarbeit. 7. aktualisierte Auflage. Klinkhard. Bad Heilbrunn.
- Frey, Karl (2012): Die Projektmethode. Der Weg zum bildenden Tun. 12. aktualisierte Auflage. Beltz, Weinheim

9 Arbeitsmaterialien

Nr.	Titel	Beschreibung
😊 M2.1	Projektphasen	Überblick über die Projektphasen
😊 M2.2	Projektplan	Arbeitshilfe für SuS zur Konkretisierung eines Projektplans
😊 M2.3	Projektprotokoll	Arbeitshilfe zur Strukturierung des Ergebnisses und der späteren Reflexion



Legende

-  Material für Schülerinnen und Schüler
-  Material für Lehrkräfte sowie Unternehmensvertreterinnen und Unternehmensvertreter
-  Zusatzmaterial



Die Projektphasen im Überblick

1. Einführung/ Initiierung

- Die Fragestellung wird festgelegt
- Die Lehrkraft erklärt die Rahmenbedingungen
- Kleingruppen werden gebildet

2. Projektplanung

- Ziele und Aufgaben werden festgelegt
- Benötigtes Material wird gesammelt und strukturiert
- Ein Projektplan wird ausgearbeitet

3. Durchführung

- Selbständige Produktion eines Produkts in der Kleingruppe
- Die Lehrkraft ist bei Problemen ansprechbar

4. Präsentation

- Die Ergebnisse werden präsentiert

5. Reflexion

- Am Ende wird das Projekt kritisch beleuchtet:
 - Wie sind wir zum Ziel gelangt?
 - Hat alles geklappt? Wie war die Teamarbeit?
 - Etc.

Projektprotokoll

Projektgruppe (Namen der Mitglieder):

Thema des Projekts:

Projekttag (Datum): _____

Ziel - Was wollen wir heute schaffen:

Was haben wir heute geschafft:

Wo gab es Probleme?

Sonstiges Bemerkungen

Arbeitsmaterial M2.3

Projekt:				
Gruppenmitglieder:				
Zeit (Z.B. 1./2. Stunde)	Inhalt und Thema - was soll in diesem Projekt- Abschnitt passieren	Ziel - Zwischenziele, Meilensteine festlegen	Was wird dafür benötigt	To Do - was gibt es sonst noch zu tun?