



Universität
Zürich^{UZH}

Department of Informatics
Dynamic and Distributed Information Systems



Informatik und Wirtschaft

Herbst 2025 — Skript

Prof. Abraham Bernstein, Ph.D.



Inhaltsverzeichnis

I Allgemeine Informationen	7
1 Zentrale Begriffe der Informationswissenschaft	8
2 Algorithmen	17
2.1 Einführung zu Computational Thinking	17
2.1.1 Definition Computational Thinking	18
2.1.2 Was ist Computational Thinking, und was ist es nicht . . .	18
2.2 Die Hauptbestandteile von CT	19
2.2.1 Dekomposition	19
2.2.2 Mustererkennung	20
2.2.3 Abstraktion	20
2.2.4 Algorithmus	20
2.3 Weitere Fertigkeiten von CT	22
2.4 Aufgaben	25
2.5 Lösungen	26
2.6 Neutralität von Algorithmen	28
2.7 Folien	33
3 Programmierung	90
3.1 Einführung in die Programmierung	90
3.2 Programmiersprachen	90
3.2.1 Drei Generationen der Programmiersprachen	91
3.2.2 Assembler, Compiler & Interpreter	92
3.2.3 Definition von Programmiersprachen	93
3.2.4 Programmierparadigmen	93
3.3 Sequenzen und Variablen	94
3.3.1 Anweisung und Sequenzen	94

3.3.2	Variablen	95
3.3.3	Datentypen	95
3.3.4	Fallunterscheidung	95
3.3.5	Schleifen	96
3.3.6	Funktionen	97
3.4	Python	97
3.4.1	Anweisungen und Sequenzen	97
3.4.2	Fallunterscheidung	98
3.4.3	Schleifen	99
3.4.4	Bedingungskontrollierte Schleifen	99
3.4.5	Funktionen	100
3.5	Folien	101
4	Datenvisualisierung	164
4.1	Einstieg & Überblick	164
4.2	Warum	165
4.3	Was	166
4.4	Für wen	167
4.5	Wozu	170
4.5.1	Visualisierung von Mengen	170
4.5.2	Visualisierung einer Verteilung	172
4.5.3	Vergleich mehrerer Verteilungen	172
4.5.4	Visualisierung von Anteilen	172
4.5.5	Visualisierung von Zusammenhängen	174
4.6	Wie	176
4.6.1	Verbesserungsmöglichkeiten für Mengendiagramme . .	176
4.6.2	Verbesserungsmöglichkeiten für Verteilungsdiagramme	177
4.6.3	Verbesserungsmöglichkeiten für Anteildiagramme . .	177
4.6.4	Verbesserungsmöglichkeiten für Diagramme, die Zusammenhänge zeigen	177
4.7	Folien	179

5 Datenmanagement	268
5.1 Wieso Datenmanagement	268
5.2 Lebenszyklus von Daten	268
5.3 Spreadsheets	269
5.4 Datenbanksysteme	272
5.5 Datenmodellierung	273
5.6 Datenbankanfragen	277
5.7 Big Data und die Cloud	282
5.8 Rückblick und Ausblick	283
5.9 Folien	284
6 Digitalisierung	326
6.1 Digitale Daten	326
6.2 Text	335
6.3 Ton	345
6.4 Bild	354
6.5 Datenspeicherung	363
6.6 Datenformate	379
7 Arten von IS	393
7.1 Einsatzgebiete	393
7.2 Wertschöpfungskette	403
7.3 Klassifizierung	412
8 Wert von IS und Information	419
8.1 Teil I	419
8.2 Teil II	432
8.3 Wert von Informationen	439
8.3.1 Einführung	439
8.3.2 Musikproduktion: To Studio or not to Studio?	439

8.3.3 Berechnungen: Eine Erwartungswertrechnung	440
8.3.4 Wert von perfekter Information	443
9 Digitale Güter	445
9.1 Eigenschaften	445
9.2 Marktverzerrungen	460
9.3 Wettbewerbsstrategien	465
9.4 Preisdifferenzierung	469
9.5 Produktdifferenzierung	476
9.6 Produktbündelung	485
10 Datenrecht	505
10.1 Einführung	505
10.2 Recht und Gesellschaft	505
10.3 Rechtsgebiete	506
10.4 Grundbegriffe	507
10.5 Daten als Gegenstand des Rechts	509
10.6 Grundsätze der Zuordnung von Daten	511
10.7 Patentrecht	512
10.8 Urheberrecht	512
10.9 Wettbewerbsrecht (UWG)	512
10.10 Vertragsrecht	513
10.11 Folien	515
11 Datenschutzrecht	534
11.1 Einführung	534
11.2 Geltungsbereich	534
11.3 Ziele des Datenschutzrechts	535
11.4 Rechtsquellen	536
11.5 Funktionsweise des Datenschutzrechts	536

11.6 Datenbearbeitungsgrundsätze	537
11.7 Rechtfertigungsgründe	538
11.8 Rechtsdurchsetzung	539
11.9 Zuordnungsinstrument	540
11.10 Folien	542
12 KI & Machine Learning	560
12.1 Begrüßung	560
12.2 Was ist künstliche Intelligenz?	561
12.3 Meilensteine der KI	562
12.4 Wissen, Schlussfolgern, Planen	563
12.5 Prototypische KI-Anwendungen	565
12.6 Der Turing-Test	565
12.7 Dialogsysteme, Teil I	566
12.8 Dialogsysteme, Teil II	566
12.9 Introduction to Robotics	567
12.10 Auswirkungen der KI	568
12.11 Maschinelles Lernen	568
12.12 Zentrale Konzepte ML, Teil I	569
12.13 Deep Learning & neurale Netze	570
12.14 Abschluss	572
12.15 Folien	574
A Skript: Mikroökonomie	622
A.1 Einleitung	623
A.1.1 Lehrziele	623
A.1.2 Aufbau des Selbstlernmoduls Mikroökonomie	623
A.2 Angebot & Nachfrage	625
A.2.1 Lehrziele	625
A.2.2 Märkte	625

A.2.3	Die Nachfrage	626
A.2.4	Das Angebot	628
A.2.5	Marktgleichgewicht	631
A.2.6	Komparative Statik	633
A.2.7	Konsumentenrente	635
A.2.8	Nachfrage- und Angebotsüberschuss	637
A.2.9	Zusammenfassung	639
A.3	Produktions- & Kostentheorie	640
A.3.1	Lehrziele	640
A.3.2	Einleitung	640
A.3.3	Produktionsfunktion	640
A.3.4	Kostenfunktion	644
A.3.5	Zusammenfassung	646
A.4	Vollkommener Wettbewerb	647
A.4.1	Lehrziele	647
A.4.2	Einleitung	647
A.4.3	Gewinnmaximierung	647
A.4.4	Zusammenfassung	651
A.5	Zusammenfassung	652
A.6	Literatur	653

I Allgemeine Informationen

Die vollständigen Informationen zur Vorlesung finden Sie im OLAT: <https://lms.uzh.ch/auth/RepositoryEntry/17745707598/CourseNode/112400698471114>.

1 Zentrale Begriffe der Informationswissenschaft

PROF. ABRAHAM BERNSTEIN



Institut für Informatik

Zentrale Begriffe der Informationswissenschaft

Prof. Abraham Bernstein, Ph.D.

1 ZENTRALE BEGRIFFE DER INFORMATIONSWISSENSCHAFT

Zentrale Begriffe der Informationswissenschaft

Begriffe & Definitionen

Zentrale Begriffe der Informationswissenschaft: Daten

Daten = Zeichen/Symbole/Signale/„rohe“ Fakten



Zentrale Begriffe der Informationswissenschaft: Information

Information = Daten + Semantik*



* In der Informatik bezeichnet „Semantik“ die formale Semantik von Sprachen.

B ist ein Buchstabe

Zentrale Begriffe der Informationswissenschaft: Information

Information = Daten + Semantik



B ist eine Zahl

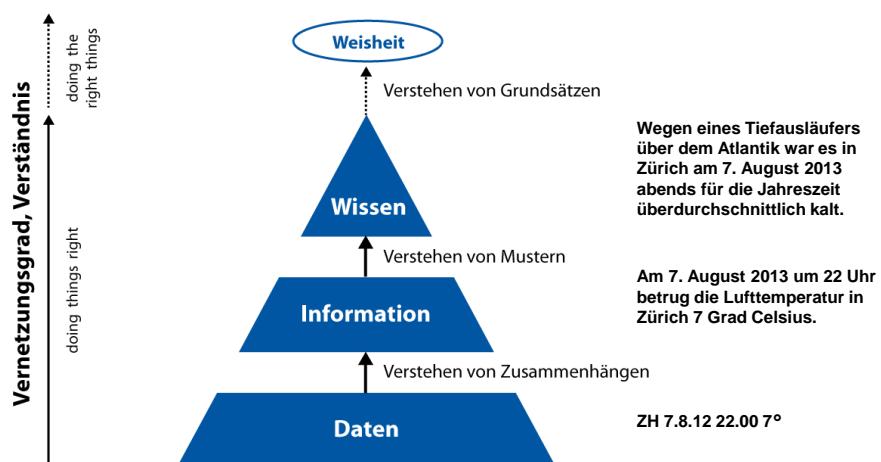
Zentrale Begriffe der Informationswissenschaft: Wissen

Wissen = Information + Verknüpfung



3 ist eine Primzahl und damit nur durch sich selbst und eins teilbar

Zusammenhänge zwischen Daten, Information und Wissen



Zusammenhänge zwischen Daten, Information und Wissen: Beispiel

Daten sind Fakten

- Können **verschieden ausgedrückt** oder **dargestellt** werden
 - Hans hat Note 4 erreicht
 - Eva hat Note 1 erreicht
 - Anna hat Note 5 erreicht
- Sind **unabhängig von ihrer Interpretation**
 - Note 1 kann als schlechte (Schweiz) oder gute (Deutschland) Leistung interpretiert werden. Das ändert aber nicht den Fakt, dass Eva die Note 1 hat.

Beispiel: Resultate von Studierenden

Hans	4
Anna	5
Eva	1

Sie sind auch in anderer **Anordnung** gleich

Eva	1
Hans	4
Anna	5

Zusammenhänge zwischen Daten, Information und Wissen: Beispiel

Informationen entstehen aus unterschiedlicher **Darstellung** der selben Daten.

z.B. Teilnehmerlisten:

- Anna
- Eva
- Hans

... oder Notenlisten:

Hans	4
Anna	5
Eva	1

Information wird auch von der **Bedeutung** der Daten bestimmt...

z.B. Ranglisten:

1. Anna
2. Hans
3. Eva

... oder deren **Interpretation**:

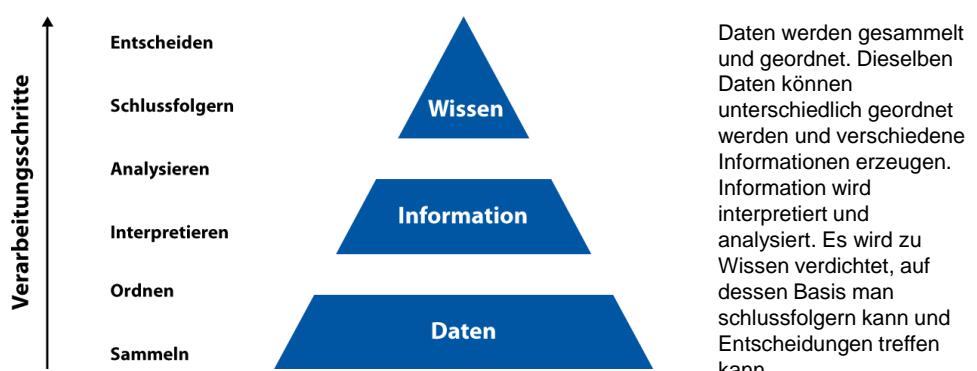
- Anna und Hans haben die Prüfung bestanden.
- Eva hat die Prüfung nicht bestanden.

Zusammenhänge zwischen Daten, Information und Wissen: Beispiel

Wissen entsteht bei der **Analyse** von Information...

- Die meisten Teilnehmer haben die Prüfung bestanden.
 - Die Lehrveranstaltung war dieses Jahr nicht gut besucht.
 - Die Durchschnittsnote war 3.33.
- ... und bedarf oft der **Verknüpfung** mit zusätzlicher Information oder zusätzlichem Wissen.
- Die Prüfungsschwierigkeit war angemessen.
 - Eva hat nicht an der Prüfung teilgenommen.
 - Anna hat die Lehrveranstaltung schon im Vorjahr besucht, aber damals nicht bestanden.

Schritte der Informations- bzw. Wissensverarbeitung



Zentrale Begriffe der Informationswissenschaft: Informationstechnologie

Informationstechnologie:
Ein Werkzeug um Daten zu

- Sammeln
- Übertragen
- Speichern
- Verarbeiten

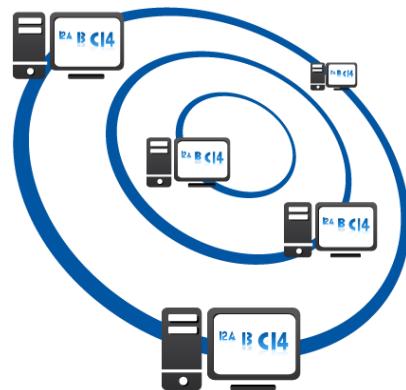


Beispiel: Buch: Es ist ein Werkzeug zum Sammeln, Übertragen, Speichern und Verarbeiten von Daten.

Zentrale Begriffe der Informationswissenschaft: Informationssystem

Informationssystem:
Kombination von Hardware, Software und
Netzwerken, die Menschen beim
Sammeln, Kreieren und Verteilen
wichtiger Daten hilft.

Beispiel: Bibliothek: Es ist eine Kombination von
Hardware (Bücher), Software (z.B. Texte in
Büchern) und es kann als Netzwerk, welches dem
Menschen beim Sammeln, Kreieren und Verteilen
wichtiger Daten hilft, interpretiert werden.



Zentrale Begriffe der Informationswissenschaft: Digitale Güter

Digitale Güter:

Produkte oder Dienstleistungen,
in Form von Binärdaten hergestellt,
verarbeitet, übertragen und konsumiert.



Beispiel: YouTube Video: Es kann digital gespeichert und übertragen werden.

Fazit

- Daten werden gesammelt, zu Informationen verdichtet und zu Wissen vernetzt
- Informationstechnologie: hilft beim Speichern und Verarbeiten von Daten
- Informationssystem: hilft beim Erfassen dieser Daten
- Digitale Güter: Produkte, die in binärer Form existieren

Credits

- Bild Schachteln: antrepostop.com
- Kuhlen, R., Informationsmarkt. Chancen und Risiken der Kommerzialisierung von Wissen, Konstanz 1995
- Shapiro, C., Varian, H.R., Information Rules. A Strategic Guide to the Network Economy, Boston 1999

2 Algorithmen

PROF DR. THOMAS FRITZ UND CLAUDIA VOGEL

Studium Digitale Kursbaustein 8

Begleit-Skript

Dieses Skript bereitet die Inhalte des Kursbaustein 8 – Computational Thinking schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

2.1 Einführung zu Computational Thinking

Computational Thinking (CT) ist entscheidend unsere moderne Welt, die voller Technologie geprägt ist, zu verstehen. Algorithmen und Technologie sind überall in unserem heutigen Leben anzutreffen, haben bereits unsere Art zu leben verändert und werden dies in Zukunft noch stärker tun. Könnten Sie Sich heute noch ein Leben ohne Google Suche vorstellen oder ohne eine Navigationsapplikation zu Ihrem Ziel gelangen? Computational Thinking beeinflusst unser Leben durch die Algorithmen die wir täglich nutzen. Deshalb ist neben dem Verstehen, wie Algorithmen funktionieren und ob man ihnen vertrauen kann, auch wichtig, algorithmisch denken zu können. Diese Denkweise führt zu dem Wandel in unserem Leben und es erlaubt uns aktiv mitzugestalten und grossartige Ideen auch in Realität umzusetzen.

Generell befasst sich Computational Thinking mit der Fähigkeit von Menschen, Probleme in einer systematischen Art und Weise zu lösen. Das Resultat dieses Prozesses sind Algorithmen. Bevor ein Problem jedoch in Angriff genommen werden kann und wir einen Algorithmus haben der das Problem löst, müssen das Problem selbst und die Art und Weise, in der es gelöst werden kann, verstanden werden. Hier kommt CT zum Einsatz. Es hilft uns die Lücke zwischen der Komplexität realer Probleme und deren Lösungen die von Computern oder auch Menschen verstanden und ausgeführt werden können zu schliessen. CT ermöglicht es uns, ein komplexes Problem zu verstehen und mögliche Lösungen zu entwickeln und so zu präsentieren dass sie ausführbar sind.

2.1.1 Definition Computational Thinking

Bis heute gibt es verschiedene Versionen wie Computational Thinking definiert wird und es wurde kein Konsens über eine gemeinsam vereinbarte Definition gefunden. Der Begriff wurde 1996 von Seymour Papert kreiert und durch Jeanette Wing in 2006 populär gemacht. Wing und Papert definieren Computational Thinking als die

«Fähigkeit, ein Problem in kleinere Probleme zu zerlegen, die deterministisch gelöst werden können.» – Wing (2006) & Papert (1996)

Beim CT geht es spezifisch um die Fähigkeit, ein Problem in kleine Teilprobleme zu zerlegen, die wiederum eindeutig gelöst werden können, das Ergebnis führt zu einem Algorithmus. Es umfasst das Finden von kreativen Lösungskonzepten, und dass Verstehen dieser Konzepte, so dass sie einer Maschine oder einem anderen Menschen beigebracht werden oder von ihnen ausgeführt werden können.

«Die **Denkprozesse**, die bei der Formulierung von **Problemen** und deren **Lösungen** ablaufen, so dass die Lösungen in einer Form dargestellt werden, die von einem informationsverarbeitenden Agenten effektiv durchgeführt werden kann.» – Cuny, Snyder, Wing (2011)

2.1.2 Was ist Computational Thinking, und was ist es nicht

Wing (2006) grenzt Computational Thinking wie folgt ab:

- CT ist Konzeptualisierung und nicht Programmierung: Wie ein Informatiker zu denken, bedeutet mehr als zu Programmieren und Computerprogramme zu schreiben. Es erfordert ein Denken auf mehreren Abstraktionsebenen und das analytische angehen von Problemen.
- CT ist eine Art und Weise, wie Menschen denken und Probleme lösen. Es beabsichtigt nicht Menschen dazu zu bringen wie Computer zu denken.
- Der Fokus von CT liegt auf Ideen und Prozessen, nicht auf physischen Software- oder Hardware Artefakten, die aus CT entstehen können. Es geht um Konzepte mit deren Hilfe wir Probleme angehen und lösen, unser tägliches Leben meistern und mit anderen Menschen kommunizieren und interagieren.
- CT ist für jeden und überall, nicht nur für Informatiker.

2.2 Die Hauptbestandteile von Computational Thinking

Die vier Hauptbestandteile von Computational Thinking sind:

- Dekomposition,
- Mustererkennung,
- Abstraktion, und
- Algorithmen.

In den folgenden Kapiteln stellen wir diese Bestandteile genauer vor.

2.2.1 Dekomposition

Dekomposition erlaubt es, ein Problem in kleinere Teilprobleme herunterzubrechen, so dass man sich auf das Lösen dieser kleineren und einfacheren Teilprobleme fokussieren kann. Dies ist vor allem bei grösseren, komplexeren Problemen sehr wichtig, denn nur mit Zerlegung ist es möglich eine Anwendung zu schreiben die, wie zum Beispiel die Google Suche, Millionen von Schritten und Anweisungen beinhaltet für die Lösung.

Der grosse Vorteil der Zerlegung ist, dass die Komplexität eines Problems reduziert wird. Das Zerlegen des Problems in kleinere Teile bedeutet, dass jedes selbstständig untersucht und gelöst werden kann. Zudem ist es einfacher einen Überblick über das gesamte Problem zu wahren.

Es gibt unterschiedliche Methoden zur Zerlegung eines Problems in Teilprobleme. Eine davon ist die Problemlösung durch Zielreduktion. Dies ist eine Methode, die eine Zerlegung in kleinere, aber sonst gleiche Teile erlaubt. Beispielsweise wird das Problem halbiert. Das führt dann zu zwei gleich grossen Teilen, die kleiner und leichter lösbar sind als das Ausgangsproblem.

Eine weitere Methode ist die rekursive Problemlösung mit Zielreduktion. Dabei wird die Teilung von einem Problem wiederholt angewendet. Ein Beispiel dafür ist, wenn ein Name im Telefonbuch gesucht wird. Erst wird das Telefonbuch in der Mitte aufgeschlagen und geprüft ob der gesuchte Namen vor oder nach der aufgeschlagenen Seite aufgeführt ist. Im nächsten Schritt beschränken wir uns dann auf den Teil, in dem der Name vorkommen sollte. Wir haben nun ein ähnliches, aber kleineres Problem: wir suchen nun den Namen nur noch in der Hälfte des Telefonbuches. Dafür wird das Buch nochmals in der Hälfte dieses einen Teiles, in dem der Name vorkommt, aufgeschlagen, und so weiter bis wir den Namen gefunden haben.

2.2.2 Mustererkennung

Ein weiterer zentraler Bestandteil von CT ist die Mustererkennung. Dabei werden Gemeinsamkeiten oder Ähnlichkeiten zwischen den einzelnen Teilproblemen sowie auch zwischen Problemen identifiziert. Dies ermöglicht eine Vereinfachung eines komplexen Problems durch die Wiederverwendung der gleichen Lösung für gleiche oder ähnliche Probleme. Hat man beispielsweise bereits ein Problem gelöst, das Gemeinsamkeiten mit dem aktuellen Problem aufweist, kann auf die bereits existierende Lösung zurückgegriffen werden. Man muss somit nicht immer wieder von Neuem beginnen und kann Erfahrungen und Wissen in die aktuelle Problemlösung mit einbeziehen.

2.2.3 Abstraktion

Ein weiterer wichtiger Bestandteil von CT ist es die wichtigen Aspekte eines Problems zu identifizieren und die unwichtigen Aspekte wegzulassen. Diese Abstraktion—also der Fokus auf relevante Eigenschaften und das Verbergen der irrelevanten Details—erlaubt es uns die Komplexität besser zu bewältigen und erleichtert die Lösung eines Problems.

Es gibt unterschiedliche Anwendungen von Abstraktion, beispielsweise die Abstraktion der Steuerung oder die Datenabstraktion. Bei der Abstraktion der Steuerung werden Anweisungen so gruppiert, dass sich neue Anweisungen ergeben, die grössere Schritte darstellen. In Folgen, können wir diese grösseren Schritte verwenden um ein Problem zu lösen und die Details der notwendigen Einzelschritte verbergen. Bei der Datenabstraktion werden irrelevante Details der Daten verborgen, wie zum Beispiel die Speicherung der Daten oder die genaue Repräsentation.

2.2.4 Algorithmus

CT ist ein Denkprozess um Probleme zu lösen und anzugehen. Die daraus resultierenden Lösungen sind meist Algorithmen. Ein Algorithmus ist eine Reihe von Anweisungen oder eine Beschreibung aus mehreren Anweisungen, um etwas zu erreichen:

«*Ein Algorithmus ist eine endliche Folge wohldefinierter Anweisungen, typischerweise zur Lösung einer Klasse von Problemen.*»

«*[...]. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten. [...] Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.*» – Leiserson et al. (2010)

Aus dieser Definition sind folgende Eigenschaften eines Algorithmus ableitbar:

- Ein Algorithmus ist ein endlicher Text in eindeutiger Sprache der eine Abfolge von Schritten festlegt. Algorithmen können zur Ausführung in einem Computerprogramm implementiert, aber auch in menschlicher Sprache formuliert werden. Die Abfolge von Schritten erzeugen aus der Eingabe eine Ausgabe.
- Die Repräsentation und der Wertebereich der Eingabe und Ausgabe/Ergebnisse müssen eindeutig definiert sein.
- Jeder Schritt des Algorithmus muss tatsächlich ausführbar sein.
- Die Schritte sind Basisoperationen, d.h. sie sind vom Computer oder Menschen direkt ausführbar, oder sie sind selbst wieder durch einen Algorithmus definiert.

Wenn für die Problemlösung ein Algorithmus vorhanden ist, sollte jeder das Problem lösen können, ohne den Algorithmus davor zu kennen oder verstehen zu müssen, was dieser im Detail macht. Solange die Schritte genau befolgt werden, wird der Algorithmus zur richtigen Lösung führen. Somit ist das Ziel eines Algorithmus, dass eine erstellte Beschreibung aus einfachen Anweisungen von einer Maschine oder jemandem anderem ausgeführt werden kann, um eine bestimmte Problemstellung zu lösen. Daraus folgt auch dass eine Maschine jede Variation des Problems lösen kann, indem sie einfach mechanisch den Anweisungen folgt. Denn das ist alles, was Computer tun: Sie folgen Algorithmen, die von Menschen geschrieben wurden.

Kriterien von Algorithmen

Es gibt drei Kriterien, die Algorithmen erfüllen müssen um von einem Computer ausgeführt werden zu können (Lustenberger, 2008): Maschinentauglichkeit, Allgemeinheit und Korrektheit.

Das erste Kriterium ist die Maschinentauglichkeit: ein Algorithmus muss von einer einfachen Maschine ausgeführt werden können. Dass heisst, der Algorithmus muss aus einer Beschreibung von einfachen Anweisungen bestehen, die von einer Maschine gelesen werden kann und die Maschine muss basierend auf der Beschreibung die beschriebenen Tätigkeiten ausführen können.

Das zweite Kriterium ist die Allgemeinheit: ein Algorithmus muss auf alle Problemfälle des zu lösenden Problems anwendbar sein. Beispielsweise wenn man einen Algorithmus zur Sortierung von Zahlen entwickelt, muss dieser für beliebige Zahlen und eine beliebige Anzahl von Zahlen anwendbar sein.

Das letzte wichtige Kriterium ist die Korrektheit: ein Algorithmus muss für alle Fälle eines gegebenen Problems in endlicher Zeit die richtige Lösung liefern.

2.3 Weitere Fertigkeiten von Computational Thinking

Computational Thinking bringt unterschiedliche Fertigkeiten zusammen, die vereint eine leistungsfähige Art des Denkens ergeben. Zu diesen Fähigkeiten gehören Modellierung, wissenschaftliches und logisches Denken, Kreativität, Heuristiken, Menschen zu verstehen und die Evaluation von Algorithmen (Curzon & McOwen 2018).

Computer Modellierung

Bei der Computer Modellierung wird ein Algorithmus erstellt, der die Realität simuliert, d.h. etwas aus der realen Welt wird in einer virtuellen Welt abgebildet. Dadurch können Experimente durchgeführt werden, die beispielsweise bei einer realen Durchführung zu lange dauern oder nicht möglich sind. Ein Beispiel für eine Computermodellierung ist eine Wettervorhersage: Indem man die reale Welt im Computer modelliert, kann man Vorhersagen darüber treffen, was in der Realität geschehen wird, etwa ob es morgen in einer bestimmten Region regnen wird oder nicht.

Wissenschaftliches Denken

Wissenschaftliches Denken ist das strukturierte und systematische Vorgehen für Erklärungen und um Vorhersagen zu treffen. Dies bedeutet, dass man eine Fragestellung so angeht, dass man Erkenntnisse gründlich recherchiert, diese systematisch durcharbeitet und versucht, die vorliegenden Erkenntnisse zu verstehen, zu bewerten und daraus Schlüsse für die eigene Fragestellung zu ziehen. Zum Schluss wird eine Hypothese aufgestellt, die dann durch weitere Daten überprüft wird.

Logisches Denken

Zum Computational Thinking gehört auch das logische Denken. Es erlaubt uns Zusammenhänge zu erschliessen und Muster zu erkennen. Es ist ein folgerichtiges und schlüssiges Denken und führt dazu, dass alle Eventualitäten sowie Ausnahmefälle berücksichtigt und Details beachtet werden.

Kreativität

Eine weitere Fertigkeit ist die Kreativität. Probleme zu lösen und Ideen zu entwickeln sowie auch deren Konzepte zur Umsetzung in der Realität auszuarbeiten ist ein kreativer Prozess. Lösungen für ein Problem können sehr kreativ sein und alte oder neue Probleme innovativ lösen.

Heuristik

In Algorithmen kommen häufig Heuristiken ins Spiel, vor allem da es nicht immer möglich ist die allerbeste Lösung in absehbarer Zeit zu finden. Deshalb gibt es heuristische Algorithmen, welche die bestmögliche, vernünftige Lösung in angemessener Zeit finden.

Menschen verstehen

Beim Computational Thinking ist es auch wichtig auf den Menschen zu achten und Menschenkenntnis einfließen zu lassen. Insbesondere da Algorithmen oft Menschen unterstützen, ist es wichtig die Stärken und Schwächen von Menschen zu verstehen und in der Technologie, resp. dem Algorithmus, zu berücksichtigen.

Menschen verstehen – Post Completion Error

Das folgende Rätsel stammt von Curzon & McOwen (2018).

Aufgabe:

Eine Bäuerin ist mit ihrem Schäferhund Maxi, der sie immer begleitet, auf dem Weg zum nächsten Dorf. Um dorthin zu gelangen, muss sie einen reissenden Fluss überqueren. Dafür hat eine Erfinderin, die auf der Dorfseite des Flusses wohnt, eine Vorrichtung erfunden, die die Flussüberquerung ermöglicht. Sie besteht aus einem Seil, Rollen und einem Sitz, der an dem Seil hängt und Platz für eine Person bietet. Die Bewohner des Ortes haben vereinbart, dass der Sitz immer auf der Seite des Dorfes zu lassen, wo die Erfinderin wohnt. Als die Bäuerin zum Fluss kommt, zieht sie den Sitz mit dem Seil zu sich herüber, setzt sich hinein, nimmt Maxi auf den Arm, zieht sich selbst auf die andere Seite und setzt ihren Weg zum Dorf fort. Eines Tages kauft sie eine Henne und einen Sack Reis. Als sie auf dem Heimweg zum Fluss kommt, stellt sie fest, dass sie nur ein Ding mit sich hinübernehmen kann, wenn sie den Sitz verwenden möchte. Deshalb benötigt sie mehrere Fahrten. Dabei gibt es folgendes Problem:

Wenn sie die Henne mit dem Getreide auf einer Seite allein lässt, wird die Henne das Getreide fressen. Lässt sie Hund Maxi und die Henne auf einer Seite zusammen, wird der Hund die Henne jagen. Maxi frisst kein Getreide, deshalb kann er mit dem Getreide allein gelassen werden.

Schreiben Sie einen Algorithmus um alle drei Sachen auf die andere Flussseite zu bringen, ohne dass etwas gefressen oder gejagt wird.

Lösung:

1. Die Bäuerin fährt mit der Henne auf die andere Seite.

2. Die Bäuerin kehrt alleine zurück.
 3. Die Bäuerin fährt mit dem Hund auf die andere Seite.
 4. Die Bäuerin kehrt mit der Henne zurück.
 5. Die Bäuerin fährt mit dem Getreide auf die andere Seite.
 6. Die Bäuerin kehrt alleine zurück.
 7. Die Bäuerin fährt mit der Henne auf die andere Seite.
- 8. Die Bäuerin schickt den leeren Sitz zum anderen Ufer zurück.**

Die meisten Lösungen für dieses Problem, vielleicht auch Ihre, enden bei Schritt 7. Bei der Lösung dieser Aufgabe konzentrieren wir uns oft sehr stark auf das Ziel die einzelnen Dinge sicher über den Fluss zu bringen. Da wir Menschen ein begrenztes Kurzzeitgedächtnis haben, kommt es öfters vor, dass die Erfüllung eines Teilziels, das zum korrekten Abschluss des Hauptziels, vergessen geht. In diesem Beispiel ist der letzte Schritt um das Problem zu lösen, dass die Bäuerin den leeren Sitz an das andere Ufer zurück sendet. Insbesondere wenn Menschen bereits viele Informationen im Kurzzeitgedächtnis haben, passiert es, dass wir diesen Schritt vergessen. Dieser Fehler wird auch Post-Completion-Error genannt. Dabei handelt es sich um Fehler, bei denen Menschen den letzten Schritt, respektive das letzte Teilziel vergessen, um ein Problem vollständig zu lösen. Wenn wir Menschenkenntnisse mit in unsere Algorithmen einfließen lassen, können wir solche Fehler vermeiden und Menschen besser unterstützen.

Ein Beispiel für einen Algorithmus der einen Post-Completion Fehler vermeidet ist bei Bankautomaten anzutreffen. Die meisten Bankautomaten geben dem Nutzer erst Geld aus, wenn die Karte entnommen wurde. Bei Automaten die zuerst das Geld ausgeben, kommt es häufiger vor, dass Menschen die Karte am Ende vergessen.

Evaluation

Die Evaluation stellt sicher, dass der Algorithmus funktioniert und eine den Zweck erfüllt. Dabei ist es auch wichtig zu überprüfen, ob der Algorithmus den Anforderungen entspricht, die das Problem beschreiben. Neben diesen Aspekten können auch weitere Aspekte wie die Benutzerfreundlichkeit oder Leistungsfähigkeit des Algorithmus evaluiert werden. Es ist wichtig, den Algorithmus nicht nur am Schluss, wenn er fertig entwickelt wurde, zu evaluieren, sondern den Algorithmus über den ganzen Prozess hinweg kontinuierlich zu evaluieren.

2.4 Aufgaben

Begabtenförderung

Jährlich erhält eine Stiftung zur Förderung eines Studiums mehrere hundert Bewerbungen, von denen sie 20 mit einem Stipendium fördert. Entwerfen Sie einen Algorithmus, der eine Rangliste der Bewerber*innen und derer CVs basierend auf den nachfolgenden Kriterien erstellt, und die besten 20 auswählt.

Förderungskriterien:

- Notenschnitt von mindestens 5.3 (Matura, Berufsmatura)
- Alter von höchstens 28 Jahren
- Kenntnisse von mindestens zwei Amtssprachen
- Gesellschaftliches Engagement, breite Interessen, intellektuelle Neugier und Kreativität

Tipp:

Um eine Rangliste zu erstellen, benutze ein Punktesystem dass basierend auf gewissen Charakteristiken die Bewerber*innen bewertet. Zum Beispiel könnte es pro gespieltem Musikinstrument +1 Punkt geben usw..

Meetingplanung

Für ein Seminar gibt es drei Gruppen die mit dem Teaching Assistant ein wöchentliches Meeting am Donnerstag vereinbaren sollen. Jede Gruppe schreibt dem TA eine E-Mail mit drei passenden Zeitfenstern mit Priorisierung.

Entwerfen Sie einen Algorithmus, der die Meetings für den TA plant so dass es keine Konflikte gibt.

Beispiele für Daten, die der TA von den Studenten erhält:

- Gruppe 1 reicht folgende Termine ein:
 - 1. Priorität: 9 - 10 Uhr
 - 2. Priorität: 10 - 11 Uhr
 - 3. Priorität: 11 - 12 Uhr
- Gruppe 2 reicht folgende Termine ein:

- 1. Priorität: 10 - 11 Uhr
 - 2. Priorität: 11 - 12 Uhr
 - 3. Priorität: 13 - 14 Uhr
- Gruppe 1 reicht folgende Termine ein:
- 1. Priorität: 9 - 10 Uhr
 - 2. Priorität: 11 - 12 Uhr
 - 3. Priorität: 16 - 17 Uhr

2.5 Lösungen

Begabtenförderung

Um den Algorithmus zur Auswahl der Kandidaten zu erstellen wird das Problem zuerst in kleinere Probleme zerlegt. Anschliessend werden Gemeinsamkeiten zwischen den Teilproblemen identifiziert und die Daten abstrahiert. Zum Schluss wird der Algorithmus definiert.

Dekomposition

Das Problem lässt sich in folgende Teilprobleme unterteilen:

- Überprüfe Förderungsfähigkeit der Bewerber*in:
- Alter von höchstens 28 Jahren
- Kenntnisse von mindestens zwei Amtssprachen
- Notenschnitt des Maturitätsabschluss von mindestens 5.3
- Bewerte Bewerber*in (CVs) basierend auf gesellschaftlichem Engagement, breiten Interessen, intellektueller Neugier und Kreativität
- Erstelle Rangliste und wähle Top 20 aus

Abstraktion & Muster

Die Bewerbungen werden alle nach gleichem Schema/Ablauf bearbeitet:

- Überprüfung der Förderungsfähigkeit ist gleich für alle Bewerber*in
- Bewertung sollte für Bewerber*innen gleich ablaufen

Bei der Erstellung des Algorithmus legt man den Fokus auf die für das Problem relevanten Eigenschaften der Bewerber*innen. Diese sind Charakteristiken, die für die Überprüfung der Aufnahmekriterien notwendig sind, dazu gehören unter anderem Alter, Sprachkenntnisse, Notendurchschnitt, Anzahl gesellschaftlicher Engagements, Musikinstrumente.

Algorithmus

Durch die Zerkleinerung des Problems, sowie Mustererkennung und Abstraktion kann der Algorithmus einfacher hergeleitet werden. Der Algorithmus wählt die Top 20 Kandidaten aus den Bewerbungen aus. Dazu wird für jede eingegangene Bewerbung zuerst überprüft, ob der/die Kandidat*in förderfähig ist und die Aufnahmekriterien erfüllt. Anschliessend wird für jede/jeden Kandidat*in bewertet indem eine Punktzahl berechnet wird.

1. Für jede*n Bewerber*in
 1. Prüfe ob Bewerber*in förderfähig ist: Alter <= 28 Jahre && Anzahl Sprachen >= 2 Amtssprachen && Notenschnitt >= 5.3
 2. Bewerte CV / Berechne Punktezahl
 1. +1 Punkt pro gesellschaftlichen Engagement (sozialer Verein, etc.)
 2. +1 Punkt pro Musikinstrument
 3. +1 Punkt pro extracurricularem Projekt
 4. + ((Notenschnitt – 5.3) * 10) Punkte // +1 pro 0.1 im Notenschnitt über 5.3
 5. + (28 – Alter) Punkte // je schneller zum Studium desto besser
 2. Erstelle Rangliste basierend auf Bewertungen für jeden Bewerber*in
 3. Wähle die Top 20 der Rangliste aus

('//' bedeutet Kommentar, d. h. ist keine ausführbare Aktion im Algorithmus)

Meetingplanung

Ein möglicher Algorithmus, der die Termine für den TA plant kann wie folgt aussehen:

1. Gehe durch alle Gruppen nach Zeitpunkt der Email (first come, first serve)
 - 1.1 Für aktuelle Gruppe, gehe durch die 3 Zeitfenster nach Priorisierung
 - 1.1.1 Ist das aktuelle Zeitfenster im TA Kalender frei?
Ja: weise dieses Zeitfenster zu und gehe zur nächsten Gruppe in Schritt 1
Nein: gehe zum nächsten Zeitfenster und wiederhole Schritt 1.1.1
2. Sende zugewiesene Zeitfenster an Studierende

2.6 Neutralität von Algorithmen

Algorithmen sind nicht zwingend neutral, d.h. dass sie aus verschiedenen Gründen bestimmte Personengruppen oder Minderheiten diskriminieren. Denn Algorithmen werden von Menschen entworfen und basieren auf Daten, die unsere Realität widerspiegeln. Der Hauptgrund liegt darin, dass Menschen einen Bias haben (können).

Bias ist ein Überbegriff für Vieles, von Vorurteilen über Verzerrungen in Entscheidungen bis hin zur Förderung oder Vernachlässigung bestimmter gesellschaftlicher Gruppen. Ein Bias ist nicht unbedingt gewollt oder überhaupt bewusst und man unterscheidet generell zwei Arten von Bias: bewusster und unbewusster Bias. Beim **bewussten Bias** weiss die Person, dass sie einen Bias, also zum Beispiel Vorurteile, hat. Beim **unbewussten Bias** kennt die Person ihre Vorurteile/Verzerrungen nicht.

Ein Beispiel des unbewussten Bias ist der Affinity Bias, oder auch Similarity Bias. Bei diesem Bias erfahren Menschen eine Tendenz sich mit Leuten verbunden zu fühlen, die einen ähnlichen Hintergrund/Interessen oder Erfahrungen haben wie sie selbst. Dies kann beispielsweise zu Implikation bezüglich Diversität am Arbeitsplatz haben, denn man wählt verstärkt die Bewerber aus die einem ähnlich sind.

Die **Bias-Blindheit** ist die Tendenz von Menschen zu glauben, dass sie selbst keinen Bias haben. Es gibt Tests wie den «**Implicit Association Test**¹» den man durchführen kann um sich selbst auf gewissen unbewussten Bias zu testen.

Bei Algorithmen kann Bias auch häufig vorkommen. Man spricht dabei von einem **algorithmischen Bias**:

«*Algorithmischer Bias beschreibt systematische und wiederholbare Fehler in einem Computersystem, die unfaire Ergebnisse erzeugen, wie zum Beispiel die Privilegierung einer willkürlichen Gruppe von Benutzern gegenüber anderen.*»

¹<http://www.understandingprejudice.org/iat/>

In der Theorie können Systeme dabei helfen, Diskriminierung sichtbar zu machen und abzubauen. Aber mit zunehmender Entscheidungsübernahme von Algorithmen über das menschliche Leben, wird die Problematik von algorithmischen Verzerrungen verstärkt. Ein **Grund** für algorithmischen Bias kann der/die Ersteller*in/Benutzer*in selbst sein, denn Algorithmen agieren nicht unabhängig von Menschen, die sie erstellen oder einsetzen. Dies führt dazu, dass die implizite Werte und damit auch der bewusste als auch der unbewusste Bias der Entwickler*innen im Algorithmus anzutreffen ist. Des weiteren sind vor allem datenbasierte Algorithmen nur so gut wie die Daten, mit denen sie arbeiten. Ein historischer Bias in Daten kann zu struktureller Benachteiligung führen, z. B. im Bereich Recruiting: wenn bei Bewerbungen eine Vorauswahl durch ein Computer Programm getroffen wird, das dann zum Beispiel automatisch Personen über 60 Jahre oder mit einem ausländischen Nachnamen aussortiert. Viele Modelle, die heutzutage zum Beispiel bei Suchmaschinen oder anderen Applikationen im Hintergrund eingesetzt werden, werden häufig auf grossen existierenden Datenmengen, wie allen Wikipedia Einträgen trainiert. Dabei kann ein inhärenter Bias auftreten, denn diese Einträge werden auch vorwiegend von bestimmten Menschengruppen geschrieben die zum Beispiel den Zugang und die Zeit dafür haben.

Aufgabe 1 Bias im Algorithmus

Diese Aufgabe basiert auf dem erstellten Algorithmus zur Auswahl der Top 20 Kandidaten, die eine Begabtenförderung erhalten. Die besten Kandidaten werden aus einer Rangliste ausgelesen, die sich aus einer Punktzahl von verschiedenen Kriterien, wie z. B. Sprachkenntnisse, gesellschaftliches Engagement, intellektuelle Neugier, zusammensetzt.

Schauen Sie Sich Ihren Algorithmus an und überlegen Sie, ob es möglich wäre, dass ein Bias im Algorithmus vorliegt. Falls ja, wo liegt dieser Bias vor?

Aufgabe 2 Bias im Algorithmus

Unter den Bewerbern gibt es folgende zwei Kandidaten:

Anna K.	Felix M.
<ul style="list-style-type: none"> • 24 Jahre alt • Mexikanische Staatsbürgerschaft • seit mehreren Jahren in der Schweiz wohnhaft • Notendurchschnitt von 5.7 bei der Matura für Erwachsene • Alleinerziehende Mutter, Unterstützung ihrer Familie in Mexiko 	<ul style="list-style-type: none"> • 20 Jahre alt • Schweizer Staatsbürgerschaft • wohnhaft in der Schweiz • Notendurchschnitt von 5.4 in der Kantonsschule • Mitglied im Orchester und Schachclub • Spielt E-Gitarre, Kontrabass und Violine • Freiwilliger Helfer bei Ponte-Sano

Gehen Sie den Algorithmus für beide Kandidaten durch und schauen Sie, wie viele Punkte jeder der beiden erhält. Überlegen Sie, ob dies gerechtfertigt ist und/oder ob der Algorithmus angepasst werden sollte, und falls ja, wie.

Lösung zur Aufgabe 1 Bias im Algorithmus

Ja, dieser Algorithmus enthält einen Bias. Basierend auf unseren Erwartungen haben wir Annahmen getroffen, die unbewusst und ohne Absicht eine gewisse Personengruppe benachteiligt.

Lösung zur Aufgabe 2 Bias im Algorithmus

Felix erhält insgesamt 6 Punkte, Anna hingegen nur 4 Punkte. Obwohl Anna eine gute Maturanote hat, hat sie neben ihren familiären Pflichten nicht viel Zeit oder auch nicht das Geld um sich gesellschaftlich in Vereinen zu engagieren und/oder Musikinstrumente zu spielen.

Referenzen

- Angwin, J., Larson, J., Mattu, S. & Kirchner, L. (2016) Machine Bias. ProPublica. Verfügbar unter <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing> (letzter Zugriff 11. Juni 2020)
- Balkow C., Eckardt I (2019) Denkimpuls Digitale Ethik: Bias in algorithmischen Systemen – Erläuterungen, Beispiele und Thesen. Initiative D21 e. V. Verfügbar unter Denkimpuls Digitale Ethik: Bias in algorithmischen Systemen (letzter Zugriff 11. Juni 2020)
- Bock L., Welle B. (2014). You don't know what you don't know: How our unconscious minds undermine the workplace. Official Blog Google. Verfügbar unter <https://googleblog.blogspot.com/2014/09/you-dont-know-what-you-dont-know-how.html> (letzter Zugriff 11. Juni 2020)
- Chalmers J & Watts T. (2014). Kids should code: why ‘computational thinking’ needs to be taught in schools. The Guardian. Abgerufen von <https://www.theguardian.com/commentisfree/2014/dec/19/kids-should-code-why-computational-thinking-needs-to-be-taught-in-schools> (letzter Zugriff am 9. Juni 2020)
- Curzon, P., & McOwan, P. W. (2018). Computational Thinking: Die Welt des algorithmischen Denkens – in Spielen, Zaubertricks und Rätseln. Springer-Verlag.
- Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. Unpublished manuscript in progress, referenced in <http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf>.
- Dvorsky G. (2014). The 10 Algorithms that dominate our world. Gizmodo. Abgerufen von <https://io9.gizmodo.com/the-10-algorithms-that-dominate-our-world-1580110464> (letzter Zugriff am 09. Juni 2020)
- Lustenberger B. (2018). Ein einfacher Sortieralgorithmus, Leitprogrammatische Unterrichtsunterlagen von Bruno Lustenberger.
- Papert, S. (1996). An exploration in the space of mathematics educations. IJ Computers for Math. Learning, 1(1), 95-123.
- Tan J. (2015). Computational thinking an essential skill for next-generation. Today Online. Abgerufen von <https://www.todayonline.com>

/singapore/computational-thinking-essential-skill-next-generation
(letzter Zugriff am 9. Juni 2020)

- Schwendimann B. A. (2019). Computational Thinking – where do Swiss school stand today?. Digital Switzerland. Abgerufen von <https://digitalswitzerland.com/2019/06/12/computational-thinking-where-do-swiss-schools-stand-today/> (letzter Zugriff am 9. Juni 2020)
- Social Psychology Network. Implicit Association Test. Verfubar unter <http://www.understandingprejudice.org/iat/> (letzter Zugriff am 11. Juni 2020)
- Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33-35.

© Digital Society Initiative

2.7 Folien



**Universität
Zürich**
UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 8: Computational Thinking (CT)

Lektion 1: Einführung zu Computational Thinking

Prof. Thomas Fritz, Institut für Informatik

Lernziele

Am Ende dieser Vorlesung wissen Sie ...

- wie Algorithmen das tägliche Leben beeinflussen,
- was Computational Thinking (CT) bedeutet,
- wieso CT eine wichtige Fähigkeit ist.

Algorithmen verändern die Welt

Algorithmen sind praktisch überall und nicht wegzudenken

Einflussreiche Algorithmen:

- Google Suche 
- Facebook Newsfeed 
- Produkt-Empfehlungen  



Computational Thinking



Reales Problem



Computational
Thinking



Algorithmus & Lösung

Computational Thinking – Definition

«Fähigkeit, ein Problem in kleinere Probleme zu zerlegen, die deterministisch gelöst werden können.» – Wing (2006) & Papert (1996)

«Die **Denkprozesse**, die bei der Formulierung von **Problemen** und deren **Lösungen** ablaufen, so dass die Lösungen in einer Form dargestellt werden, die von einem informationsverarbeitenden Agenten effektiv durchgeführt werden kann.» – Cuny, Snyder, Wing (2011)

Computational Thinking – Beispiel



ZVV Route zur UZH

1. Finde nahegelegene ZVV Haltestellen
2. Schaue nach welche Linien Richtung UZH fahren
3. Fährt eine direkt zur UZH?
4.



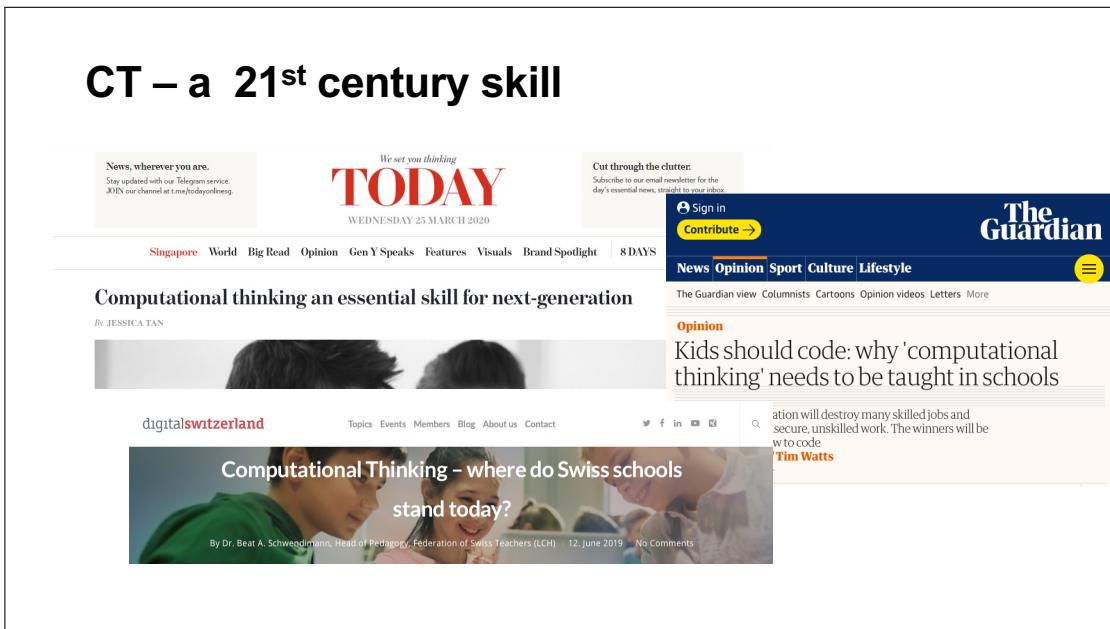
**Algorithmen sind
überall, nicht nur
in Software**

«**Computational Thinking** ist eine Art und Weise, wie Menschen Probleme lösen; es ist nicht der Versuch, Menschen dazu zu bringen, wie Computer zu denken. Computer sind stumpfsinnig und langweilig; Menschen sind klug und einfallsreich. Wir Menschen machen Computer spannend. »

- Wing (2006)

Was ist CT und was ist es nicht?

- Konzeptualisierung **nicht** Programmierung
- Fokus auf Ideen & Prozessen, **nicht** Software & Hardware
- Für jeden, **nicht** nur für Informatiker*innen



Übersicht CT Baustein

- Einführung & Definition
- Hauptbestandteile von CT
- Fall-Beispiele
- Neutralität von Algorithmen

Referenzen

- Chalmers J & Watts T. (2014). Kids should code: why 'computational thinking' needs to be taught in schools. *The Guardian*. Abgerufen von <https://www.theguardian.com/commentisfree/2014/dec/19/kids-should-code-why-computational-thinking-needs-to-be-taught-in-schools> (letzter Zugriff am 9. Juni 2020)
- Cuny, J., Snyder, L., & Wing, J. M. (2010). Demystifying computational thinking for non-computer scientists. *Unpublished manuscript in progress, referenced in http://www.cs.cmu.edu/~CompThink/resources/TheLinkWing.pdf*.
- Dvorsky G. (2014). The 10 Algorithms that dominate our world. *Gizmodo*. Abgerufen von <https://io9.gizmodo.com/the-10-algorithms-that-dominate-our-world-1580110464> (letzter Zugriff am 09. Juni 2020)
- Papert, S. (1996). An exploration in the space of mathematics educations. *IJ Computers for Math. Learning*, 1(1), 95-123.
- Tan J. (2015). Computational thinking an essential skill for next-generation. *Today Online*. Abgerufen von <https://www.todayonline.com/singapore/computational-thinking-essential-skill-next-generation> (letzter Zugriff am 9. Juni 2020)
- Schwendimann B. A. (2019). Computational Thinking – where do Swiss school stand today?. *Digital Switzerland*. Abgerufen von <https://digitalswitzerland.com/2019/06/12/computational-thinking-where-do-swiss-schools-stand-today/> (letzter Zugriff am 9. Juni 2020)
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 8: Computational Thinking

Lektion 2: Hauptbestandteile von Computational Thinking – Teil 1

Prof. Thomas Fritz, Institut für Informatik

Lernziele

Am Ende dieser Vorlesung können Sie ...

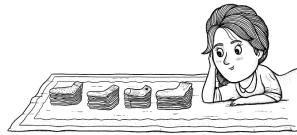
- erklären was die vier Hauptbestandteile von CT sind,
- Mustererkennung, Dekomposition und Abstraktion auf ein gegebenes Problem anwenden.

Beispiel aus dem Alltag – Wäsche zusammenlegen



CT

- Sortieren
 - Farbe
 - Grösse
- Zusammenlegen

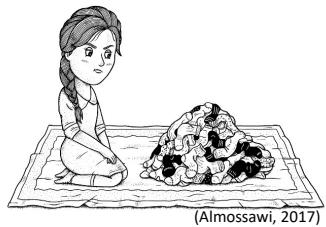


(Almossawi, 2017)



(Almossawi, 2017)

Dekomposition: sortieren, zusammenlegen



(Almossawi, 2017)

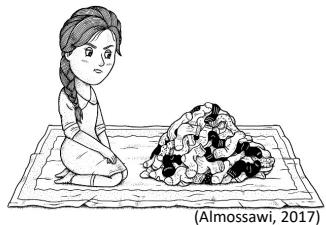
Dekomposition
Mustererkennung



(Almossawi, 2017)

Dekomposition
Mustererkennung
Abstraktion
Fokussieren auf das
Wesentliche / Wichtigste





(Almossawi, 2017)

Dekomposition

Mustererkennung

Abstraktion

Algorithmus: Ablauf festlegen

Spielt eine grosse Rolle

Hauptkomponenten von Computational Thinking

- Dekomposition
- Mustererkennung
- Abstraktion
- *Algorithmus*

Methoden der Dekomposition

- Zielreduktion (z.B. Problem halbieren)
- Rekursive Zielreduktion: wiederholte Anwendung der Problemzerlegung



Beispiel: Dekomposition

Schreiben Sie einen Aufsatz über ein selbstgewähltes Thema im Bereich Deep Learning

Teilprobleme / Schritte:

- Suche und Festlegung auf Thema
- Tiefere Recherche zum Thema
- Strukturieren/Aufteilen des Aufsatzes: Einleitung, Hauptteil, etc.
- Korrekturlesen
- ...



Dekomposition

Erstellung einer Smartphone Applikation “Buch2Go” für den Handel von secondhand Büchern

Aufgabe: Nehmen Sie Sich ein paar Minuten Zeit und überlegen Sie, wie Sie vorgehen würden, um dieses komplexe Problem zu lösen, und in welche Teilprobleme man es zerlegen kann.



Dekomposition

Erstellung einer Smartphone Applikation “Buch2Go” für den Handel von secondhand Büchern

- ⌚ Budget?
- ⚙️ Funktionalitäten?
Erfassen, Suchen, Kaufen, ...
- 👤 Zielpublikum?
- taboola Marketing?
- 📅 Zeitplan?

Hauptkomponenten von Computational Thinking

- Dekomposition
- **Mustererkennung**
- Abstraktion
- *Algorithmus*

Mustererkennung

Ähnlichkeiten & Gemeinsamkeiten zwischen und / oder innerhalb von Problemen identifizieren

- Vereinfachung der Problemlösung durch Wiederverwendung der gleichen Lösung



Beispiel: Mustererkennung

Entwerfe Design von zwei Küchen

Gemeinsame Muster

- Herd
- Kühlschrank
- Waschbecken
- Arbeitsfläche
- ...



Mustererkennung



“Buch2Go” Funktionalität: Erfassen von verschiedenen Büchern die zum Verkauf oder Tausch angeboten werden

Aufgabe:

Nehmen Sie Sich ein paar Minuten Zeit und überlegen Sie ob es bei dieser Funktionalität gewisse Muster / Ähnlichkeiten / Gemeinsamkeiten gibt.



Mustererkennung

“Buch2Go” Funktionalität: Erfassen von verschiedenen Büchern die zum Verkauf oder Tausch angeboten werden

Ablauf Erfassung eines Buches für alle Benutzer und alle Bücher gleich:

- Titel, ISBN, Zustand des Buches und Beschreibung angeben
- Preis festlegen
- Optional Foto hochladen



Mustererkennung

Buch2Go-App: **zusätzlich zur Smartphone App wird eine Web App erstellt**

Aufgabe: Gibt es Gemeinsamkeiten zwischen den zwei Problemen: Smartphone App und Web App?



Mustererkennung

Buch2Go-App: zusätzlich zur Smartphone App wird eine Web App erstellt

Gemeinsame Teilprobleme: Funktionalität, Marketing, Zielgruppe, etc.
→ Wiederverwendung von Lösungen



The diagram illustrates the reuse of solutions. It shows two identical systems side-by-side. Each system consists of a smartphone icon containing a gear icon, which is connected by double-headed arrows to a central 'ADS' box, a calendar icon, a hand holding a dollar sign, and another gear icon. This visualizes how common functional components like user interface elements and data storage can be reused across different platforms (Smartphone and Web App).

Hauptkomponenten von Computational Thinking

- Dekomposition
- Mustererkennung
- **Abstraktion**
- *Algorithmus*

Abstraktion & Datenrepräsentation

Einteilung in wichtige und unwichtige Eigenschaften des Problems;
Fokus auf relevante Aspekte

Arten

- Abstraktion der Steuerung
- Datenabstraktion



Datenabstraktion

Buch2Go-App – Buch



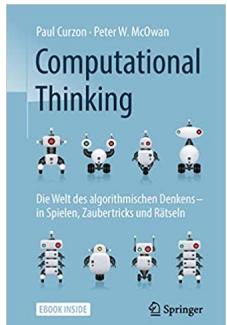
Aufgabe: Welche Eigenschaften eines Buches sind relevant bei der Erfassung in der App?



Datenabstraktion

Buch2Go-App – Buch





Paul Curzon · Peter W. McOwan
Computational Thinking
Die Welt des algorithmischen Denkens –
in Spielen, Zaubertricks und Rätseln
EBOOK INSIDE Springer

Produktinformation

Taschenbuch: 243 Seiten
Verlag: Springer; Auflage: 1. Aufl. 2018 (25. Juli 2018)
Sprache: Deutsch
ISBN-10: 3662567733
ISBN-13: 978-3662567739
Größe und/oder Gewicht: 19,3 x 1,3 x 22,9 cm



Datenabstraktion

Buch2Go-App – Buch



Relevante Eigenschaften

- ✓ Autor*in
- ✓ Titel
- ✓ ISBN
- ...

Irrelevante Eigenschaften

- ✗ Farbe des Covers
- ✗ Geburtsort Autor*in
- ✗ Schriftgrösse
- ...

Hauptkomponenten von Computational Thinking

- Dekomposition
- Mustererkennung
- Abstraktion
- *Algorithmus*

Referenzen

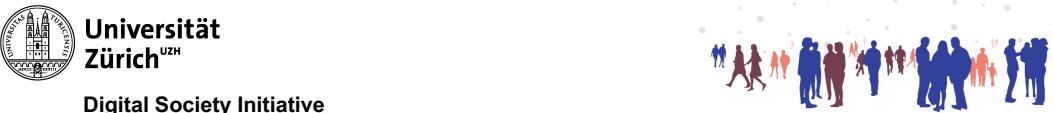
Almossawi, A. (2017). *Bad Choices: How Algorithms Can Help You Think Smarter and Live Happier*. Penguin.

Curzon, P., & McOwan, P. W. (2018). *Computational Thinking: Die Welt des algorithmischen Denkens – in Spielen, Zaubertricks und Rätseln*. Springer-Verlag.

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative



**Universität
Zürich^{UZH}**
Digital Society Initiative

Studium Digitale

Kursbaustein 8: Computational Thinking

Lektion 2: Hauptbestandteile von Computational Thinking – Teil 2

Prof. Thomas Fritz, Institut für Informatik

Lernziele

Am Ende dieser Vorlesung können Sie ...

- den Algorithmus-begriff erklären,
- Algorithmen für ein Problem entwerfen und in Pseudocode erfassen,
- Algorithmen nachvollziehen,
- die Korrektheit von Algorithmen bewerten.

Pfannkuchen Rezept

Zutaten für 4 Portionen

400 g Mehl

750 ml Milch

...

Zubereitung (1h30)

1. Trenne Eiweiss vom Eigelb
2. Mehl, Milch, Eigelb und Prise Salz verrühren

...

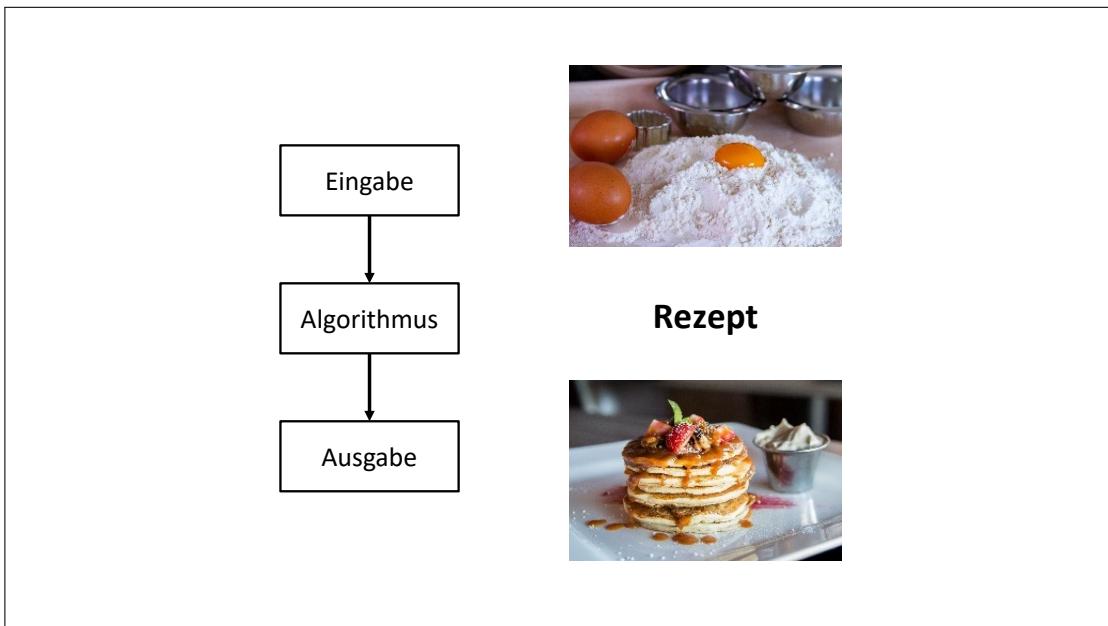


Algorithmus – Definitionen

«Ein Algorithmus ist eine endliche Folge wohldefinierter Anweisungen, typischerweise zur Lösung einer Klasse von Problemen.»

«[...]. Algorithmen bestehen aus endlich vielen, wohldefinierten Einzelschritten. [...] Bei der Problemlösung wird eine bestimmte Eingabe in eine bestimmte Ausgabe überführt.»

– Leiserson et al. (2010)



Beispiel: Algorithmus

Erstelle einen Algorithmus für den Kauf eines Zug-Tickets von Zürich nach Bern in der SBB-App.

SBB Ticket

1. Auf Zeitplan tippen
2. Felder «Von», «Nach», «Datum» ausfüllen
3. Zugverbindung auswählen
4. Ticket kaufen

Kriterien von Algorithmen

- Maschinentauglichkeit
- Allgemeinheit
- Korrektheit

Können Algorithmen jedes Problem lösen?

Nein, Halte-Problem ist ein ungelöstes Problem



Algorithmus



Kauf eines Buches in “Buch2Go”

Aufgabe: Nehmen Sie Sich ein paar Minuten Zeit und erstellen Sie einen Algorithmus für den Kauf eines Buches in der Buch2Go-App. Nehmen Sie an, der Benutzer ist bereits registriert.



Algorithmus



Kauf eines Buches in “Buch2Go”

1. User meldet sich an
2. Eingabe des Titels im Suchtext und suchen drücken
3. Buch auswählen und in den Warenkorb legen
4. Kauf abschliessen



Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?

100 Kürbisse liegen nebeneinander

1. Fange von ganz links (1. Kürbis) an
 - a) Vergleiche den aktuellen Kürbis mit seinem rechten Nachbarn
 - b) Ist der Kürbis grösser als sein Nachbar, dann vertausche sie
 - c) Gehe zum rechten der zwei Kürbisse und wiederhole Schritte a)
bis c)
2. Wiederhole den 1. Schritt 99 mal



Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?



4 Kürbisse als Beispiel

1. Fange von ganz links (1. Kürbis) an



Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?



1. Iteration

- Vergleiche den **aktuellen** Kürbis mit seinem **rechten** Nachbarn
- Ist der Kürbis grösser als sein Nachbar, dann vertausche sie



Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?



1. Iteration

- Vergleiche den aktuellen Kürbis mit seinem rechten Nachbarn
- Ist der Kürbis grösser als sein Nachbar, dann vertausche sie

Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?

1. Iteration

c) Gehe zum rechten der zwei Kürbisse (blau) und wiederhole Schritte a) bis c)

Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?

1. Iteration

linker Kürbis > rechter Kürbis? Nein
→ nicht tauschen, gehe an die Position des rechten Kürbis (rot)

Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?

1. Iteration

The diagram shows four pumpkins in a row. The first two are highlighted with blue boxes, and the last two with orange boxes. A black arrow points from the blue box to the orange box, indicating a swap between the first and second pumpkins. Below the row, the pumpkins are shown again, with the first two swapped positions highlighted by their original colors.

linker Kürbis > rechter Kürbis? Ja → vertausche sie

Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?

1. Iteration

The diagram shows the state of the four pumpkins after the first iteration. The first two pumpkins are now highlighted with orange boxes, and the last two with blue boxes. A black arrow points from the orange box to the blue box, indicating a swap between the second and third pumpkins. Below the row, the pumpkins are shown again, with the second and third pumpkins swapped positions highlighted by their original colors.

Ende erster Iteration.
Starte die nächste Iteration wieder beim 1. Kürbis ganz links ...



Algorithmus

Aufgabe:
Was bewirkt/beschreibt der folgende Algorithmus?

Pseudocode Bubble Sort

```
BubbleSort(Array array)
    for i=0 to array.length - 2 {
        for j=0 to array.length - 2 {
            if array[j] > array[j+1]
                swap(array[j], array[j+1])
        }
    }
```

Referenzen

Curzon, P., & McOwan, P. W. (2018). *Computational Thinking: Die Welt des algorithmischen Denkens – in Spielen, Zaubertricks und Rätseln*. Springer-Verlag.

Leiserson, C. E., Rivest, R., & Stein, C. (2010). *Algorithmen-Eine Einführung*. Oldenbourg Verlag, München

Lustenberger Bruno, Ein einfacher Sortieralgorithmus, Leitprogrammatische Unterrichtsunterlagen von Bruno Lustenberger, EDUCETH Das Bildungsportal der ETH Zürich, verfügbar unter [https://ethz.ch/content/dam/ethz/special-interest/dual/educeth-dam/documents/Unterrichtsmaterialien/informatik/Einfacher%20Sortieralgorithmus%20\(Leitprogrammatische%20Unterrichtsunterlagen\)/Lustenberger_SlectionSort_jun_08.pdf](https://ethz.ch/content/dam/ethz/special-interest/dual/educeth-dam/documents/Unterrichtsmaterialien/informatik/Einfacher%20Sortieralgorithmus%20(Leitprogrammatische%20Unterrichtsunterlagen)/Lustenberger_SlectionSort_jun_08.pdf) (letzter Zugriff am 9. Juni 2020)

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative

Lernziele

Am Ende dieser Vorlesung ...

- kennen Sie sieben weitere Fertigkeiten für gutes CT,
- können Sie die Fertigkeiten an einem Beispiel erklären / anwenden.

Beispiel aus dem Alltag

Im Auto durch die Stadt navigieren



↓ CT



CT-Fertigkeiten

- **Modellierung**



CT-Fertigkeiten

- Modellierung
- **Wissenschaftliches Denken**



CT-Fertigkeiten

- Modellierung
- Wissenschaftliches Denken
- **Logisches Denken**



CT-Fertigkeiten

- Modellierung
- Wissenschaftliches Denken
- Logisches Denken
- **Kreativität**



CT-Fertigkeiten

- Modellierung
- Wissenschaftliches Denken
- Logisches Denken
- Kreativität
- **Menschenkenntnis**



Beispiel: Menschenkenntnis

Algorithmus zum sicheren Flussüberqueren

Beispiel aus Curzon & McOwan (2018)

Eine Bäuerin ist mit ihrem Schäferhund Maxi, der sie immer begleitet, auf dem Weg zum nächsten Dorf. Um dorthin zu gelangen, muss sie einen reisenden Fluss überqueren. Dafür hat eine Erfinderin, die auf der Dorfseite des Flusses wohnt, eine Vorrichtung erfunden, die die Flussüberquerung ermöglicht. Sie besteht aus einem Seil, Rollen und einem Sitz, der an einem Seil hängt und Platz für eine Person bietet. Die Bewohner des Ortes haben vereinbart, den Sitz immer auf der Seite des Dorfes zu lassen, wo die Erfinderin wohnt.



Beispiel: Menschenkenntnis

Algorithmus zum sicheren Flussüberqueren

Beispiel aus Curzon & McOwan (2018)

Als die Bäuerin zum Fluss kommt, zieht sie den Sitz mit dem Seil zu sich herüber, setzt sich hinein, nimmt Maxi auf den Arm, zieht sich selbst auf die andere Seite und setzt ihren Weg zum Dorf fort. Eines Tages kauft sie eine Henne und einen Sack Reis. Als sie auf dem Heimweg zum Fluss kommt, stellt sie fest, dass sie nur ein Ding mit sich hinübernehmen kann, wenn sie den Sitz verwenden möchte. Deshalb benötigt sie mehrere Fahrten. Dabei gibt es folgendes Problem:



Beispiel: Menschenkenntnis

Algorithmus zum sicheren Flussüberqueren

Beispiel aus Curzon & McOwan (2018)

Wenn sie die Henne mit dem Getreide auf einer Seite allein lässt, wird die Henne das Getreide fressen. Lässt sie Hund Maxi und die Henne auf einer Seite zusammen, wird der Hund die Henne jagen. Maxi frisst kein Getreide, deshalb kann er mit dem Getreide allein gelassen werden.

Aufgabe:

Schreiben Sie einen Algorithmus um alle drei Sachen auf die andere Flussseite zu bringen, ohne dass etwas gefressen oder gejagt wird.
Beachten Sie: Sowohl der Hund und die Henne als auch die Henne und das Getreide voneinander getrennt gehalten werden müssen.



Beispiel: Menschenkenntnis

Algorithmus zum sicheren Flussüberqueren

Beispiel aus Curzon & McOwan (2018)

1. Sie fährt mit der Henne auf die andere Seite.
2. Sie kehrt alleine zurück.
3. Sie fährt mit dem Hund auf die andere Seite.
4. Sie kehrt mit der Henne zurück.
5. Sie fährt mit dem Getreide auf die andere Seite.
6. Sie kehrt alleine zurück.
7. Sie fährt mit der Henne auf die andere Seite.



Beispiel: Menschenkenntnis

Algorithmus zum sicheren Flussüberqueren

Beispiel aus Curzon & McOwan (2018)

1. Sie fährt mit der Henne auf die andere Seite.
2. Sie kehrt alleine zurück.
3. Sie fährt mit dem Hund auf die andere Seite.
4. Sie kehrt mit der Henne zurück.
5. Sie fährt mit dem Getreide auf die andere Seite.
6. Sie kehrt alleine zurück.
7. Sie fährt mit der Henne auf die andere Seite.
- 8. Sie schickt den leeren Sitz zum anderen Ufer zurück.**

Menschen verstehen

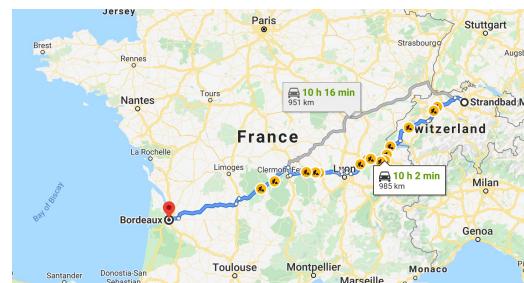
Post-Completion Error

Weiteres Beispiel: Geldautomat



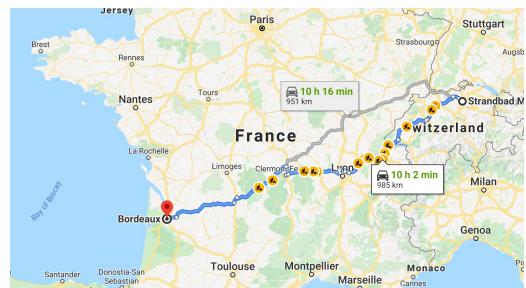
CT-Fertigkeiten

- Modellierung
- Wissenschaftliches Denken
- Logisches Denken
- Kreativität
- **Menschenkenntnis**



CT-Fertigkeiten

- Modellierung
- Wissenschaftliches Denken
- Logisches Denken
- Kreativität
- Menschenkenntnis
- **Heuristiken**



Weitere Aspekte von CT

- Computermodellierung
- Wissenschaftliches Denken
- Logisches Denken
- Kreativität
- Menschenkenntnis
- Heuristiken
- **Evaluation**

Evaluation

Überprüfung auf

- Funktionale Richtigkeit
- Zweckerfüllung
- Leistungsfähigkeit
- Benutzerfreundlichkeit/Bedienbarkeit
- ...

Nicht nur am Ende, sondern kontinuierlich



Dekomposition
Mustererkennung
Abstraktion
Algorithmus

Modellierung
Wissenschaftliches Denken
Logisches Denken
Kreativität
Menschenkenntnis
Heuristiken
Evaluation

Referenzen

Curzon, P., & McOwan, P. W. (2018). *Computational Thinking: Die Welt des algorithmischen Denkens – in Spielen, Zaubertricks und Rätseln.* Springer-Verlag.

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative

Lernziele

Am Ende dieser Vorlesung

- Sind Sie in der Lage Computational Thinking auf ein Problem anzuwenden und einen Algorithmus zu erstellen der das Problem löst.



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Jährlich erhält eine Stiftung zur Förderung eines Studiums mehrere hundert Bewerbungen, von denen sie 20 mit einem Stipendium fördert.

Aufgabe (ca. 20 Minuten):

Entwerfen Sie einen Algorithmus, der ein Rangliste der Bewerber*innen und derer CVs basierend auf den nachfolgenden Kriterien erstellt, und die besten 20 auswählt.



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Förderungskriterien:

- Notenschnitt von mindestens 5.3 (Matura, Berufsmatura)
- Alter von höchstens 28 Jahren
- Kenntnisse von mindestens zwei Amtssprachen
- Gesellschaftliches Engagement, breite Interessen, intellektuelle Neugier und Kreativität



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Tipp:

Um eine Rangliste zu erstellen, benutzen Sie ein Punktesystem, dass Bewerber*innen basierend auf gewissen Charakteristiken bewertet. Zum Beispiel könnte es pro gespieltem Musikinstrument +1 Punkt geben, usw..



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Dekomposition

- Überprüfe Förderungsfähigkeit der Bewerber*innen:
 - Alter <= 28 Jahre; Kenntnisse >= 2 Amtssprachen; Notenschnitt >= 5.3
- Bewerte Bewerber*in (CVs) basierend auf gesellschaftlichem Engagement, breiten Interessen, intellektueller Neugier und Kreativität
- Erstelle Rangliste und wähle Top 20 aus



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Abstraktion & Muster

- Bewerbungen werden alle nach gleichem Schema/Ablauf bearbeitet
 - Überprüfung der Förderungsfähigkeit ist gleich für alle Bewerber*in
 - Bewertung sollte auch gleich ablaufen für alle Bewerber*in
- Fokus auf relevante Charakteristika: Alter, Sprachkenntnisse, Note, Anzahl gesellschaftlicher Engagements, Musikinstrumente,...



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Algorithmus

1. Für jeden Bewerber*in
 1. Prüfe ob Bewerber*in förderfähig ist: Alter <= 28 Jahre && #Sprachen >= 2 Amtssprachen && Notenschnitt >= 5.3
 2. Bewerte CV / Berechne Punktezahl



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

1.2 Bewerte CV / Berechne Punktezahl

1. +1 Punkt pro gesellschaftlichen Engagement (sozialer Verein, etc.)
2. +1 Punkt pro Musikinstrument
3. +1 Punkt pro extracurricularem Projekt
4. + ((Notenschnitt – 5.3) * 10) Punkte // +1 pro 0.1 im Notenschnitt über 5.3
5. + (28 – Alter) Punkte // je schneller zum Studium desto besser



Begabtenförderung



Erstellen eines Ranglisten & Auswahl Algorithmus

Algorithmus

1. Für jeden Bewerber*in
 1. Prüfe ob Bewerber*in förderfähig ist: Alter <= 28 Jahre && #Sprachen >= 2 Amtssprachen && Notenschnitt >= 5.3
 2. Bewerte CV / Berechne Punktezahl
 3. Erstelle Rangliste basierend auf Bewertungen für jeden Bewerber*in
2. Wähle die Top 20 der Rangliste aus



Meetingplanung



Erstelle Algorithmus zur zeitlichen Planung von Meetings

Für ein Seminar gibt es drei Gruppen die mit dem Teaching Assistant ein wöchentliches Meeting am Donnerstag vereinbaren sollen. Jede Gruppe schreibt dem TA eine E-Mail mit drei passenden Zeitfenstern mit Priorisierung.

Aufgabe (ca. 20 Minuten):

Entwerfe einen Algorithmus, der die Meetings für den TA plant so dass es keine Konflikte gibt.



Meetingplanung

Erstelle Algorithmus zur zeitlichen Planung von Meetings

Icon: Calendar with two people.

Beispiel Daten für TA:

- **G1:** 09-10(1), 10-11(2), 11-12 (3)
- **G2:** 10-11(1), 11-12 (2), 13-14 (3)
- **G3:** 09-10(1), 11-12 (2), 16-17 (3)



Meetingplanung

Erstelle Algorithmus zur zeitlichen Planung von Meetings

Icon: Calendar with two people.

Algorithmus

1. Gehe durch alle Gruppen nach Zeitpunkt der Email (first come first serve)
 1. Für aktuelle Gruppe, gehe durch die 3 Zeitfenster nach Priorisierung
 1. Ist das aktuelle Zeitfenster im TA Kalender frei? [Schritt 1.1.1]
Ja: weise Zeitfenster zu und gehe zur nächsten Gruppe in Schritt 1
Nein: gehe zum nächsten Zeitfenster und wiederhole Schritt 1.1.1
 2. Sende zugewiesene Zeitfenster an Studenten



Meetingplanung

Erstelle Algorithmus zur zeitlichen Planung von Meetings



- Ist der Algorithmus TA- oder Studenten-freundlich?
- Was passiert wenn es mehr als 3 Gruppen sind?

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative

Lernziele

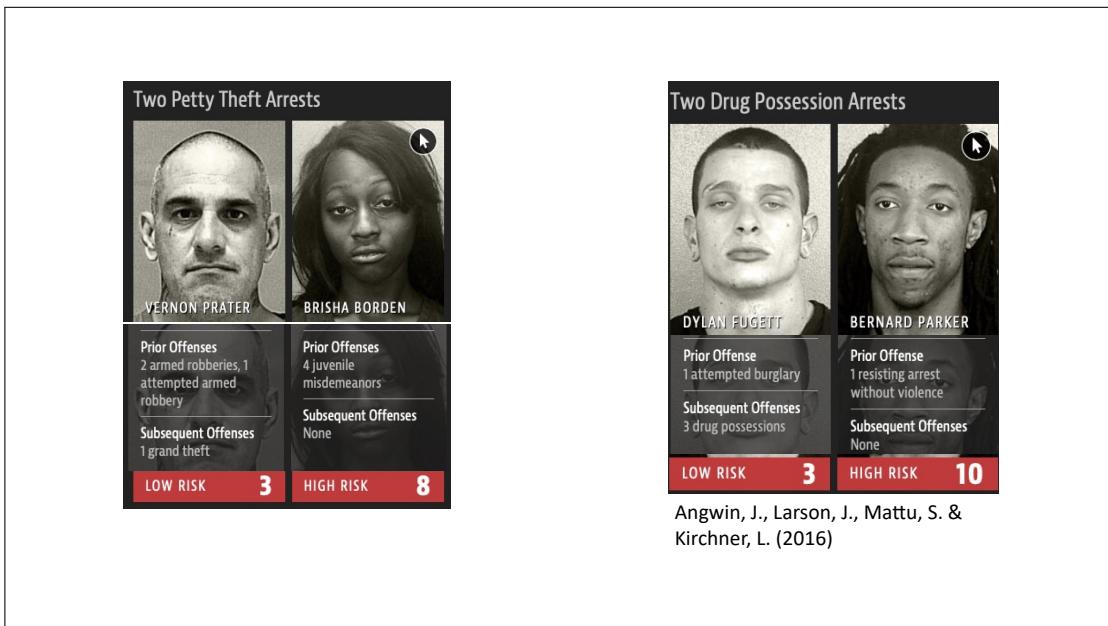
Am Ende dieser Vorlesung ...

- wissen Sie was algorithmischer Bias ist.
- Können Sie gewissen Bias in Algorithmen erkennen.

Beispiel: COMPAS Software

«*There's software used across the country to predict future criminals. And it's biased against blacks.*»

-- Angwin, J., Larson, J., Mattu, S. & Kirchner, L. (2016) *Machine Bias*.



Bias

- Vorurteile
- Verzerrungen
- Vernachlässigung gesellschaftlicher Gruppen

Arten von Bias

Bewusster Bias

- Person weiss, dass sie z.B. Vorurteile hat, beeinflusst ist, etc.

Unbewusster Bias

- Bias ist Person selber nicht bewusst, z.B. Name oder Affinity Bias

Implicit Association Test <http://www.understandingprejudice.org/iat/>

Algorithmischer Bias – Definition

«Algorithmischer Bias beschreibt systematische und wiederholbare Fehler in einem Computersystem, die unfaire Ergebnisse erzeugen, wie zum Beispiel die Privilegierung einer willkürlichen Gruppe von Benutzern gegenüber anderen.»

Gründe für algorithmischer Bias

- Algorithmen wider-spiegeln impliziten Werte der Menschen, die beim Erstellen beteiligt sind
- Daten auf denen Algorithmus basiert (z. B. im Recruiting)



Begabtenförderung & Bias

Erstellen eines Ranglisten & Auswahl Algorithmus

Recap

Entwerfe einen Algorithmus, der ein Rangliste der Bewerber*innen (CVs) basierend auf Kriterien erstellt, und die besten 20 auswählt.

Kriterien:

Notenschnitt, Alter, Sprachkenntnisse, gesellschaftliches Engagement, breite Interessen, intellektuelle Neugier und Kreativität



Begabtenförderung & Bias

Erstellen eines Ranglisten & Auswahl Algorithmus

Aufgabe

- Wäre es möglich dass ein Bias im Algorithmus vorliegt?
Falls ja, wo?



Algorithmischer Bias

Erstellen eines Ranglisten & Auswahl Algorithmus

2 Kandidaten, wer wird besser bewertet und ist dies fair?

<p>Anna K.</p> <ul style="list-style-type: none">- 24 J, mexikanische Staatsbürgerschaft, seit mehreren Jahren wohnhaft in der Schweiz- 5.7 (Matura für Erwachsene)- Alleinerziehende Mutter, Unterstützung der Familie in Mexiko	<p>Felix M.</p> <ul style="list-style-type: none">- 20 J, schweizer Staatsbürgerschaft, wohnhaft in CH- 5.4 (Kantonsschule)- Mitglied im Orchester&Schachclub- Spielt E-Gitarre, Kontrabass&Violine- Freiwilliger Helfer bei Pontesano
---	--



Algorithmischer Bias

Erstellen eines Ranglisten & Auswahl Algorithmus



Algorithmus

1. Für jeden Bewerber*in
 1. Prüfe ob Bewerber*in förderfähig ist: Alter <= 28 Jahre && #Sprachen >= 2 Amtssprachen && Notenschnitt >= 5.3
 2. Bewerte CV / Berechne Punktezahl
 1. +1 Punkt pro gesellschaftlichen Engagement (sozialer Verein, etc.)
 2. +1 Punkt pro Musikinstrument
 3. +1 Punkt pro extracurricularem Projekt
 4. + ((Notenschnitt – 5.3) * 10) Punkte // +1 pro 0.1 im Notenschnitt über 5.3
 5. + (28 – Alter) Punkte // je schneller zum Studium desto besser
 2. Erstelle Rangliste basierend auf Bewertungen für jeden Bewerber*in
 3. Wähle die Top 20 der Rangliste aus



Algorithmischer Bias

Erstellen eines Ranglisten & Auswahl Algorithmus

2 Kandidaten, wer wird besser bewertet und ist dies fair?

Anna K.	Felix M.
<ul style="list-style-type: none"> - 24 J, mexikanische Staatsbürgerschaft, seit mehreren Jahren wohnhaft in der Schweiz - 5.7 (Matura für Erwachsene) - Alleinerziehende Mutter, Unterstützung der Familie in Mexiko 	<ul style="list-style-type: none"> - 20 J, schweizer Staatbürgerschaft, wohnhaft in CH - 5.4 (Kantonsschule) - Mitglied im Orchester&Schachclub - Spielt E-Gitarre,Kontrabass&Violine - Freiwilliger Helfer bei Pontesano

Algorithmen verändern die Welt

Computational Thinking
kreieren und verstehen von
Algorithmen

FACEBOOK **amazon**
NETFLIX **Google**



Referenzen und zum Weiterlesen

Angwin, J., Larson, J., Mattu, S. & Kirchner, L. (2016) Machine Bias. *ProPublica*. Verfügbar unter <https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing> (letzter Zugriff 11. Juni 2020)

Balkow C., Eckardt I (2019) Denkimpuls Digitale Ethik: Bias in algorithmischen Systemen – Erläuterungen, Beispiele und Thesen. *Initiative D21 e. V.* Verfügbar unter Denkimpuls Digitale Ethik: Bias in algorithmischen Systemen (letzter Zugriff 11. Juni 2020)

Social Psychology Network. Implicit Association Test. Verfügbar unter <http://www.understandingprejudice.org/iat/> (letzter Zugriff am 11. Juni 2020)

Bock L., Welle B. (2014). You don't know what you don't know: How our unconscious minds undermine the workplace. *Official Blog Google*. Verfügbar unter <https://googleblog.blogspot.com/2014/09/you-dont-know-what-you-dont-know-how.html> (letzter Zugriff 11. Juni 2020)

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative

3 Programmierung

PROF DR. THOMAS FRITZ UND CLAUDIA VOGEL

Studium Digitale Kursbaustein 9 Begleit-Skript

Dieses Skript bereitet die Inhalte des Kursbaustein Einführung in Programmierung schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

3.1 Einführung in die Programmierung

Software ist heutzutage überall anzutreffen, und bestimmt in vielen Bereichen auch Teile unseres Lebens. Die Kosten für die Software eines Autos sind heute oft teurer als der Motor des Autos, und wer benutzt heute noch keine Email oder Messaging Applikation um mit Freunden zu kommunizieren. Im Kursbaustein zu Computational Thinking haben wir schon erwähnt dass Algorithmen überall sind, aber um etwas zu bewirken, müssen wir Algorithmen in Software / Programme überführen die von Computern ausgeführt werden können.

Beim Programmieren geht es darum Ideen und Lösungen zu einer Abfolge von Anweisungen zu transformieren, die dann vom Computer ausgeführt werden können. Dass heisst, Programme sagen dem Computer was genau gemacht werden soll und sie müssen dazu in einer Sprache geschrieben werden, die vom Computer verstanden wird. Doch Programmierung ist mehr als nur das Schreiben von Programmen in einer Programmiersprache. Zum Programmieren gehört es auch gute Lösungen zu Problemen zu finden. Wurde eine Lösung oder ein Algorithmus gefunden, wird dieser mit einer Programmiersprache formal ausgedrückt, so dass der Computer es ausführen kann.

3.2 Programmiersprachen

Eine Programmiersprache ist eine formale Sprache und dient wie die natürliche Sprache der Kommunikation, in diesem Fall der Kommunikation mit dem Computer. Um Computer zu steuern, ist es notwendig, ihnen klar verständliche Befehle zu geben. Da Computer auf ihrer untersten Ebene auf die Verarbeitung von 0en und 1en beschränkt sind und eine Folgen von 0en und 1en für Menschen schwer verständlich ist, wurden über die Jahre verschiedenste Programmiersprachen entwickelt um das Programmieren zu erleichtern.

Das Artefakt, das durch das Programmieren entsteht nennt sich Quellcode, Quelltext, Programmcode oder auch einfach nur Code.

3.2.1 Drei Generationen der Programmiersprachen (Peroni 2018, Boles 1999)

Programmiersprachen können grob in drei Generationen unterteilt werden:

1. Generation: Maschinensprachen

Maschinensprache ist eine Reihe von Befehlen, die direkt vom Prozessor (CPU: Central Processing Unit) eines Computers ausgeführt werden können, dass heisst, die Programmiersprach wurde sehr stark von der Hardware beeinflusst. Die Programme in Maschinensprache werden in der Regel binär kodiert, d.h. in einer Sequenz von 0 und 1 ausgedrückt. Diese Sprache ist nur begrenzt von Menschen lesbar und führt zu erheblichen Schwierigkeiten bei der Wartung und Fehlerbeseitigung.

2. Generation: Assemblersprachen

Die 2. Generation von Programmiersprachen hat eine Abstraktionsebene über dem Maschinencode eingeführt, die sogenannten Assemblersprachen. In Assemblersprachen werden die in der Maschinensprache verwendeten Zahlenfolgen durch lesbare Abkürzungen ersetzt. Dadurch kann man Programme schreiben, die für Menschen leichter lesbar und verständlicher sind. Ein sogenannter Assembler übersetzt dann wie ein Dolmetscher das Programm von Assembler-sprache in Maschinensprache. Dadurch wurden die Vorteile der hohen Ausführungs geschwindigkeit und des geringen Speicherbedarfs von Programmen beibehalten.

3. Generation: Höhere Programmiersprachen

Bei der 3. Generation handelt es sich um höhere Programmiersprachen. Höhere Programmiersprachen sind Sprachen die sich durch eine starke Abstraktion von der Maschinensprache auszeichnen und näher an der natürlichen Sprache von Menschen sind. Eine höhere Programmiersprache kann natürlich sprachliche Wörter für spezifische Konstrukte verwenden, um für den Menschen leicht benutzbar und verständlich zu sein. Im Allgemeinen gilt, dass die Programmiersprache umso verständlicher ist, je stärker die Abstraktion von der Maschinensprache gegeben ist. Die Programme, die in einer höheren Programmiersprache geschrieben sind, müssen zunächst in eine maschinenlesbare Form, beispielsweise binär in 0en und 1en, übersetzt werden, bevor sie vom Computer ausgeführt werden können.

3.2.2 Assembler, Compiler & Interpreter

Für die Übersetzung von Code in Maschinensprache wird je nach Sprache ein Assembler, Compiler, Interpreter oder eine Kombination davon verwendet. Assembler werden für die Übersetzung von Assemblersprachen genutzt. Compiler bzw. Interpreter oder eine Kombination aus beiden, werden für höhere Programmiersprachen eingesetzt. Der Hauptunterschied zwischen Compiler und Interpreter liegt darin, dass ein Compiler **vor** der Ausführung eines Programmes den gesamten Quellcode übersetzt, wohingegen der Interpreter **während** der Ausführung die benötigten Teile übersetzt.

Assembler

Ein Assembler transformiert ein Programm in Assemblersprache in ein Programm in Maschinensprache.

Compiler

Ein Compiler wandelt Quellcode aus einer höheren Programmiersprache in Maschinensprache um, er übersetzt also das gesamte Programm von einer Programmiersprache in Maschinensprache. Der Code wird vollständig übersetzt, bevor das Programm vom Computer ausgeführt wird. Deshalb benötigt er vergleichsweise mehr Zeit und Ressourcen als ein Interpreter. Sobald das fertige Programm läuft, kann es oft jedoch effizienter als interpretierte Programme ausgeführt werden, da alle Anweisungen bereits vollständig in Maschinensprache übersetzt wurden. Ein Compiler erzeugt bei vielen Sprachen plattformabhängigen Maschinencode, d. h. der erzeugte Maschinencode kann nur auf bestimmten Betriebssystemen ausgeführt werden. Reine Compiler-Sprachen sind zum Beispiel C und C++.

Interpreter

Ein Interpreter verarbeitet den Code eines Programms zur Laufzeit. Dazu geht der Interpreter Zeile für Zeile vor: Eine Anweisung wird eingelesen, analysiert und sofort ausgeführt. Dann geht es mit der nächsten Anweisung weiter. Ein Interpreter erzeugt keine Datei in Maschinensprache, die man mehrmals ausführen könnte. Da jede Anweisung bei jeder Ausführung des Programms einzeln verarbeitet wird, können interpretierte Programme langsamer als kompilierte Sprachen sein, allerdings gibt es heute auch häufig Kombinationen wie Just-In-Time-Compilation. Python oder Perl sind Beispiele für Programmiersprachen, die einen Interpreter verwenden.

3.2.3 Definition von Programmiersprachen

Programmiersprachen sind sehr exakte, künstliche Sprachen zur Formulierung von Programmen. Sie dürfen keine Mehrdeutigkeiten bei der Programmerstellung zulassen, damit der Computer das Programm auch korrekt ausführen kann. Bei der Definition einer Programmiersprache muss ihre Lexikalk, Syntax und Semantik definiert werden:

- Lexikalk: sie definiert die gültigen Zeichen bzw. Wörter, aus denen Programme der Programmiersprache zusammengesetzt sein dürfen.
- Syntax: sie definiert den korrekten Aufbau der Sätze aus gültigen Zeichen, d. h. sie legt fest, in welcher Reihenfolge lexikalisch korrekte Zeichen im Programm auftreten dürfen.
- Semantik: sie definiert die Bedeutung syntaktisch korrekter Sätze, d.h. sie beschreibt, was passiert, wenn beispielsweise bestimmte Anweisungen ausgeführt werden.

3.2.4 Programmierparadigmen

Bevor ein Programm erstellt wird, wird entschieden mit welcher Programmiersprache es umgesetzt wird. Es gibt viele verschiedene höhere Programmiersprachen, die es erlauben Ideen und Lösungen auf unterschiedliche Art und Weise auszudrücken und zu lösen. Jede weist ihre eigenen Vor- und Nachteile auf. Generell kann man Programmiersprachen nach Paradigmen kategorisieren. Bei einem Programmierparadigma geht es um den grundlegenden Stil, in dem ein Programm geschrieben wird. Es beschreibt welche Prinzipien und Denkmuster angewandt und welche Herangehensweisen genutzt werden. Programmierparadigmen sind beispielsweise die deklarative, imperitative, objektorientierte, funktionale, oder logische Programmierung. Dabei gibt es keine exakte Definitionen der einzelnen Paradigmen und gewisse Programmiersprachen können auch mehreren Paradigmen gleichzeitig zugewiesen werden.

Je nach zu lösendem Problem eignen sich manche Programmierparadigmen besser als andere, denn sie bestimmen wie wir unsere Ideen in Code ausdrücken können. Wenn wir zum Beispiel ein Programm schreiben wollen bei dem wir dem Computer eine genaue Abfolge von Instruktionen geben wollen die er abarbeiten soll, dann benutzen wir oft eine imperative Programmiersprache. Wenn wir dagegen eher die reale Welt ausdrücken wollen, in der es

Objekte gibt die verschieden Verhalten haben, dann benutzen wir eher eine objektorientiert Programmiersprache.

Deklarative Programmiersprachen legen den Fokus auf die Beschreibung des gewünschten Ergebnisses anstatt die Arbeitsschritte aufzulisten. Das *funktionale* Programmierparadigma stammt aus der Mathematik, genauer aus der Theorie der Funktionen. Das zentrale Element dieses Paradigma ist die Funktion, im traditionellen mathematischen Sinne $y=f(x)$. Alle Elemente können als Funktion aufgefasst und der Code kann durch aneinander gereihte Funktionsaufrufe ausgeführt werden. Die *imperative* Programmierung ist dadurch charakterisiert, dass Programmcode aus einer Abfolge von Anweisungen oder auch Instruktionen zur Steuerung besteht. Die Idee ist dass alle Anweisungen Schritt für Schritt nach dem Prinzip «erledige zuerst das, dann das, etc.» abgearbeitet werden, d.h. der Fokus ist hier auf dem «wie» anstelle von dem «was» (deklarativ).

Objektorientierte Programmiersprachen haben in den letzten Jahren viel an Popularität gewonnen. Objektorientierung erlaubt es uns die reale Welt als Objekte zu repräsentieren die miteinander interagieren können und gewisse Eigenschaften haben. Dies ist häufig sehr nah daran, wie Menschen die reale Welt beschreiben.

3.3 Sequenzen und Variablen

Obwohl sich Programmiersprachen in vielen Aspekte unterscheiden können, gibt es gewisse grundlegende Programmierkonstrukte die in sehr vielen Sprachen anzutreffen sind. Zwei der grundlegenden logischen Strukturen in der Programmierung sind Anweisungen und Sequenzen.

3.3.1 Anweisung und Sequenzen

Eine **Anweisung** (Statement, Befehl, usw.) ist eine elementare Einheit im Programm und steht für einen einzelnen Abarbeitungsschritt im Algorithmus. Eine Anweisung wird auch häufig Statement, Kommando oder Befehl genannt. Meistens entspricht eine Anweisung auch einer Programmierzeile im Code aber die Syntax unterscheidet sich je nach Programmiersprache.

Eine Abfolge von Anweisungen, die nacheinander ausgeführt werden, nennt man **Sequenz**. Eine Sequenz kann beliebig viele Anweisungen enthalten, sie werden der Reihe nach von oben nach unten ausgeführt, aber es können keine Anweisungen übersprungen / ausgelassen werden.

3.3.2 Variablen

Während das Programm läuft, müssen oft viele verschiedene Werte (zwischen-)gespeichert werden. Dieses Speichern von einfachen Werten ist mit **Variablen** möglich. Variablen erlauben es gewisse Wert zu speichern und den gespeicherten Wert zu einem späteren Zeitpunkt auszulesen oder zu verändern. Bevor eine Variable verwendet werden kann, muss sie deklariert, d.h. erstellt werden. Die Deklaration definiert den Namen der Variable und je nach Programmiersprache auch deren Datentyp. Wenn eine Variable deklariert wird, merkt sich der Compiler/Interpreter den Namen und reserviert Speicherplatz für diese. Nach oder während der Erstellung kann in dieser Variable ein Wert gespeichert werden. Erstmaliges Speichern/Zuweisen eines Wertes nennt man Initialisierung. Erst nach der Initialisierung hat die Variable einen Wert gespeichert und kann im fortlaufenden Programm verwendet werden. D.h. man kann über den Namen der Variable auf den Wert der Variable zugreifen und diesen auslesen. Die Deklaration und Initialisierung können miteinander oder nacheinander erfolgen. Je nach Datentyp kann der Wert der Variable verändert werden, respektive kann der Variable auch ein neuer Wert zugewiesen werden.

3.3.3 Datentypen

Die zu verarbeitenden Daten, resp. die Variablen, können von unterschiedlichen Datentypen sein. Ein Datentyp beschreibt eine Menge von Datenobjekten, die alle die gleiche Struktur haben und mit denen die gleichen Operationen ausgeführt werden können. Also zum Beispiel kann ich alle Datenobjekte des Datentyps Ganze Zahlen addieren, multiplizieren oder subtrahieren. Beispiele für Datentypen sind Ganze Zahlen, Gleitkommazahlen, Zeichen(-ketten), Listen, Wahrheitswerte. Diese Datentypen können sich je nach Programmiersprache unterscheiden.

3.3.4 Fallunterscheidung

Die Fallunterscheidung ermöglicht es alternative Abläufe zu haben und so sequentielle Abläufe zu steuern. In einer Fallunterscheidung wird eine Frage gestellt, die entweder mit Ja (wahr/true) oder Nein (falsch/false) beantwortet werden kann. Je nach Antwort wählt das Programm eine von zwei Handlungsabläufen. Diese Fragenstruktur braucht einen Eingabewert, eine sogenannte Variable, die überprüft/ausgewertet wird und entweder eine Bedingung erfüllt (true) oder nicht (false).

Es gibt die einseitige und zweiseitige Fallunterscheidung. Bei einer einseitigen Fallunterscheidung werden die angegebenen Anweisungen ausgeführt, falls die Bedingung erfüllt ist. Falls die Bedingung nicht erfüllt ist, geschieht nichts. Bei der zweiseitigen Fallunterscheidung werden für beide Fälle Anweisungen angegeben und je nach Auswertung die Anweisungen des einen oder des anderen Falles ausgeführt.

3.3.5 Schleifen

Schleifen ermöglichen es einen Block von Anweisungen wiederholt auszuführen. Dabei müssen die Anweisungen nur einmal geschrieben werden und die Schleife führt diese Anweisung dann mehrmals aus, wobei spezifiziert werden kann wie oft die Anweisungen wiederholt werden sollen.

In diesem Kurs schauen wir uns zwei verschiedene Schleifentypen an:

- zählerkontrollierte Schleifen, und
- bedingungskontrollierte Schleifen.

Zählerkontrollierte Schleifen

Eine zählerkontrollierte Schleife ist ein Programmkonstrukt bei dem die Anzahl der Wiederholungen (oft spezifiziert in einer Zählervariable) von vornherein feststeht. Die Anweisungen im Schleifenkörper werden dann genau so oft ausgeführt wie durch die Anzahl vorgegeben ist.

Bedingungskontrollierte Schleifen

Bei der bedingungskontrollierten Schleife wird der Block von Anweisungen im Schleifenkörper solange wiederholt ausgeführt bis die angegebene Bedingung wahr ist. Die tatsächliche Anzahl der Iterationen ist bei der bedingungskontrollierten Schleife erst zur Ausführungszeit bekannt und wird nicht beim Programmieren von vornherein definiert.

Endlos-Schleife

Eine Endlosschleife ist eine Schleife, die sich unendlich oft wiederholt. Eine Endlosschleife entsteht, wenn eine Abbruchbedingung bei einer bedingungskontrollierten Schleife fehlerhaft ist und immer als wahr ausgewertet wird. Normalerweise führt eine Endlosschleife zu einem Fehler im Programm und sollte vermieden werden.

3.3.6 Funktionen

Wird ein Stück Code oft wiederholt oder wird es an verschiedenen Stellen eines Programms eingesetzt, kann es in eine Funktion ausgelagert werden. Dann wird jeweils die Funktion aufgerufen wenn der Code der Funktion ausgeführt werden soll. Um die Funktion aufrufen zu können, wird die Funktion mit einem Namen definiert. Der Funktion können gewisse Werte übergeben werden, die man Parameter nennt, beispielsweise wenn die Funktion mehrmals aufgerufen wird, aber unterschiedliche Werte verwenden soll.

Funktionen sind ein wichtiges Konstrukt um «modular» zu programmieren und Probleme in kleinere Teilprobleme zu zerlegen. Für jedes Teilproblem kann eine eigene Funktion programmiert werden und dass gesamte Problem kann dann gelöst werden in dem die Funktionen für die Teilprobleme ausgeführt werden.

3.4 Python

Häufig verwendete höhere Programmiersprachen für Computerprogramme oder Applikationen sind Sprachen wie Python, Java oder JavaScript. Python wird zum Beispiel für grosse Teile von Dropbox, Spotify, Netflix und Youtube eingesetzt.

Die bereits vorgestellten Programmierkonstrukte—Anweisungen, Sequenzen, Fallunterscheidungen, Schleifen, und Funktionen—sind grundlegende Konstrukte, die bei fast jeder höheren Programmiersprache anzutreffen sind. Im Folgenden erläutern wir wie diese Konstrukte in Python programmiert werden.

Um direkt in Python zu programmieren und die folgenden Code Snippets auch gleich auszuprobieren, können Sie zum Beispiel den online Python Editor von repl.it direkt im Browser benutzen (<https://repl.it/languages/python3>).

3.4.1 Anweisungen und Sequenzen

Eine Zeile Code in Python entspricht einer Anweisung. Ein Python-Programm setzt sich aus mehreren Anweisungen zusammen, die dann von oben nach unten abgearbeitet werden und eine Sequenz darstellen.

Beispiel für Sequenzen Das folgende Programm besteht aus drei Anweisungen. Es fragt den Benutzer nach seinem Namen, speichert den Wert in der Variable *name* und gibt dann “Hallo <name>” aus.

1. `print("Wie heisst du?")`
2. `name = input()`
3. `print("Hallo" + name)`

3.4.2 Fallunterscheidung

Die ein- und zweiseitige Fallunterscheidungen lassen sich durch das If-Else-Konstrukt in Python umsetzen. Bei der einseitigen Fallunterscheidung wird einfach der Else Zweig weggelassen.

Die einseitige Fallunterscheidung:

```
if (Bedingung):  
... Anweisungsblock falls Bedingung wahr ...
```

Die zweiseitige Fallunterscheidung:

```
if (Bedingung):  
... Anweisungsblock falls Bedingung wahr ...  
else:  
... Anweisungsblock falls Bedingung falsch ...
```

Beispiel für eine einseitige Fallunterscheidung:

1. `a = 10`
2. `if (a<12):`
3. `print("Zahl ist zu klein")`

Wichtig hier ist der Einzug / die Einrückung bei der dritten Zeile, in diesem Fall vor dem print-Statement. Dieser Einzug definiert, dass der Anweisungsblock (oder in diesem Fall die print Anweisung) zum if gehört und nur ausgeführt wird, falls die Bedingung ($a < 12$) erfüllt ist.

Aufgabe für eine zweiseitige Fallunterscheidung Erstelle eine Variable und initialisiere diese mit 10. Programmiere eine Fallunterscheidung die abfragt ob der Wert der Variable kleiner als 12 ist, und falls ja gib aus, dass die Zahl zu klein ist. Ansonsten gib aus, dass die Zahl gross genug ist.

Lösung zur zweiseitigen Fallunterscheidung

1. $a = 10$
2. `if (a < 12):`
3. `print("Zahl ist zu klein.")`
4. `else:`
5. `print("Zahl ist gross genug")`

3.4.3 Schleifen

In Python gibt es unterschiedliche Schleifenkonstrukte. Im Folgenden betrachten wir wieder zählerkontrollierte und bedingungskontrollierte Schleifen.

Zählerkontrollierte Schleifen Die zählerkontrollierte Schleife wird in Python mit einer sogenannten For-Loop umgesetzt. Dabei kann mit `range(start, stop)` eine Zahlensequenz/-liste von start nach stop erstellt werden (wobei stop nicht eingeschlossen ist in der Zahlensequenz) über die dann in der For-Loop iteriert wird.

Ein Beispiel für eine zählerkontrollierte Schleife, die die Zahlen von 1 bis 10 in die Konsole schreibt kann dann wie folgt programmiert werden:

`for i in range(1, 11):`

`print(i)`

`range(1,11)` erstellt dabei eine Zahlenliste von 1 bis 10.

3.4.4 Bedingungskontrollierte Schleifen

Die bedingungskontrollierte Schleife kann durch einen While-Loop in Python programmiert werden.

`while (Bedingung):`

... Anweisungsblock falls Bedingung wahr ...

3.4.5 Funktionen

Werden Anweisungsblöcke mehrmals wiederholt oder an unterschiedlichen Stellen im Programm verwendet, lohnt es sich diese in Funktionen auszulagern. Dies erleichtert die Wartbarkeit des Programms und erhöht die Effizienz beim Programmieren. Die Funktion kann Parameter, d.h. Werte von ausserhalb der Funktion entgegennehmen. Es können eine beliebige Anzahl Parameter definiert werden. Doch beim Aufruf sollte darauf geachtet werden, die Parameter korrekt zu übergeben, d. h. die Anzahl und Reihenfolge beachten.

In Python wird eine Funktion mit dem Schlüsselwort “def” erstellt:

```
def funktion1OhneParameter():
... Anweisungsblock der Funktion ...
def funktion2MitParameter (parameter1, parameter2):
... Anweisungsblock der Funktion ...
```

Im Programm kann die Funktion über ihren Namen und mit den Parametern, falls definiert, aufgerufen werden:

```
funktion1OhneParameter ()
oder
funktion2MitParameter (parameter1, parameter2)
```

Referenzen

- Balzert, H. (1999). Lehrbuch Grundlagen der Informatik: Konzepte und Notationen in UML, Java und C++; Algorithmik und Software-Technik; Anwendungen. Spektrum Akad. Verlag.
- Boles, D. (1999). Programmieren spielend gelernt mit dem Java-Hamster-Modell (Vol. 2). Teubner.
- Harper, Robert (1 May 2017). “What, if anything, is a programming-paradigm?”. FifteenEightyFour. Cambridge University Press
- Wagenknecht, C. (2004). Programmierparadigmen: eine Einführung auf der Grundlage von Scheme. Teubner Verlag.

3.5 Folien



**Universität
Zürich**
UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 9: Programmieren.

Lektion 1: Einführung in die Programmierung

Prof. Thomas Fritz, Institut für Informatik

Lernziele

Am Ende dieser Lektion wissen Sie ...

- was Programmieren ist,
- wieso Programmieren wichtig ist.



*«Software is eating
the world»*

- Marc Andreessen



Beispiel aus dem Alltag – Zugverbindung finden

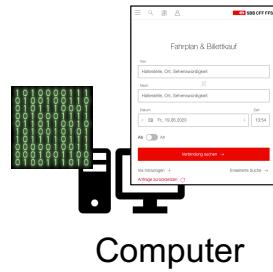


Beispiel aus dem Alltag – Zugverbindung finden

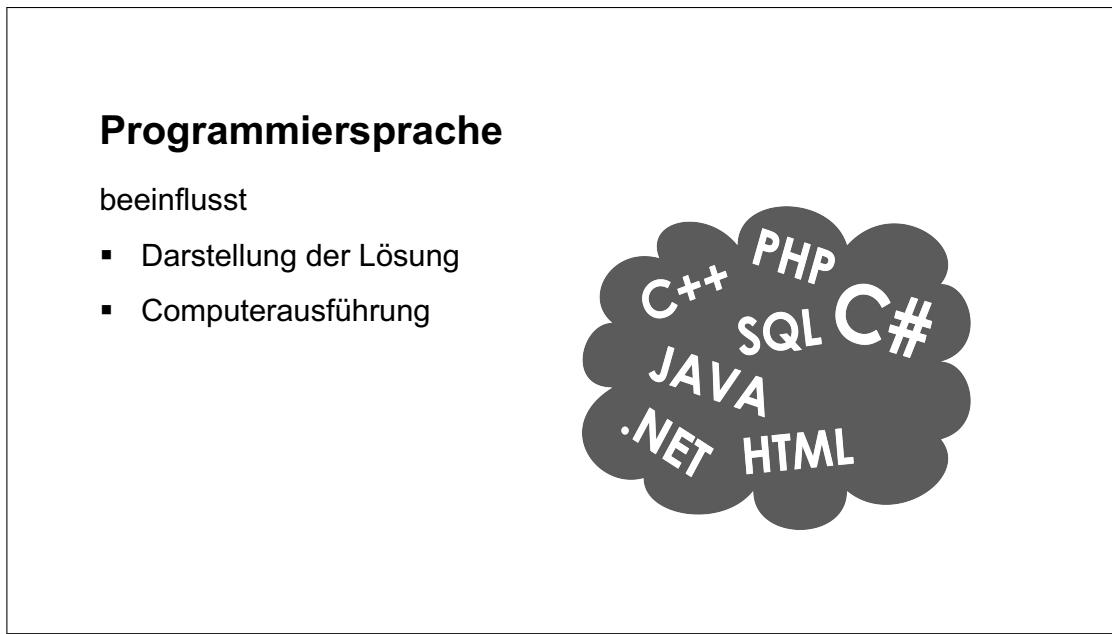
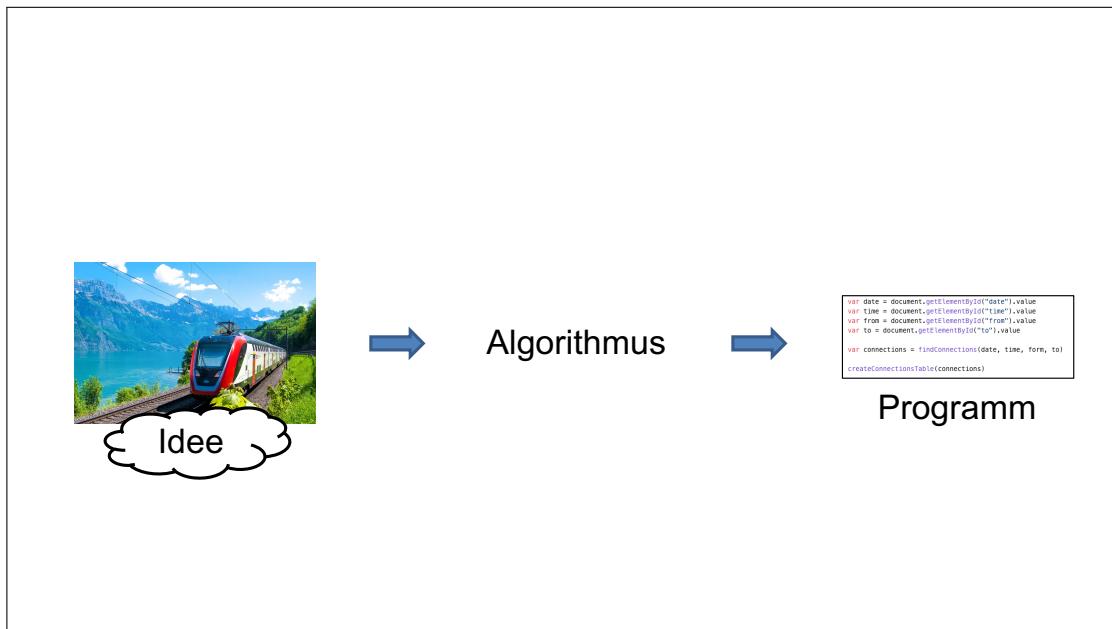


```
var date = document.getElementById("date").value  
var time = document.getElementById("time").value  
var from = document.getElementById("from").value  
var to = document.getElementById("to").value  
  
var connections = findConnections(date, time, from, to)  
createConnectionsTable(connections)
```

Programm



Programmieren



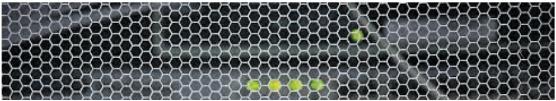
BILDUNG UND GESELLSCHAFT

Montag, 3. Juni 2013 · Nr. 125
Neue Zürcher Zeitung

Programmieren oder programmiert werden

Die Grundlagen der Informatik sollten an den Schulen in einem eigenständigen, obligatorischen Fach unterrichtet werden

tionsgesellschaft leute mit Informatik- „Menschen, die en und Grenzen der m Informationsverar- eurteilen wissen. mer Stundenlang am



komplexität, die Grenzen der Automatisierbarkeit, Sicherheit, Kommunikation, Modellbildung und Problemlösungstechnik. Es geht nicht darum, aus Gymnasiasten Informatikexperten zu machen, sondern ihnen Einblicke zu vermitteln, die zur verantwortungsvollen Nutzung und zur Vermeidung grösseren

Software / Programm



Lime-Tretroller werfen Fahrer ab

Lime hat seine elektrischen Tretroller in der Schweiz stillgelegt, nachdem es möglicherweise durch einen Softwarefehler dazu kam, dass die kleinen Fahrzeuge während der Fahrt unvermittelt stoppten.

Verunglücktes Tesla-Auto fuhr per Autopilot

Vor etwa einer Woche kam der Fahrer eines Tesla-Fahrzeugs bei einem Unfall in Kalifornien ums Leben. Ermittler konzentrieren sich nun auf den Autopiloten.

Grundlegender Softwarefehler in der Boeing 737 Max gefunden

Tests der US-Luftfahrtbehörde decken weitere Fehlfunktionen auf, die erneute Verzögerungen für die Rückkehr des Flugzeugtyps bedeuten.

Software / Programm

Lime-Tretroller werfen Fahrer ab

Lime hat seine elektrischen Tretroller in der Schweiz stillgelegt, nachdem es möglicherweise durch einen Softwarefehler dazu kam, dass die kleinen Fahrzeuge während der Fahrt unvermittelt stoppten.

Verunglücktes Tesla-Auto fuhr per Autopilot

Vor etwa einer Woche kam der Fahrer eines Tesla-Fahrzeugs bei einem Unfall in Kalifornien ums Leben. Ermittler konzentrieren sich nun auf den Autopiloten.

Grundlegender Softwarefehler in der Boeing 737 Max gefunden

Tests der US-Luftfahrtbehörde decken weitere Fehlfunktionen auf, die erneute Verzögerungen für die Rückkehr des Flugzeugtyps bedeuten.

Übersicht Programmieren

- Einführung
- Programmiersprachen
- grundlegende Programmierkonstrukte
- Scratch-Einführung
- Übungen

Referenzen

Stefan Betschon (2013). Porgrammieren oder porgrammiert werden. NZZ. Abgerufen von http://fit-in-it.ch/sites/default/files/small_box/nzz_zehnder_2013-06-03.pdf (letzter Zugriff am 17. Juni 2020)

Andreas Donath (2019). Lime-Tretroller werfen Fahrer ab. golem.de. Abgerufen von <https://www.golem.de/news/softwarefehler-lime-tretroller-werfen-fahrer-ab-1901-138694.html> (letzter Zugriff am 17. Juni 2020)

Reuters (2018). Verunglücktes Tesla-Auto fuhr per Autopilot. NZZ. Abgerufen von <https://www.nzz.ch/mobilitaet/verungluecktes-tesla-auto-fuhr-per-autopilot-id.1371004> (letzter Zugriff am 17. Juni 2020).

awp/dpa (2018). Autopilot beschleunigte Tesla vor tödlichem Crash. NZZ. Abgerufen von <https://www.nzz.ch/panorama/toedlicher-tesla-unfall-autopilot-beschleunigte-vor-crash-id.1393013> (letzter Zugriff am 17. Juni 2020).

Andreas Spaeth (2019). Grundlegender Softwarefehler in der Boeing 737 Max gefunden. NZZ. Abgerufen von <https://www.nzz.ch/mobilitaet/luftfahrt/boeing-737-max-grundlegender-softwarefehler-gefunden-id.1499955?reduced=true> (letzter Zugriff am 17. Juni 2020)

Referenten

Prof. Thomas Fritz & Claudia Vogel
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 9: Einführung in Programmierung

Lektion 2: Programmiersprachen

Prof. Thomas Fritz, Institut für Informatik

Lernziele

Am Ende dieser Lektion ...

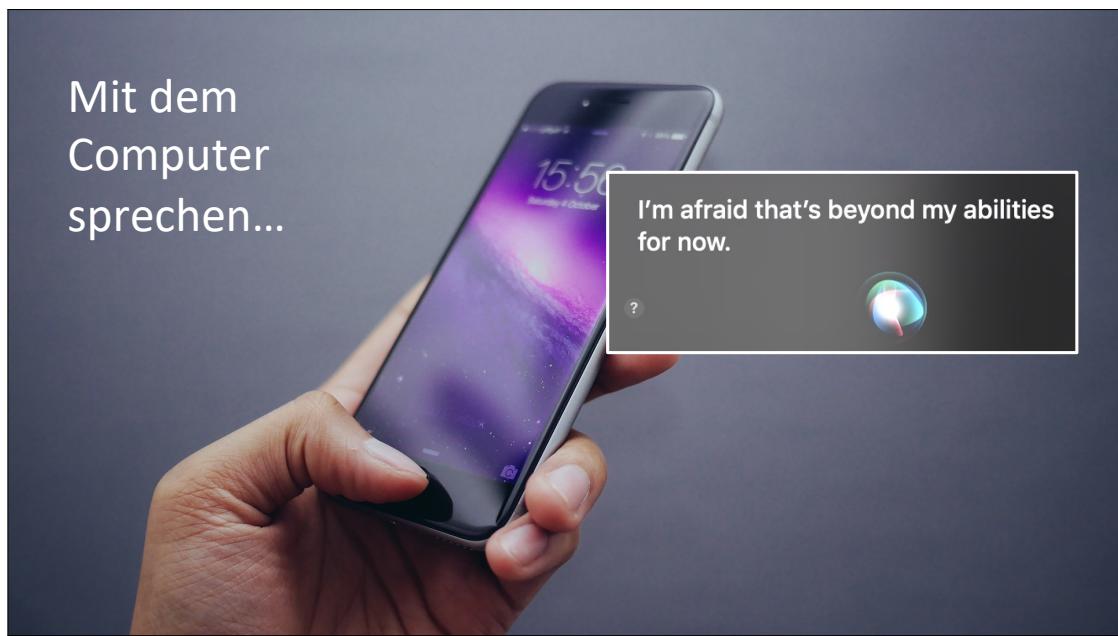
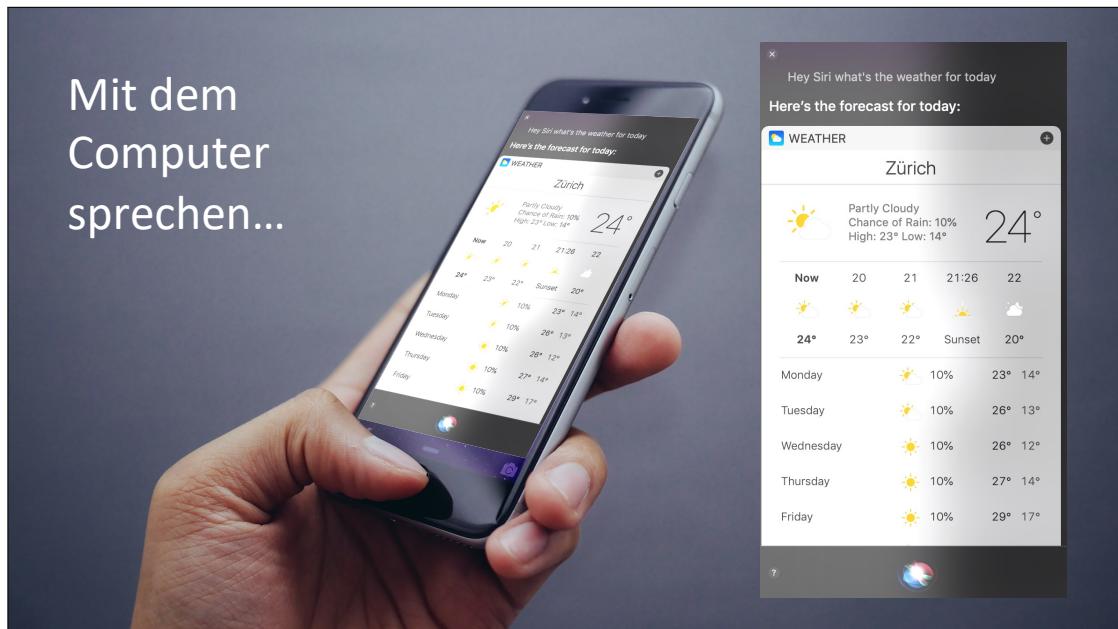
- wissen Sie was eine Programmiersprache ist und wie sie definiert ist,
- kennen Sie die drei Generationen der Programmiersprachen,
- verstehen Sie was Programmierparadigmen sind und können die vier vorgestellten Paradigmen an Hand eines Beispiels erklären und erkennen.

Wie sprechen Sie mit einem Computer?



Mit dem
Computer
sprechen...





3 Generationen der Programmiersprachen

Höhere Programmiersprache

Assemblersprache

Maschinensprache

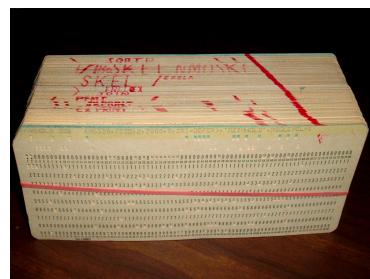


Computer

Generation Maschinensprache



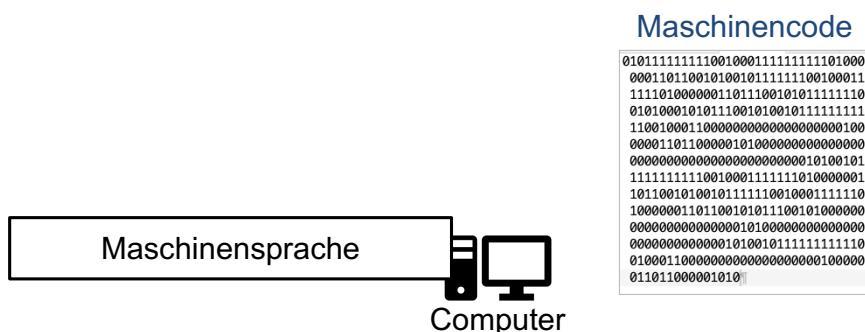
“Programmieren”



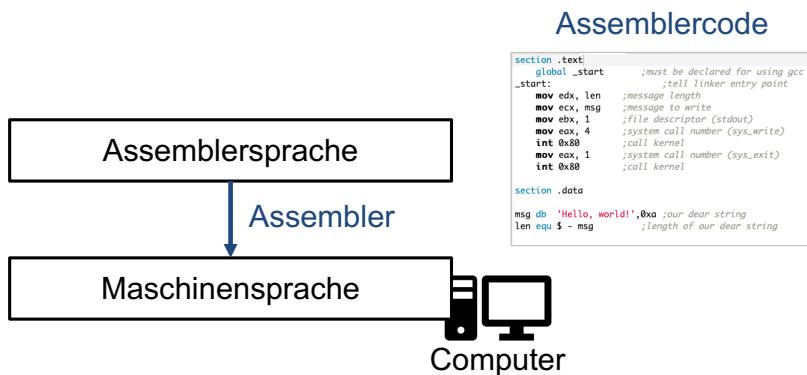
Lochkarten Programm

Bild von Arnold Reinhold, CC BY-SA 3.0
<https://creativecommons.org/licenses/by-sa/3.0/>

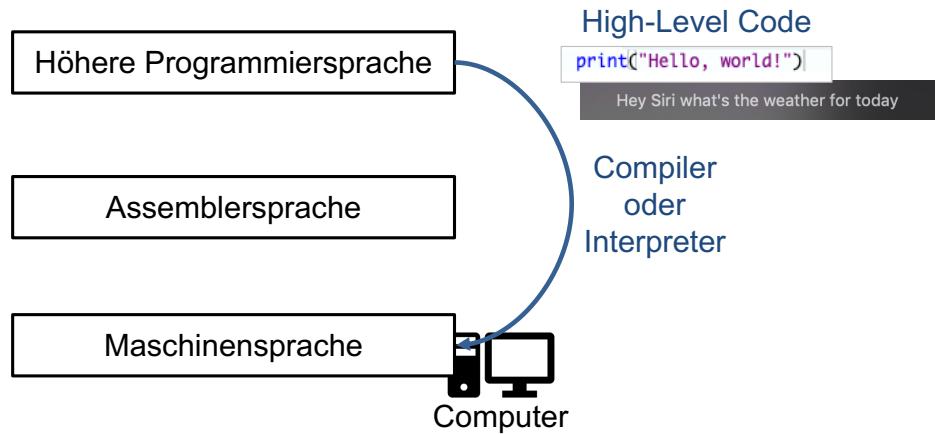
Generationen der Programmiersprachen



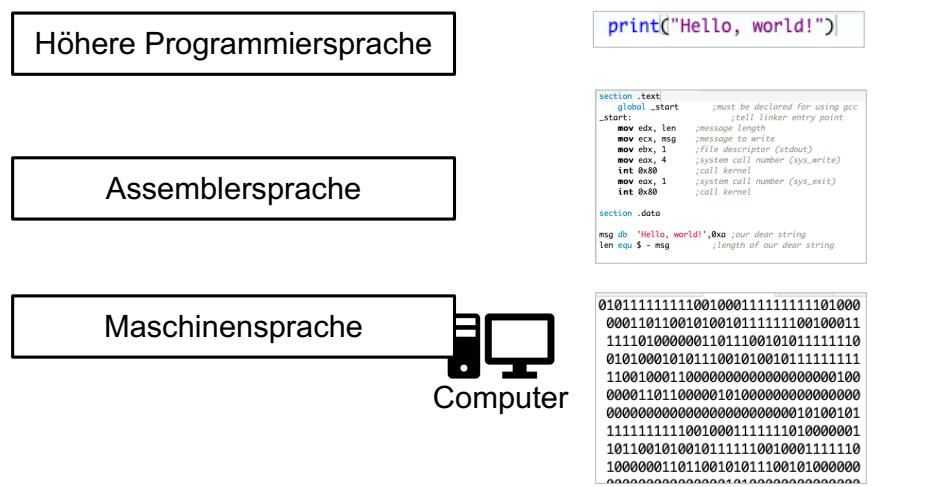
Generationen der Programmiersprachen



Generationen der Programmiersprachen



Generationen der Programmiersprachen



Beispiel: SBB App für Zugverbindungen



```

<html lang="de">
  <!--[if lt IE 9]>
    <script>document.createElement('video');</script>
  <![endif]>
  <!--[if !IE]>
    <script>document.createElement('map');</script>
  <![endif]>
</html>

```

The screenshot shows the SBB mobile application's search interface. It includes input fields for 'Von' (From) and 'Nach' (To), a date selector ('Fr., 19.06.2020'), a time selector ('13:54'), and a red search button labeled 'Verbindung suchen →'. Below the search bar are links for 'Via hinzufügen +' and 'Anfrage zurücksetzen'.

```

<html lang="de">
  <!--[if lt IE 9]>
    <script>document.createElement('video');</script>
  <![endif]>
  <!--[if !IE]>
    <script>document.createElement('map');</script>
  <![endif]>
</html>

```

HTML

```

<html lang="de">
  <!--[if lt IE 9]>
    <script>document.createElement('video');</script>
  <![endif]>
  <!--[if !IE]>
    <script>document.createElement('map');</script>
  <![endif]>
</html>

```

```

var date = document.getElementById("date").nodeValue
var time = document.getElementById("time").nodeValue
var from = document.getElementById("from").nodeValue
var to = document.getElementById("to").nodeValue

var connections = findConnections(date, time, from, to)

createConnectionsTable(connections)

```

Javascript

```

var date = document.getElementById("date").nodeValue
var time = document.getElementById("time").nodeValue
var from = document.getElementById("from").nodeValue
var to = document.getElementById("to").nodeValue

var connections = findConnections(date, time, from, to)

createConnectionsTable(connections)

```

Definition Programmiersprache

- Lexikalik
- Syntax
- Semantik

Definition Programmiersprache

Lexikalik – definiert gültigen Zeichen

A-Z, a-z, 0-9, =, <, >, ...

```
var date* = document.getElementById("date").value
var time0 = document.getElementById("time").value
var from_ = document.getElementById("from").value
var to$ = document.getElementById("to").value

var connections = findConnections(date, time, form, to)
```

Definition Programmiersprache

Syntax – definiert korrekten Aufbau der Sätze

```
var date + time = document.getElementById("date").value
var time = document.getElementById("time").value
var from = document.getElementById("from").value
var to = document.getElementById("to").value

var connections = findConnections(date, time, form, to)

createConnectionsTable(connections)
```

Definition Programmiersprache

Semantik – definiert Bedeutung semantisch korrekter Sätze

```
var date = document.getElementById("date").nodeValue  
var time = document.getElementById("time").nodeValue  
var from = document.getElementById("from").nodeValue  
var to = document.getElementById("to").nodeValue  
  
var connections = findConnections(date, time, from, to)  
  
createConnectionsTable(connections)
```

Definition Programmiersprache

Semantik – definiert Bedeutung semantisch korrekter Sätze

```
var number1 = 2  
var number2 = 4  
var sum = number1 + number2  
  
var divBy0 = sum / 0
```

... wenn alles korrekt



... wenn nicht

```
error: unknown: Unexpected token, expected ";" (1:9)
> 1 | var date + time = document.getElementById("date")
|           ^
2 | var time = document.getElementById("time").value
3 | var from = document.getElementById("from").value
4 | var to = document.getElementById("to").value
```



Programmiersprache Paradigmen

«Paradigma bezeichnet eine fundamentale Denkungsart, einen Denkstil oder ein Denkmuster»

-- Wagenknecht 2004

Programmierparadigmen

- Deklarative Programmierung
- Funktionale Programmierung
- Imperative Programmierung
- Objekt-orientierte Programmierung
- ...

Programmierparadigmen

Deklarativ

- «Was?»
- Beschreibung des gewünschten Endergebnisses
- z.B. SQL, Oz, ...

Programmierparadigmen

Deklarativ

"Gib mir alle
Zugverbindungen von
Sargans nach Zürich am
19.06.2020 zwischen
13:00 und 14:00 Uhr"

WAS?



Datenbank: Speichert
alle Zugverbindungen

Id	Time	From	To
1	13:28	Sargans	Zürich
2	13:31	Sargans	Chur
...			

Programmierparadigmen

Funktional

- Stützt auf mathematischen Grundlagen
- Fokus auf Funktionen
 - $y = f(x)$
- z.B. Scheme, SML, Lisp, Haskell, ...

Programmierparadigmen

Funktional

Hinfahrt	
Sargans → Zürich HB	
2. Klasse	
1x Streckenbillett, Halbtax-Abo	①
Gültig: Fr., 19.06.2020	
bis Sa., 20.06.2020 05:00	Ab CHF 16.50
Rückfahrt	
Zürich HB → Sargans	
2. Klasse	
1x Streckenbillett, Halbtax-Abo	①
Gültig: So., 21.06.2020	
bis Mo., 22.06.2020 05:00	Ab CHF 16.50
Gesamtpreis	CHF 33.00

Berechne
Gesamtpreis

```
fun sum [] = 0
| sum (x::xs) = x + sum xs
```

Programmierparadigmen

Imperativ

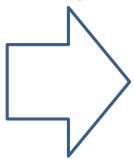
- «Wie?»
- Genauer Ablauf: «Zuerst mach das, dann das...»
- z.B. C, PHP, Python, ...

Programmierparadigmen

Imperativ

Hinfahrt	
Sargans	→ Zürich HB
2. Klasse	
1x Streckenbillett, Halbtax-Abo	
Gültig: Fr., 19.06.2020	Ab CHF 16.50
bis Sa., 20.06.2020 05:00	
Rückfahrt	
Zürich HB	→ Sargans
2. Klasse	
1x Streckenbillett, Halbtax-Abo	
Gültig: So., 21.06.2020	Ab CHF 16.50
bis Mo., 22.06.2020 05:00	
Gesamtpreis	CHF 33.00

Berechne
Gesamtpreis



```
var total = 0;  
for (connection in bookedConnections) {  
    total = total + connection.price  
}
```

Programmierparadigmen

Objektorientiert

- Fokus auf Objekte: Eigenschaften, Manipulation, und Kommunikation von und zwischen Objekten
- z.B. Java, Smalltalk

Programmierparadigmen

Objektorientiert

Hinfahrt	
Sargans	→ Zürich HB
2. Klasse	
1x Streckenbillett, Halbtax-Abo	
Gültig: Fr., 19.06.2020	①
bis Sa., 20.06.2020 05:00	
	Ab CHF 16.50
Rückfahrt	
Zürich HB	→ Sargans
2. Klasse	
1x Streckenbillett, Halbtax-Abo	
Gültig: So., 21.06.2020	①
bis Mo., 22.06.2020 05:00	
	Ab CHF 16.50
Gesamtpreis	CHF 33.00

Berechne
Gesamtpreis



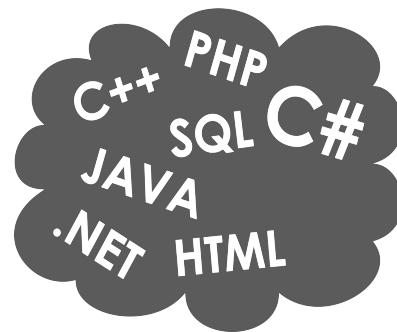
```
class Connection {
    private Location from, to;
    private float cost;

    public float getCost() {
        return cost;
    }
    ...
}
```

```
class ShoppingCart {
    private List<Connection> allConns;

    public float getTotalCost() {
        float totalCost;
        for (Connection con : allConns) {
            totalCost = con.getCost() + totalCost;
        }
        return totalCost;
    }
    ...
}
```

Programmiersprachen



Referenzen

- Balzert, H. (1999). *Lehrbuch Grundlagen der Informatik: Konzepte und Notationen in UML, Java und C++; Algorithmik und Software-Technik; Anwendungen*. Spektrum Akad. Verlag.
- Wagenknecht, C. (2004). *Programmierparadigmen: eine Einführung auf der Grundlage von Scheme*. Teubner Verlag.

Referentin

Prof. Thomas Fritz
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 9: Einführung in Programmierung

Lektion 3: Sequenzen und Variablen

Prof. Thomas Fritz, Institut für Informatik

Lernziele

Am Ende dieser Lektion ...

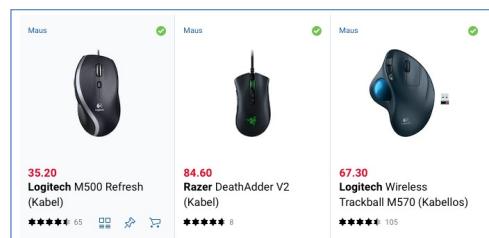
- wissen Sie was ein Datentyp ist,
- verstehen Sie was eine Variable ist und wie man sie nutzt,
- wissen Sie was eine Sequenz ist und wie man sie anwendet.

Online Bestellung der Home-Office Einrichtung: Tastatur, Maus, Monitor

1. Produkt suchen



2. Produkt auswählen



Online Bestellung der Home-Office Einrichtung: Tastatur, Maus, Monitor

3. Produkt in den Warenkorb legen

In den Warenkorb

4. Zur Kasse & Bestellung absenden

Online Bestellung der Home-Office Einrichtung: Tastatur, Maus, Monitor

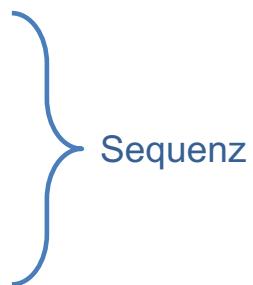
1. Produkt suchen
 2. Produkt auswählen
 3. Produkt in Warenkorb legen
 4. Zur Kasse & Bestellung absenden
- ← Anweisung

Anweisung

- elementare Einheit im Programm
- steht für einen einzelnen Abarbeitungsschritt im Algorithmus
- auch Statement, Kommando, oder Befehl genannt

Online Bestellung der Home-Office Einrichtung: Tastatur, Maus, Monitor

1. Produkt suchen
2. Produkt auswählen
3. Produkt in Warenkorb legen
4. Zur Kasse & Bestellung absenden

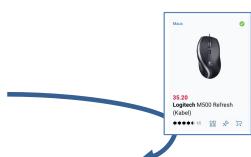


Sequenz

- Abfolge von Anweisungen
- sequenzielle Ausführung in der gegebenen Reihenfolge von oben nach unten
- kein Überspringen von Anweisungen

Online Bestellung der Home-Office Einrichtung: Tastatur, Maus, Monitor

1. Produkt suchen
2. Produkt auswählen
3. Produkt in Warenkorb legen
4. Zur Kasse & Bestellung absenden



Variablen: Leerer Warenkorb

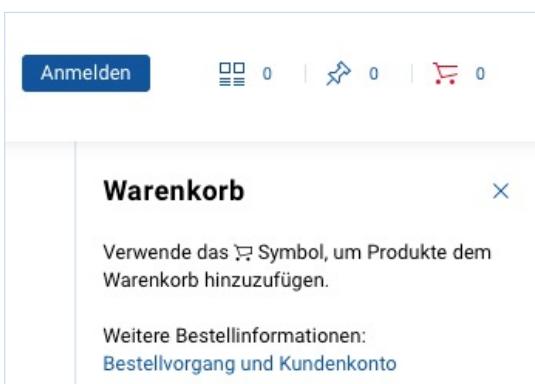
A screenshot of a web-based shopping cart interface. At the top, there is a navigation bar with a 'Anmelden' button, a user icon (0), a search icon (0), and a shopping cart icon (0). Below this is a header with the text 'Warenkorb' and a close 'X' button. A message in the center says: 'Verwende das ✦ Symbol, um Produkte dem Warenkorb hinzuzufügen.' Below this message are links for 'Weitere Bestellinformationen:', 'Bestellvorgang und Kundenkonto'. To the right of the screenshot, there is some code.

```
var shoppingCartSize;
```

Variable

- speichert einfache Werte in Programmen
- erlaubt auf den gespeicherten Wert zuzugreifen

Variablen: Leerer Warenkorb



```
var shoppingCartSize;
```

Deklaration

'var' erstellt neue Variable

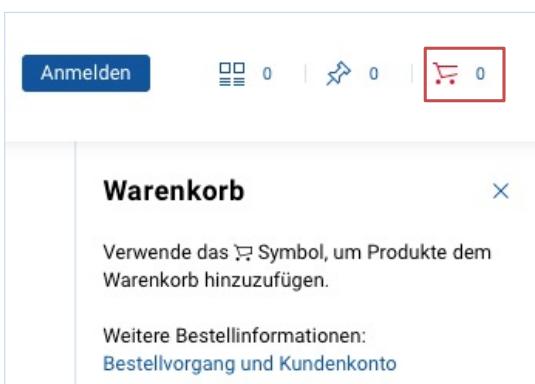
Verwendung von Variablen

Deklaration

- erstellt Variable
- definiert Typ und Namen der Variable

```
var shoppingCartSize;
```

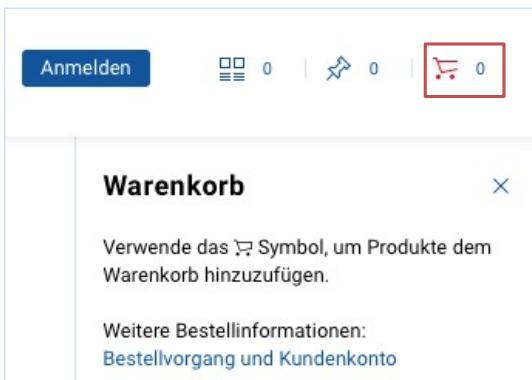
Variablen: Leerer Warenkorb



```
var shoppingCartSize = 0;
```

Deklaration & Initialisierung

Variablen: Leerer Warenkorb



```
var shoppingCartSize;
```

Deklaration

```
shoppingCartSize = 0
```

Initialisierung

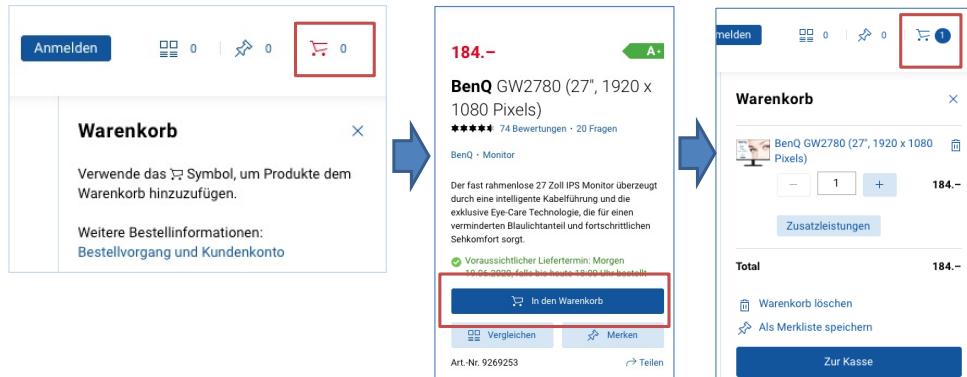
Verwendung von Variable

Initialisierung

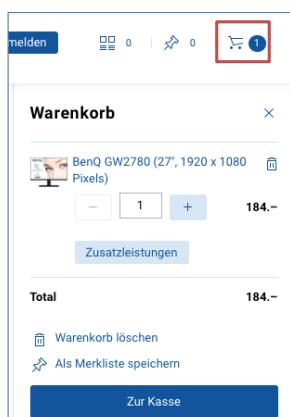
- Erstmaliges Speichern von einem Wert in der Variable

Name	Value
shoppingCartSize	0
...	...

Variablen: Produkt im Warenkorb hinzufügen



Variablen: Aktualisierter Warenkorb



```
var shoppingCartSize = 0;
```

```
shoppingCartSize = 1;
```

Zuweisung

Verwendung von Variablen

Zuweisung

- Veränderung des Wertes

Variablen: Bestellübersicht



```
var shoppingCartSize = 1
```

```
var orderOverviewTitle = "Bestellung " +  
    shoppingCartSize + " Artikel"
```

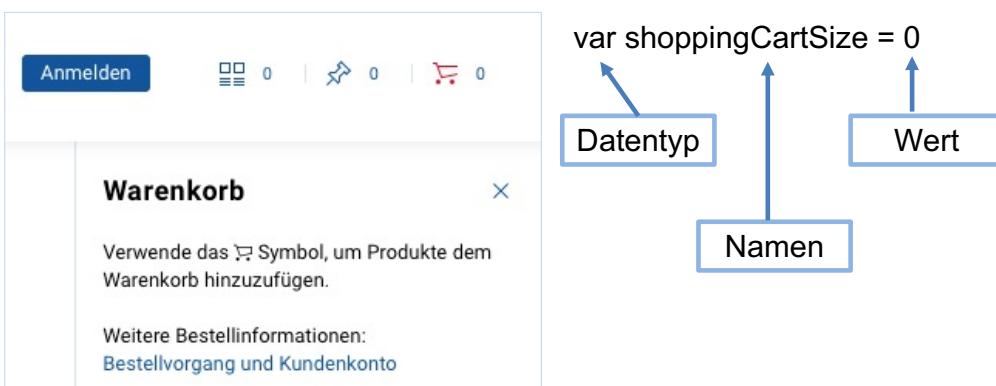
Gebrauch

Verwendung von Variablen

Gebrauch

- Zugriff auf Variable, resp. auslesen des gespeicherten Wertes

Variablen: Eigenschaften



Datentypen: Detailansicht eines Produkts



Datentyp

- beschreibt Menge von Datenobjekten,
- Datenobjekte eines Typs haben alle die gleiche Struktur, und
- auf Datenobjekten des gleichen Typs können die gleichen Operationen durchgeführt werden

Datentypen: Detailansicht eines Produkts



```
var price = 184.00
```

Arten von Datentypen

Numerische Datentypen

- Ganze Zahlen (1, 49)
- Gleitkommazahlen (5.90, 1.999)

Datentypen: Detailansicht eines Produkts



```
var name = "BenQ .."  
var description = "Der fast  
rahmlose ..."  
var image = "../img/2784.jpg"
```

Arten von Datentypen

Zeichen

- Einzelne Ziffern
`var currency = '€'`

Zeichenkette

- Wörter, Sätze
`var productName = "BenQ"`

Datentypen: Detailansicht eines Produkts



```
var ratings = List<Ratings>(74);
```

```
var ratings = ['gut', 'sehr gut', ...]
```

```
var ratings = [rating1, rating2, ...]
```

Arten von Datentypen

Listen

- Kollektion von Datenobjekten

Datentypen: Detailansicht eines Produkts



```
var productInStock = true;
```

Arten von Datentypen

Wahrheitswerte

- Boolean
- Entweder wahr oder falsch

Weiteres Beispiel Datentypen:

The left screenshot shows a product page for a BenQ GW2780 monitor. The price is listed as 184.-. The right screenshot shows a shopping cart with one item: the BenQ GW2780 monitor at 184.-.

```
var product1 = {
  name: "BenQ",
  price: 184.00,
  amount: 1
}
```

```
var products = [product1]
```

Variablen speichern Werte

Datentyp definiert was gespeichert wird, z.B.:

$\downarrow \frac{1}{9}$ Zahl

$\downarrow \frac{n}{z}$ Zeichenketten

$\times \checkmark$ Wahrheitswert

$\frac{1}{2} \equiv \frac{3}{3}$ Liste

...

Referenzen

Boles, D. (1999). Programmieren spielend gelernt mit dem Java-Hamster-Modell (Vol. 2). Teubner.

Referentin

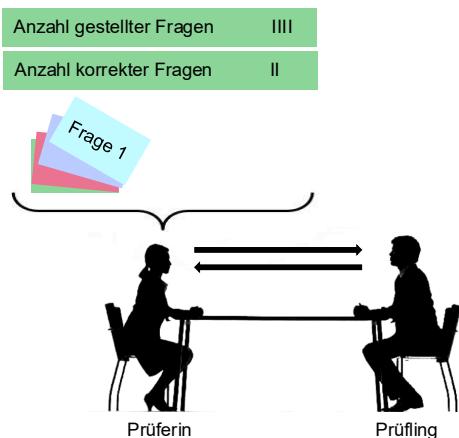
Prof. Thomas Fritz
Institut für Informatik

© Universität Zürich
Digital Society Initiative

Einführung in die Programmierung

- **Szenario: Quiz**
- Code
- Programmierkonstrukte im Detail

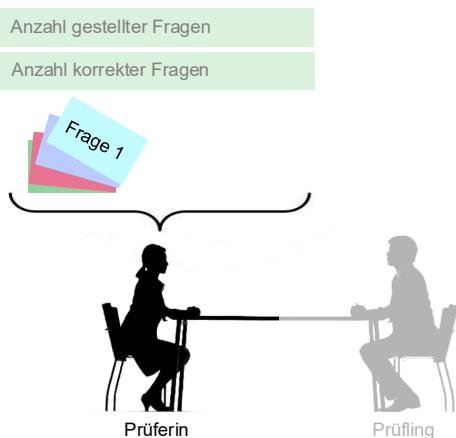
Quiz Beispielprogramm – Szenario



Eine Prüferin fragt einen Prüfling ab.

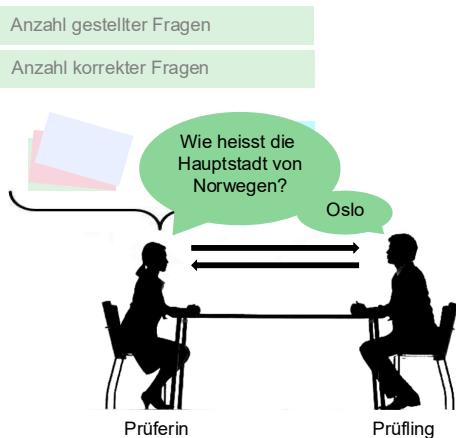
- Gegeben sei ein Stapel mit Lernkarten. Auf der Vorderseite ist jeweils die Frage notiert, auf der Rückseite die Antwort.
- Die Lernkarten werden nun der Reihe nach abgefragt. Der Prüfling antwortet und die Prüferin entscheidet, ob die Antwort korrekt war.
- Wenn alle Lernkarteien abgearbeitet sind, wird der Prüfling über die Anzahl korrekter Antworten informiert.

Quiz Beispielprogramm – Formalisierung der Anforderungen



- ① Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **gestellten Fragen** notiert wird
- ② Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **korrekten Antworten** notiert wird
- ③ Prüferin nimmt den **Stapel** in die Hand

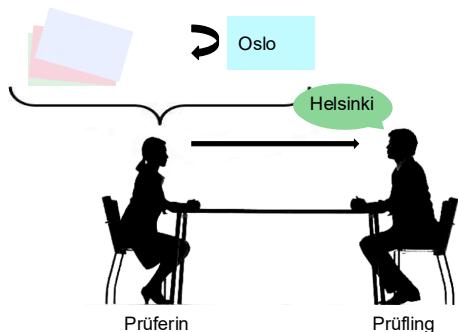
Quiz Beispielprogramm – Formalisierung der Anforderungen



- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**
 - Prüferin nimmt oberste Karte vom Stapel
 - Prüferin stellt dem Prüfling die **Frage**, dieser gibt eine **Antwort**

Quiz Beispielprogramm – Formalisierung der Anforderungen

Anzahl gestellter Fragen	1
Anzahl korrekter Fragen	1

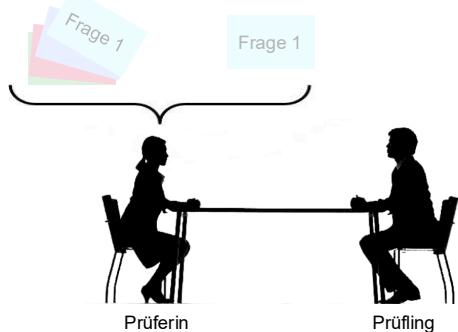


- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- ...
 - Prüferin kontrolliert, ob die **Antwort** korrekt ist
- Wenn korrekt:
- Prüferin gibt dem Prüfling positives Feedback
 - Prüferin erhöht die Anzahl **der korrekten Antworten** um 1
- Wenn falsch:
- Prüferin gibt dem Prüfling negatives Feedback

Quiz Beispielprogramm – Formalisierung der Anforderungen

Anzahl gestellter Fragen	1
Anzahl korrekter Fragen	1



- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- Prüferin nimmt oberste Karte vom Stapel
- Prüferin stellt dem Prüfling die **Frage**, dieser gibt eine **Antwort**
- Prüferin kontrolliert, ob die **Antwort** korrekt ist
 - ...
 - ...
- Prüferin erhöht Anzahl der gestellten Fragen um 1

Quiz Beispielprogramm – Formalisierung der Anforderungen

- ① Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **gestellten Fragen** notiert wird
- ② Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **korrekten Antworten** notiert wird
- ③ Prüferin nimmt den **Stapel** in die Hand
- ④ Prüferin geht durch jede **Lernkarte im Stapel**
 - Prüferin nimmt oberste Karte vom Stapel
- Prüferin stellt dem Prüfling die **Frage**, dieser gibt eine **Antwort**
- Prüferin kontrolliert, ob die **Antwort** korrekt ist
 - ...
 - ...
- Prüferin erhöht Anzahl der gestellten Fragen um 1
- ⑤ Prüferin sagt, wie viele **korrekte Antworten** von allen **gestellten Fragen** der Prüfling beantwortet hat

Quiz Beispielprogramm – Programmierung

```

# Initialize the total number of questions
nr_total_questions_asked = 0

# Initialize the number of correct answers
nr_correct_answers = 0

# 2 dimensional list. Nested list contains a list with 2 entries.
# The first entry holds the question, the second the answer as a string.
questionnaire = [
    ["Hauptstadt von Norwegen?", "Oslo"],
    ["Hauptstadt von Finnland?", "Helsinki"]
]

# Go through every question answer pair
for question_answer_pair in questionnaire:
    # First entry is the question
    # Second entry is the question
    question = question_answer_pair[0]
    answer = question_answer_pair[1]
  
```

https://files_ifi_uzh_ch_ddis_web-python/

Einführung in die Programmierung

- Szenario: Quiz
- **Code**
- Programmierkonstrukte im Detail

Quiz Beispielprogramm – Formalisierung der Anforderungen

- ① Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **gestellten Fragen** notiert wird
 - ② Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **korrekten Antworten** notiert wird
 - ③ Prüferin nimmt den **Stapel** in die Hand
 - ④ Prüferin geht durch jede **Lernkarte** im **Stapel**
 - Prüferin nimmt oberste Karte vom Stapel
- Prüferin stellt dem Prüfling die **Frage**, dieser gibt eine **Antwort**
 - Prüferin kontrolliert, ob die **Antwort** korrekt ist
 - ...
 - ...
 - Prüferin erhöht Anzahl der gestellten Fragen um 1
- ⑤ Prüferin sagt, wie viele **korrekte Antworten** von allen **gestellten Fragen** der Prüfling beantwortet hat

Quiz Beispielprogramm – Von Anforderungen zu Code

Formalisierte Anforderungen

- ① Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **gestellten Fragen** notiert wird
- ② Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **korrekten Antworten** notiert wird
- ③ Prüferin nimmt den **Stapel** in die Hand
- ④ Prüferin geht durch jede **Lernkarte im Stapel**
 - Prüferin nimmt oberste Karte vom Stapel
 - Prüferin stellt dem Prüfling die **Frage**. Dieser gibt eine **Antwort**
 - Prüferin kontrolliert, ob die **Antwort** korrekt ist
 - ...
 - ...
 - Prüferin erhöht Anzahl der gestellten Fragen um 1
- ⑤ Prüferin sagt, wie viele **korrekte Antworten** von allen **gestellten Fragen** der Prüfling beantwortet hat

Code

```

nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]

for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
      str(nr_total_questions_asked))

```

Quiz Beispielprogramm – Code

- ① Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **gestellten Fragen** notiert wird
- ② Prüferin bereitet ein Papier vor, auf welchem die Anzahl der **korrekten Antworten** notiert wird

```

nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]

for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
      str(nr_total_questions_asked))

```

Quiz Beispielprogramm – Code

- ③ Prüferin nimmt den **Stapel** in die Hand

Stapel

Frage ↘ Antwort

[“Wie heisst die Hauptstadt von Norwegen”, “Oslo”]

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ③ Prüferin nimmt den **Stapel** in die Hand

Stapel



[
[“Wie heisst die Hauptstadt von Norwegen?”,
“Oslo”],
[“Wie heisst die Hauptstadt von Finnland?”,
“Helsinki”]
]

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- Prüferin nimmt oberste Karte vom Stapel
- Die **neue** Lernkarte aus questionnaire wird jeweils in der Variablen question_answer_pair zwischengespeichert
- Dieses Konstrukt nennt man eine Schleife
- Alles was zur Schleife gehört, wird eingerückt

```

nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]

for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
      str(nr_total_questions_asked))

```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- Prüferin nimmt oberste Karte vom Stapel
- Zuerst wollen wir die Frage in einer separaten Variablen speichern
- Auf die Frage innerhalb der question_answer_pair Variable können wir mit einem Index zugreifen
- Der Index gibt an, auf welchen Eintrag wir zugreifen möchten
- In der Informatik startet die Indexierung bei 0

```

nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]

for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
      str(nr_total_questions_asked))

```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- ...
- Prüferin stellt dem Prüfling die **Frage**, dieser gibt eine **Antwort**
 - Dieser Code öffnet ein Fenster mit dem Text der Variablen question und einem Textfeld, in welches der **Prüfling** die Antwort eingeben kann
 - Die Antwort wird in der Variable answer gespeichert

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- ...
- ...
- Prüferin kontrolliert, ob die **Antwort** korrekt ist

- Auf die Antwort innerhalb der question_answer_pair Variable können wir mit dem Index 1 zugreifen

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- ...
- ...
- ...

Wenn korrekt:

- Prüferin gibt dem Prüfling positives Feedback
- Prüferin erhöht die Anzahl der **korrekten Antworten** um 1

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]

for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]

    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1

print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- ...
- ...
- ...

Wenn falsch:

- Prüferin gibt dem Prüfling negatives Feedback

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]

for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]

    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1

print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ④ Prüferin geht durch jede **Lernkarte** im **Stapel**

- ...
- ...
- ...
- Prüferin erhöht Anzahl der gestellten Fragen um 1
- Danach nimmt die Prüferin die nächste Karte bzw. question_answer_pair

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Code

- ⑤ Prüferin sagt, wie viele **korrekte Antworten** von allen **gestellten Fragen** der Prüfling beantwortet hat

```
nr_total_questions_asked = 0
nr_correct_answers = 0
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
for question_answer_pair in questionnaire:
    question = question_answer_pair[0]
    answer = raw_input(question)
    correct_answer = question_answer_pair[1]
    if answer == correct_answer:
        print("Correct!")
        nr_correct_answers = nr_correct_answers + 1
    else:
        print("Wrong!")
    nr_total_questions_asked = nr_total_questions_asked
    + 1
print("Correctly " + str(nr_correct_answers) + " of total " +
str(nr_total_questions_asked))
```

Quiz Beispielprogramm – Demonstration Visualisierung

The screenshot shows the Online Python Tutor interface. On the left, the code for a quiz program is displayed:

```
1 # Initialize the total number of questions
2 nr_total_questions_asked = 0
3
4 # Initialize the number of correct answers
5 nr_correct_answers = 0
6
7 # 2 dimensional list. Nested list contains a list
8 # The first entry holds the question, the second
9 questionnaire = [
10     ["Hauptstadt von Norwegen?", "Oslo"],
11     ["Hauptstadt von Finnland?", "Helsinki"]
12 ]
13
14 # Go through every question answer pair
15 for question_answer_pair in questionnaire:
16     # First entry holds the question
17     question = question_answer_pair[0]
18     # Second entry holds the answer
```

On the right, the state of global variables and objects is visualized:

- Global variables:** nr_total_questions_asked = 0, nr_correct_answers = 0, questionnaire = [list, list]
- Objects:** A list object containing two entries: "Hauptstadt von Norwegen?" and "Oslo". Another list object contains "Hauptstadt von Finnland?" and "Helsinki".

Below the visualization, there are buttons for "Edit code", navigation ("First", "Back", "Step 5 of 6", "Forward", "Last"), and sharing ("Generate URL", "Generate embed code").

<https://files.ifi.uzh.ch/ddis/web-python/>

Einführung in die Programmierung

- Szenario: Quiz
- Code
- **Programmierkonstrukte im Detail**

Konstrukte im Detail – Übersicht

- Kommentare
- Variablen
- Werte
- Operatoren
- Funktionen
- Iteration
- Bedingungen

Konstrukte im Detail – Kommentare

- Kommentare haben keine Auswirkungen auf das Programm
- Sie dienen dazu, den Code zu erklären
- Ein Kommentar beginnt mit einem Hashtag #

```
# Initialize the total number of questions  
nr_total_questions_asked = 0
```

```
# Initialize the number of correct answers  
nr_correct_answers = 0
```

Konstrukte im Detail – Variablen

- Variablen speichern Werte
- Die Zuweisung eines Wertes erfolgt mit dem Gleichheitszeichen =
- Variablen können jederzeit überschrieben werden
- Leerzeichen und Sonderzeichen sind in Variablennamen nicht erlaubt
- Name der Variablen kann beliebig gewählt werden

```
# Initialize the total number of questions  
nr_total_questions_asked = 0
```

```
# Initialize the number of correct answers  
nr_correct_answers = 0
```

...

```
# Increment the question counter  
nr_total_questions_asked =  
    nr_total_questions_asked + 1
```

Konstrukte im Detail – Werte

- Werte sind fundamentale Bausteine für die Konstruktion eines Programmes
- Werte haben einen Datentyp, welcher sich durch unterschiedliche Wertbereiche definiert.
- Bei Zahlen z.B.
 - **Integers (int)**: Ganze Zahlen
 - 3
 - **FLOATS (float)**: Mit Dezimalstelle
Das Dezimalstellentrennzeichen ist ein Punkt!
 - 2.0
 - 3.22123

```
# Initialize the total number of questions
nr_total_questions_asked = 0 ← Integer
```

```
# Initialize the number of correct answers
nr_correct_answers = 0 ← Integer
```

Konstrukte im Detail – Werte

- Zeichenketten sind **Strings (str)**
- Werden mit " oder mit "" deklariert
 - 'Hello World'
 - "Hello World"

```
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
...
if answer == correct_answer:
    print("Correct!")
```

Konstrukte im Detail – Werte

- **Boolsche Variablen (Booleans, bool)**
kennen nur zwei Werte
 - True
 - False
- Sie werden oft als Entscheidungsvariablen bei Bedingungen oder Schleifen verwendet
- Bedingungen werden ausgewertet und geben eine boolsche Variable zurück
- Vergleiche werden mit == ausgeführt

```
correct_answer = question_answer_pair[1]

if answer == correct_answer:
    print("Correct!")
    nr_correct_answers = nr_correct_answers + 1
```

Konstrukte im Detail – Werte

- **Listen (arrays)** sind eine Reihe von Werten
- Sie werden mit [] erstellt und die darin enthaltenen Werte mit Komma getrennt
 - [1, 2, 4]
 - ['Monty', 'Python']
- Datentypen innerhalb einer Liste können gemischt werden
 - ['monty', 10, 2.2]

```
["Wie heisst die Hauptstadt von Norwegen?", "Oslo"]
```

```
["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
```

Konstrukte im Detail – Werte

- Listen können verschachtelt werden.
D.h., dass eine Liste in einer Liste erzeugt werden kann.
- Die Liste questionnaire besteht aus mehreren Frage/Antwort Paaren, welche wiederum selbst Listen sind
- In der Informatik beginnt die Nummerierung mit 0

```
questionnaire = [
    ["Wie heisst die Hauptstadt von Norwegen?", "Oslo"],
    ["Wie heisst die Hauptstadt von Finnland?", "Helsinki"]
]
```

	0	1
0	Hauptstadt von Norwegen?	Oslo
1	Hauptstadt von Finnland?	Helsinki

Konstrukte im Detail – Werte

- Mit [INDEX] wird auf ein Element in einer Liste zugegriffen
- Der Index ist ein Kennzeichen für die Position eines Elementes

```
question =
    ["Hauptstadt von Norwegen?", "Oslo"]
# "Hauptstadt von Norwegen"
question[0]
# "Oslo"
Question[1]
```

...

```
questionnaire = [
    ["Hauptstadt von Norwegen?", "Oslo"],
    ["Hauptstadt von Finnland?", "Helsinki"]
]
# "Oslo"
Questionnaire[0][1]
```

Konstrukte im Detail – Operatoren

- Werte können mittels Operatoren kombiniert werden
- Ähnlich wie in der Mathematik werden dabei die Operationen nach einer bestimmten Reihenfolge ausgeführt:
 1. ()
 2. ** (potenzieren z.B. $2^{**}4 = 16$)
 3. *, /, %
 4. +, -

```
# Increment the question counter  
nr_total_questions_asked =  
    nr_total_questions_asked + 1
```

Konstrukte im Detail – Operatoren

- Strings (Zeichenketten) können mit dem + Operator zusammengefügt werden
- Es können nur Strings miteinander verbunden werden
 - Im Beispiel verwenden wir zwei Integer (nr_correct_answers, nr_total_questions_asked)
 - Damit diese mit den anderen Strings verbunden werden können, müssen die Zahlen zuerst in Strings konvertiert werden
 - Dies ist möglich mit str()

```
"You have correctly answered " + str(nr_correct_answers) + " of  
total " + str(nr_total_questions_asked) + " questions."
```

Konstrukte im Detail – Funktionen

- In Python gibt es viele vordefinierte Funktionen z.B. print(), str(), raw_input()
- Es können auch eigene Funktionen definiert werden
- In den Klammern können Argumente übergeben werden
- Als Resultat können Funktionen neue Werte zurückgeben

```
print("Correct!")  
  
number_as_string = str(122.20)  
  
answer = raw_input(question)
```

Konstrukte im Detail – Iteration

- Die gleiche Aktion kann beliebig oft wiederholt werden, sie ist also in einer Schleife
- Es gibt verschiedene Ausdrücke um zu iterieren
- Wir besprechen an dieser Stelle die for Schleife, welche erlaubt, durch eine Liste zu iterieren
- Es wird durch jedes Element innerhalb der Liste iteriert und das aktuelle Element wird in der angegeben Variablen zwischengespeichert
- Alles was zur Schleife gehört, **muss einen Tabulator eingerückt sein**

```
for VARIABLE in LIST:  
    BODY  
  
...  
  
# Go through every question answer pair  
for question_answer_pair in questionnaire:  
    # Enter loop with tab, first entry holds the question  
    question = question_answer_pair[0]  
    # Leaving loop  
    Print("not in the loop anymore")
```

Konstrukte im Detail – Bedingungen

- Boolsche Algebra / Logik ist ein Bereich der Mathematik
- Ein boolscher Ausdruck ist entweder wahr oder falsch
 - $5 == 5 \rightarrow \text{True}$
 - $5 == 6 \rightarrow \text{False}$
- Operatoren
 - $x == y$ gleich (Achtung: das einfache Gleichheitszeichen ist eine Wertzuweisung)
 - $x != y$ ungleich
 - $x > y$ grösser als
 - $x < y$ kleiner als
 - $x >= y$ grösser oder gleich
 - $x <= y$ kleiner oder gleich

Konstrukte im Detail – Bedingungen

- Das if Konstrukt erlaubt es, Anweisungen auszuführen, wenn eine Bedingung True ist.
- Ist die Bedingung wahr (True), werden die eingerückten Statements ausgeführt

```
if answer == correct_answer:  
    print("Correct!")  
    # Increment the score  
    nr_correct_answers =  
        nr_correct_answers + 1
```

Konstrukte im Detail – Bedingungen

- Das if Konstrukt kann um eine else Klausel erweitert werden
- Sie wird ausgeführt, wenn die if Bedingung falsch (False) ist
- «Entweder oder» in der deutschen Sprache
- Pro if darf nur ein else vorkommen

```
if answer == correct_answer:  
    print("Correct!")  
    # Increment the score  
    nr_correct_answers =  
        nr_correct_answers + 1  
  
else:  
    print("Wrong!")
```

Fazit

Folgende Konstrukte helfen beim Schreiben eines Programmes:

- Kommentare
- Variablen
- Werte
- Operatoren
- Funktionen
- Iteration
- Bedingungen

4 Datenvisualisierung

PROF. DR. SARA IRINA FABRIKANT

Studium Digitale Kursbaustein Effektive Informationsvisualisierung

Begleit-Skript

Dieses Skript bereitet die Inhalte des Kursbaustein Effektive Informationsvisualisierung schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

4.1 Lektion 1: Einstieg und Überblick

Die Lernziele dieses Kursbausteins sind:

- Sie erkennen, wann es sich lohnt, Ihre Zeit in eine Visualisierung zu investieren.
- Sie kennen diverse Visualisierungsmethoden und können diese zielgerecht einsetzen.
- Sie können die für Ihre Aufgaben und Daten angemessene Visualisierungsmethode auswählen und anwenden.
- Sie können selbständig attraktive Visualisierungen mit Microsoft Excel erstellen.

Als Orientierungshilfe ist der Kursbaustein in die 5 Ws der Visualisierung gegliedert: Warum, Was, für Wen, Wozu und Wie wird visualisiert?

Eine Visualisierung macht dann Sinn, wenn damit auf einen Blick aufgenommen werden kann, was sonst in vielen Worten beschrieben werden müsste. Bei der Wahl der Visualisierungsmethode ist es wichtig, für welche Zielgruppe sie bestimmt ist und wie diese sie wahrnimmt.

Zuerst gilt es abzuklären, welche Fragen die Visualisierung beantworten soll und welche Daten dafür zur Verfügung stehen. Erst danach kann eine passende Visualisierungsform gewählt werden. Häufig werden Mengen, Verteilungen, Anteile und Zusammenhänge visualisiert. Zeitreihen und Daten mit Raumbezug sind ebenfalls Gegenstand von Visualisierungen, auf diese wird hier aber nicht genauer eingegangen.

Zum begleiteten Selbststudium werden die folgenden Bücher empfohlen:

- Claus Wilke, <https://serialmentor.com/dataviz/>
- Stephanie Evergreen, <https://stephanieevergreen.com/books/>

4.2 Lektion 2: Warum will ich visualisieren?

Diese Lektion hat folgende Lernziele:

- Sie können informiert entscheiden, wann es sich lohnt, Daten zu visualisieren.
- Sie können selbständig eine statistische Aussage mit einem Bild visualisieren.
- Sie können mit einem guten Beispiel illustrieren, warum die Visualisierung von Daten wichtig und zielführend ist.

Ein wichtiger Grund für die Visualisierung ist wenn die Kommunikation eines Sachverhalts durch ein Bild prägnanter möglich ist, als durch Text. Interessante Muster in den Daten können so viel effizienter erkannt werden, als wenn die Daten in Rohform vorliegen. Dies ist auch der Grund dafür, dass Daten vor statistischen Analysen zuerst grafisch dargestellt werden sollten. Sind die Daten bekannt, beantwortet das erhaltene Datenmuster den Sachverhalt der kommuniziert werden soll. Müssen die Daten zuerst weiter erkundet werden, wirft das visualisierte Muster weiterführende Fragen oder zu testende Hypothesen auf.

Die Visualisierung kann genutzt werden, um real existierende Muster von Scheinmustern in Datenhaufen zu unterscheiden. Dafür muss das Muster durch die grafischen Mittel auch unmissverständlich wahrgenommen werden können.

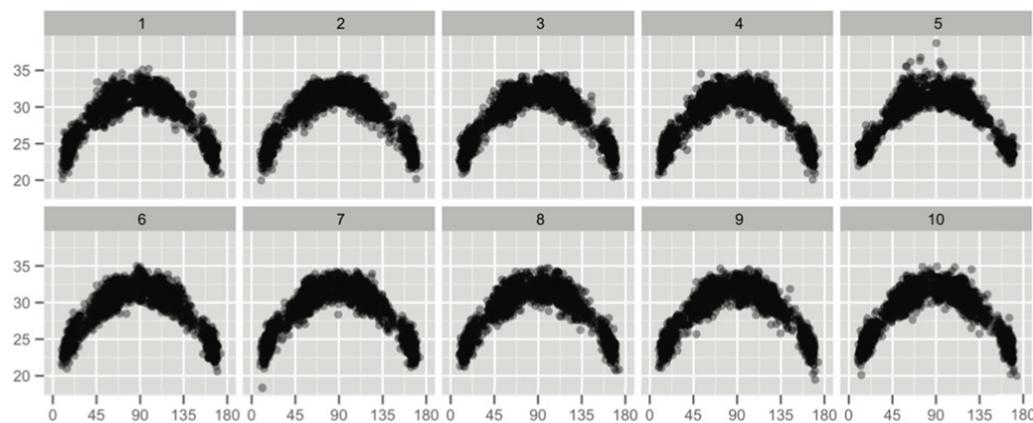


Abbildung aus: Wickham, H., Cook, D., Hofmann, H.. Buja, A. (2010). Graphical Inference for Infovis. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6: 973-979.

4.3 Lektion 3: Was will ich visualisieren?

Diese Lektion hat folgende Lernziele:

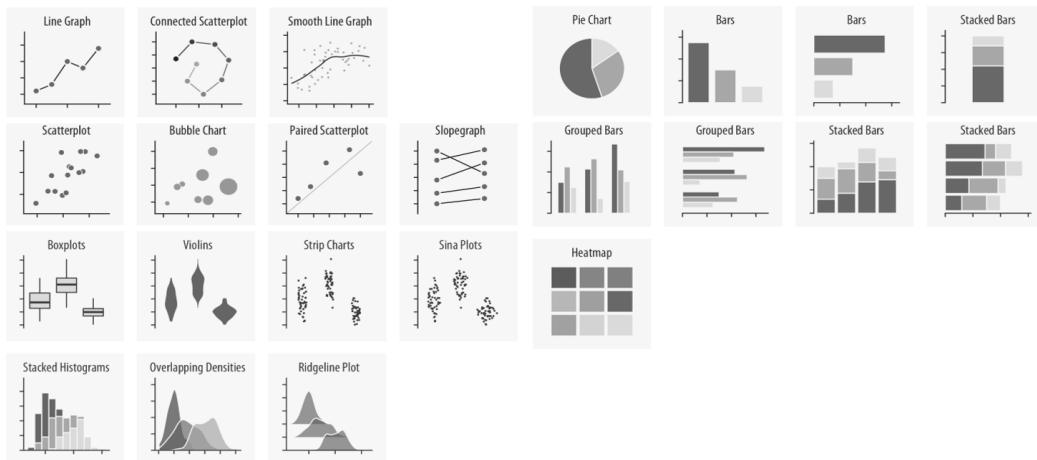
- Sie kennen zentrale Dateneigenschaften (Datentyp, Datenmerkmale, Messniveau, etc.).
- Sie kennen die graphischen Variablen für die Visualisierung.
- Sie haben einen Überblick über das Visualisierungsinventar.

Es gibt qualitative Daten, wie z.B. Texte, die wir unstrukturierte Daten nennen, da nicht in eine Datentabelle passen. Quantitative Daten befinden sich bereits aufbereitet in einer Datentabelle. Bei diesen Daten geht es in einem nächsten Schritt darum, ihre Datentypen und -merkmale zu identifizieren. Diskrete Daten z.B. die Anzahl Einwohner eines Landes und kontinuierliche Daten der durchschnittliche Flächenverbrauch eines Golfplatzes. Dieses Wissen ist wichtig, um richtig zu visualisieren.

Typ	Merkmal	Wert	Messniveau	Beschreibung
Numerisch	kontinuierlich	1.3, 10 ²	ratio	Distanz in Metern
Numerisch	diskret	1, 10, -25	interval	Temperatur in Grad Celsius
alphabetisch	diskret	niedrig, mittel, hoch	ordinal	Erfahrung in Visualisierung
alphabetisch	diskret	Rot, Grün, Blau	nominal	Farbton
Alpha-numerisch	Datum/Zeit	1. August 1291	interval/ratio	Nationalfeiertag

Tabelle 1: Metadaten einer Datentabelle

Für die Darstellung von Daten stehen diverse geometrische Möglichkeiten zur Verfügung. Diese Formen sind nach geometrischen Grundformen (Punkte, Linien und Flächen) sortiert.



Neben der Geometrie steht auch ein Inventar an graphischen Variablen zur Verfügung. Dazu zählen:

- die Lage der Datenpunkte
- die Orientierung der Linien
- der Farbton
- die Helligkeit einer Farbe
- deren Sättigung
- die Textur
- die Form einer Linie oder Fläche

4.4 Lektion 4: Für wen wird visualisiert?

Die Lernziele dieser Lektion sind:

- Sie kennen typische visuelle Wahrnehmungseffekte.
- Sie wissen um das Prinzip der graphischen Prägnanz
- Sie können die Bausteine des Gesetzes der guten Gestalt nennen.

- Sie können die Gestaltbausteine selbstständig in Visualisierungen anwenden.

Genau wie Texte, Musik oder Filme haben Visualisierungen ein Zielpublikum. Der Kontext der Nutzung und das Vorwissen der Personen sind dabei von grosser Bedeutung.

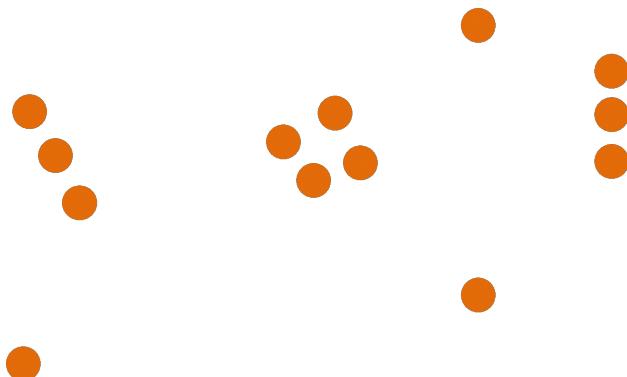
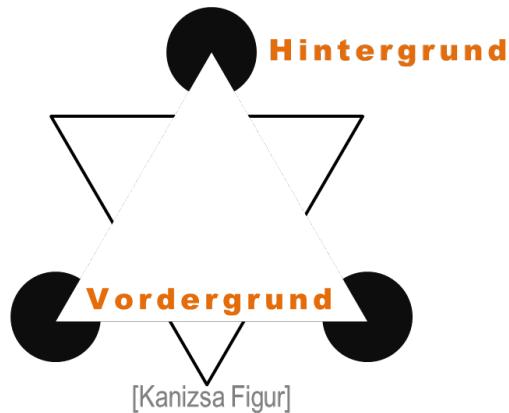
Die menschliche Wahrnehmung ist nicht perfekt und lässt sich täuschen. Wir sind keine physikalischen Lichtsensoren, sondern können mithilfe unserer Erfahrung und des Nutzungskontexts visuelle Eindrücke besser einordnen und daraus sinnvolle Informationen gewinnen.

Das Vorlesen der Farbnamen, die in jeweils anderen Farben geschrieben sind, fällt schwer, da der Inhalt der Wörter nicht zum visuellen Input passt. Wir sprechen dabei von einem Konflikt zwischen einem bottom-up- und einem top-down-Prozess. Der bottom-up-Prozess liefert die sensorische, visuelle Information des Sehsinns, während der top-down-Prozess der Wissensverarbeitung gleichzeitig widersprüchliche Informationen zum Inhalt der Wörter liefert. Was Leute sehen, hängt von ihrem Vorwissen und ihrer Erfahrung ab.

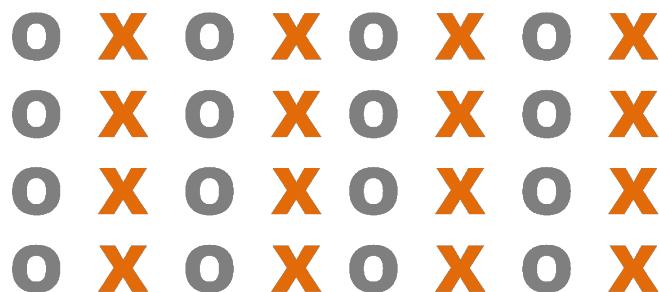
In der Visualisierung sollen zweideutige Darstellungen um jeden Preis verhindert werden. Wichtig dafür ist das Prinzip der graphischen Prägnanz. Mittels graphischer Kontraste kann inhaltlich Wichtiges auch wahrnehmungspsychologisch in den Vordergrund gerückt werden. Alles weniger Wichtige wird hingenommen in den Hintergrund verschoben.



Gestaltpsychologen haben Prinzipien postuliert, die visuelle Prägnanz schaffen, um bewusst Gestalt vom Hintergrund zu unterscheiden.



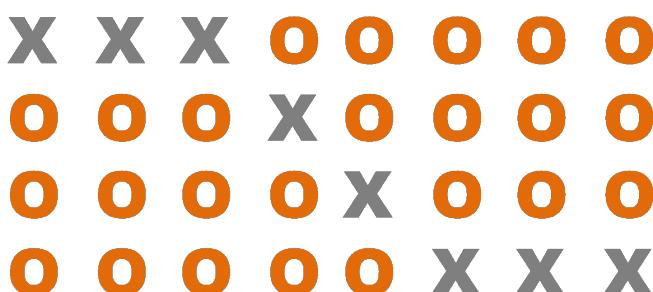
Um zweideutige Informationen zu verhindern, kann das Prinzip der Nähe angewandt werden. Elemente, die graphisch Gruppen bilden, werden als zusammengehörig identifiziert. Auch die Ähnlichkeit ist wichtig bei der Erkennung von zusammengehöriger Information.



Ein weiteres wichtiges Prinzip ist das der Geschlossenheit. Trotz der unterbrochenen Linien nehmen wir die Figur als oranges Quadrat wahr.



Auch das Prinzip der Kontinuität hilft, Gruppen zu erkennen.



4.5 Lektion 5: Wozu visualisiere ich?

Diese Lektion hat folgende Lernziele:

- Sie können gezielte Frage stellen um Mengen, Verteilungen, Anteile, oder Zusammenhänge zu visualisieren.
- Sie sind befähigt, den geeigneten Diagrammtyp für Ihre Fragestellung auszuwählen.
- Sie können diverse Visualisierungslösungen vorschlagen, um Mengen, Verteilungen, Anteile oder Zusammenhänge effektiv darzustellen.

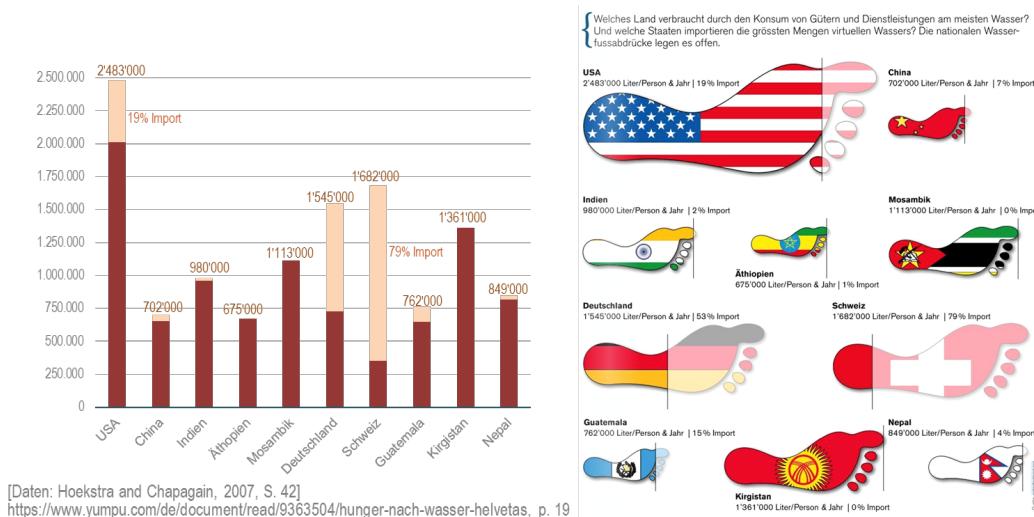
4.5.1 Visualisierung von Mengen

“Wie viel gibt es von x, y oder z in meinem Datensatz?” ist hier die Frage. Ein Beispiel dafür ist der durchschnittliche virtuelle Wasserfußabdruck von Ländern im globalen Vergleich). Als Grundlage für die Visualisierung dient die Tabelle 2.

Land	Verbrauch (Liter/Person und Jahr)	Import (%)
USA	2'483'000	19
China	702'000	7
Indien	980'000	2
Äthiopien	675'000	1
Mosambik	1'113'000	0
Deutschland	1'545'000	53
Schweiz	1'682'000	79
Guatemala	762'000	15
Kirgistan	1'361'000	0
Nepal	849'000	4

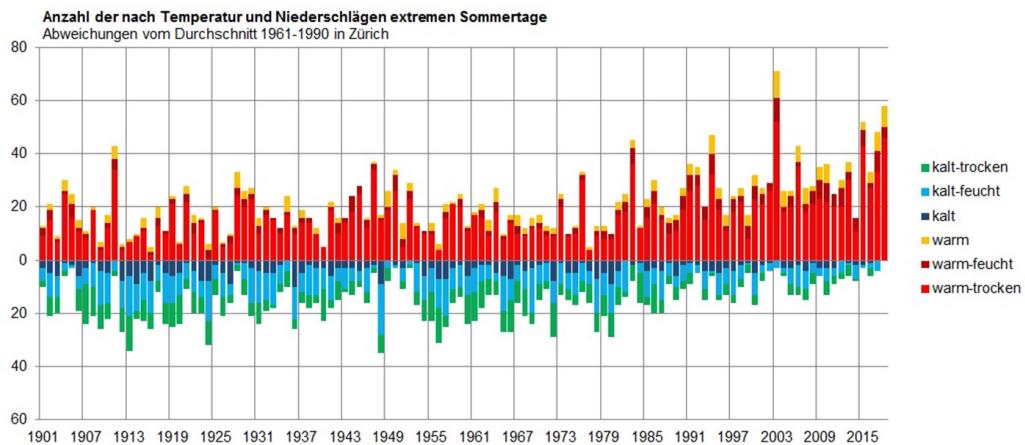
Tabelle 2: Datentabelle zum Wassermengenvergleich

Zur Visualisierung dieses Mengenvergleichs könnten gestaffelte Säulen- oder Balkengrafiken verwendet werden.



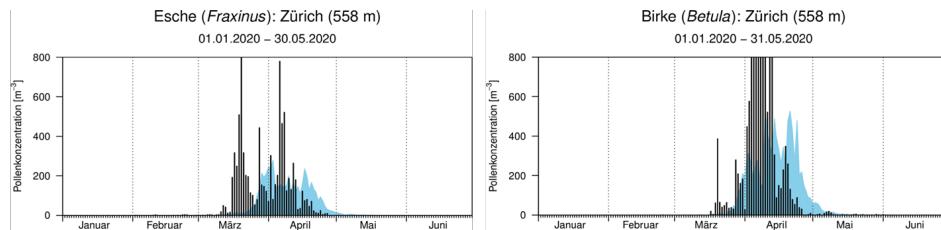
4.5.2 Visualisierung einer Verteilung

Eine Beispielfrage ist hier ob der Sommer heisser wird in der Stadt Zürich. Zur Visualisierung von Verteilungen könnte man mit Histogrammen arbeiten.



4.5.3 Vergleich mehrerer Verteilungen

“Wie stark war die Pollenbelastung von Eschen und Birken in Zürich während der Saison 2020 im Vergleich zum langjährigen Durchschnitt” ist hier die Frage. Zu ihrer Beantwortung wird wieder ein Histogramm verwendet, dieses Mal allerdings in Kombination mit Dichten.

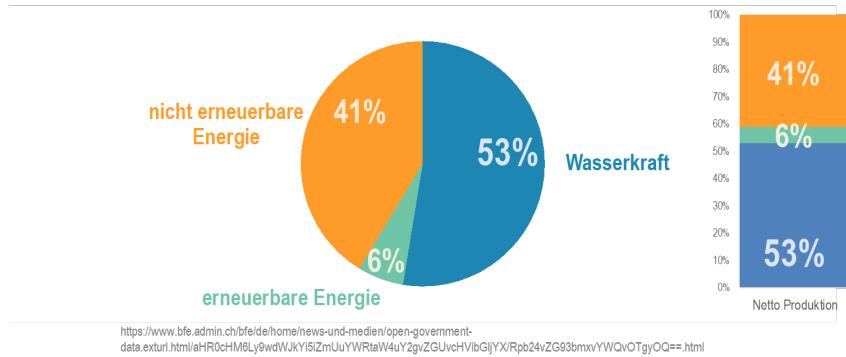


https://www.meteoschweiz.admin.ch/content/dam/meteoswiss/de/Ungebundene-Seiten/Publikationen/Klimabulletin/doc/klimabulletin_fruhling_2020_d.pdf

4.5.4 Visualisierung von Anteilen

Hier geht es um die Frage, wie gross die Anteile A, B und C an Total-X sind. Konkret soll dies zuerst an folgender Frage betrachtet werden: “Wie gross ist der Anteil der Wasserkraft an der gesamten Stromproduktion in der Schweiz?”. Zur Be-

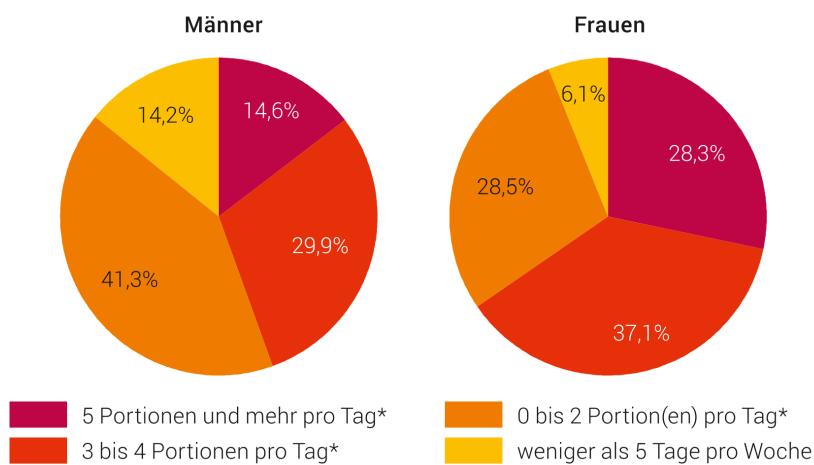
antwortung dieser Frage kann man sowohl Kreissektorendiagramme wie auch gestaffelte Säulen oder Balken verwenden.



Diese Diagrammtypen können aber auch zum Vergleich mehrerer Anteile verwendet werden, was anhand der Frage “Wie gross ist der Anteil der Personen in der Schweiz, die mindestens fünf Portionen Obst oder Gemüse pro Woche konsumieren? Gibt es Geschlechterunterschiede?” veranschaulicht werden soll.

Obst- und Gemüsekonsum, 2017

Bevölkerung ab 15 Jahren in Privathaushalten



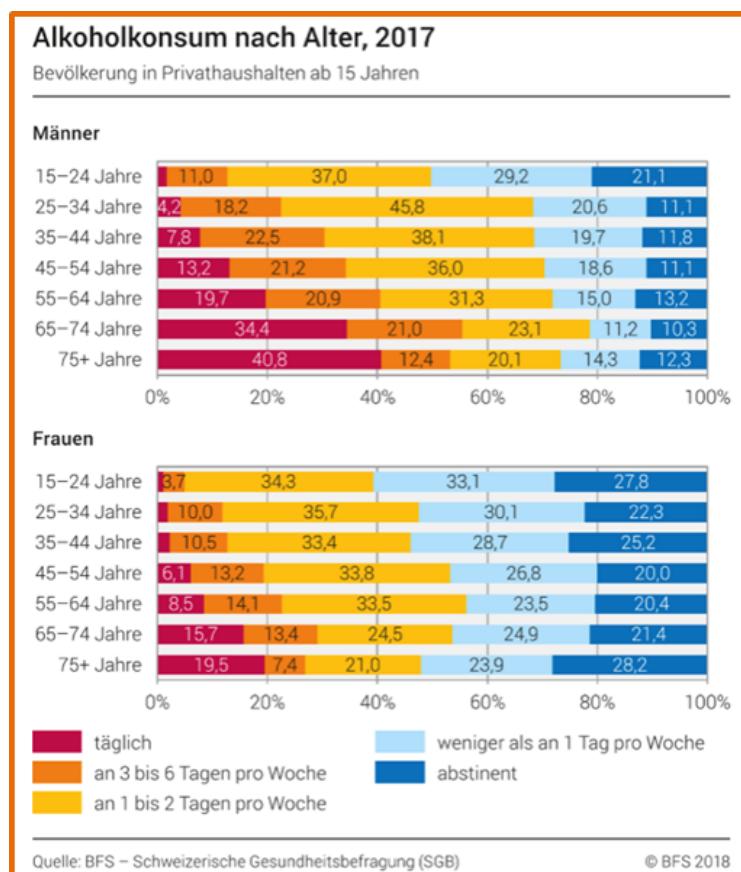
Quelle: BFS – Schweizerische Gesundheitsbefragung (SGB)

© BFS 2018

Beachtenswert ist hier, dass für grosse Anteile an Obst und Gemüse die Farbe Rot verwendet wurde. Diese wird in der westlichen Welt mit Gefahr assoziiert. Die Frage, ob die Entwickler dieser Grafik uns mitteilen wollten, dass viel Obst und Gemüse gesundheitsschädlich ist, bleibt offen.

Ein weiteres Beispiel für die Visualisierung mehrerer Anteile wird anhand der Fra-

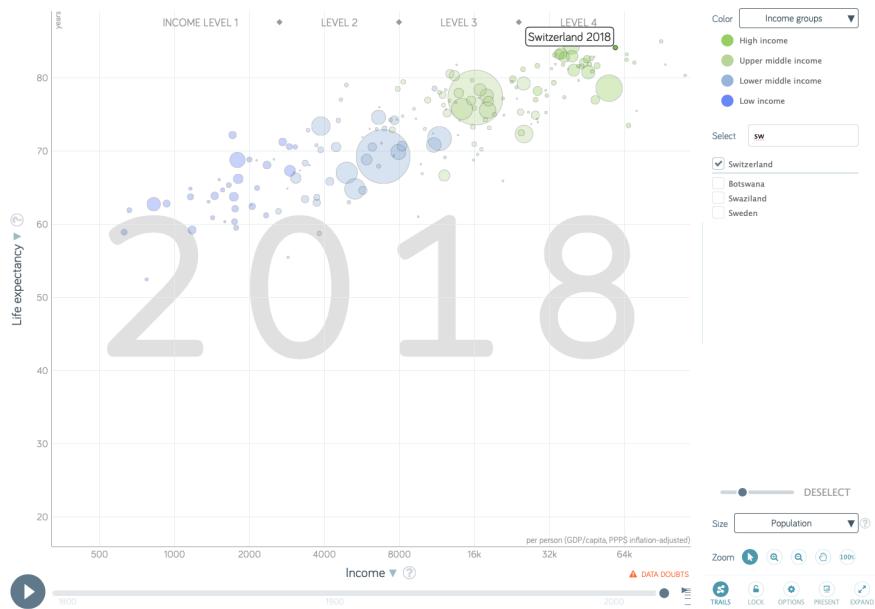
ge "Welche Altersklasse hat den höchsten Anteil am täglichen Alkoholkonsum in der Schweiz? Gibt es Geschlechterunterschiede?" gezeigt.



Hier ist die Farbwahl sehr gelungen. Kategorien mit häufigen Alkoholkonsum, der bekanntlich gesundheitsschädlich ist, sind in Rot-Gelb-Tönen dargestellt. Die gesünderen Kategorien hingegen in Blautönen, die an Wasser erinnern.

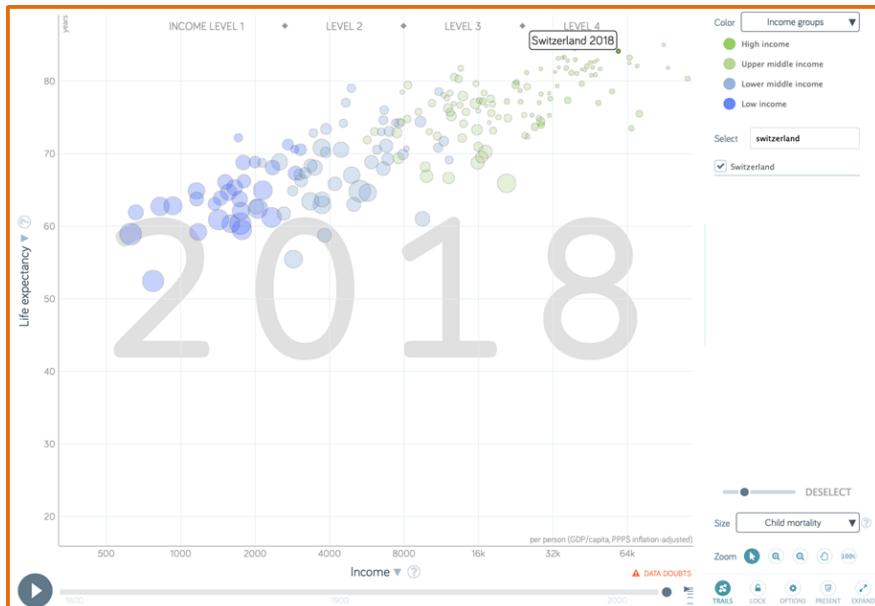
4.5.5 Visualisierung von Zusammenhängen

Hier geht es um die Frage, wie X und Y sich zueinander verhalten. Die einleitende Frage ist folgende: "Besteht ein Zusammenhang zwischen dem Einkommen und der Lebenserwartung in der Weltbevölkerung". Zur Beantwortung dieser Frage verwenden typischerweise Streudiagramme wie das untenstehende verwendet.



In diesem Koordinatensystem zeigt die y-Achse die Lebenserwartung der Bevölkerung eines Staates und die x-Achse stellt das Einkommen dar. Wir sehen sofort, dass reiche Länder eine höhere Lebenserwartung haben. Eine dritte Variable - die Menge der Bevölkerung pro Staat - ist durch die Grösse der Kreise abgebildet.

Eine weiteres Beispiel: "Besteht ein Zusammenhang zwischen dem Einkommen, der Lebenserwartung und der Kindersterblichkeit der Weltbevölkerung?"



<https://www.gapminder.org/>

In dieser Grafik wird die Grösse der Kreise durch die Kindersterblichkeit variiert. Aus dieser Grafik lässt sich erkennen, dass nicht nur die Lebenserwartung mit dem Wohlstand eines Staates steigt, sondern auch die Kindersterblichkeit sinkt.

4.6 Lektion 6: Wie visualisiere ich?

Die Lektion hat die folgenden Lernziele:

- Sie wissen welche Diagrammkomponenten bearbeitet werden sollten, um die Lesbarkeit der Visualisierung zu erhöhen.
- Sie kennen die graphischen Variablen, um Diagrammtypen attraktiv und gut lesbar zu gestalten.
- Sie können diverse graphische Lösungen vorschlagen, um Mengen, Verteilungen, Anteile oder Zusammenhänge prägnant darzustellen.

Für jeden Diagrammtyp existieren zahlreiche Möglichkeiten, die Grundform zu verändern. Sie sollten sich nie auf die Standardeinstellungen einer Grafik-Software verlassen, egal ob es sich dabei um R oder Excel handelt. Automatisch erstellte Grafiken müssen eigentlich immer nachbearbeitet werden.

4.6.1 Verbesserungsmöglichkeiten für Mengendiagramme

Im Verlauf des Kursbausteins wurden bereits zwei Alternativen zu Säulen gezeigt: Skalierter Text und skalierte Füsse. Die vertikale Säulen können aber auch horizontalen Balken geändert werden. Dies macht dann Sinn, wenn die Beschriftungen der Säulen zu viel Platz einnehmen. Bei den Balken fallen ausserdem die horizontale Längenvergleiche leichter. Zudem wurden die Zwischenräume der Balken verkleinert, um die Vergleiche zu vereinfachen. Drittens wurde das rote Farbschema der Säulen durch Blautönen in den Balken ersetzt, da Wasser eher mit blau als mit rot assoziiert wird. So wird auch der Kontrast zu den in Orange hervorgehobenen Werten erhöht. Die Reihenfolge der Länder wurde neu ausserdem nach der Länge der dunkelblauen Balken, die die Menge des nicht-importierten Wassers darstellen, sortiert.

Standardmäßig wird in vielen Statistikpaketen automatisch eine Legendenbox ausserhalb des Diagramms generiert. Bei beiden Grafiken wurde hier darauf

verzichtet, da es sich lohnt, die Information wann immer möglich in die Grafik zu integrieren. Ein wichtiges Element einer Grafik ist der Titel. Dieser sollte die Betrachtenden auf die wichtige Information im Diagramm aufmerksam machen oder prägnant die Antwort auf die gestellte Frage liefern.

4.6.2 Verbesserungsmöglichkeiten für Verteilungsdiagramme

Histogramme zeigen Häufigkeiten, also Daten, die in Klassen eingeteilt sind. Diese Klassen können einen Wert oder Wertebereich darstellen. Im Gegensatz zu den Säulendiagrammen sollten die Säulen bei kontinuierlichen Wertebereichen keine Abstände aufweisen. Alle Verbesserungsmöglichkeiten, die bereits für Säulen- und Balkendiagramme genannt wurden, gelten auch für Histogramme.

Im Beispiel der Grafiken zur Altersstruktur sollte der Titel durch den Folientitel “Die Schweizer Bevölkerung wird immer älter” ersetzt werden, die dieser die Kernaussage der Abbildung wiedergibt. Wichtig sind auch klare Kontraste zwischen dem Hintergrund und Vordergrund einer Visualisierung.

4.6.3 Verbesserungsmöglichkeiten für Anteildiagramme

Excel ermöglicht das Erstellen von 3D-Kreissektorendiagrammen. Darauf sollte unbedingt verzichtet werden, wenn die Daten nicht 3D sind. Ein Hauptkritikpunkt an Kreissektorendiagrammen im Allgemein ist, dass ihre Winkel weniger gut geschätzt werden können als die Längen von Säulen- oder Balkendiagrammen. Dies wird in 3D durch die perspektivische Verzerrung noch verstärkt. Außerdem sollten nicht mehr als 2-4 Sektoren in einem solchen Diagramm dargestellt werden, da zu kleine “Tortenstücke” schwierig zu lesen sind. Sektoren sollten immer bei 0° bzw. auf 12 Uhr beginnen und im Uhrzeigersinn gelesen werden können. Zudem gelten die bisher gemachten Verbesserungsvorschläge natürlich auch hier.

4.6.4 Verbesserungsmöglichkeiten für Diagramme, die Zusammenhänge zeigen

Vergleicht man die beiden Streudiagramme fällt auf, dass der Titel geändert, wichtige historische Daten hinzugefügt, sowie die Grösse und Farbe der Datenpunkte und der Schrift geändert wurden. Auch das Layout der Achsen wurde verändert und das Koordinatengitter entfernt. Mit dem vertikalen Strich können letztlich nun auch positive und negativen Werte besser unterschieden werden.

Referenzen

- Kanizsa, G., 1955. Margini quasi-percettivi in campi con stimulazione omogenea, Rivista di Psicologia, 49 (1). 7-30.

© Digital Society Initiative

4.7 Folien



**Universität
Zürich**^{UZH}

Digital Society Initiative

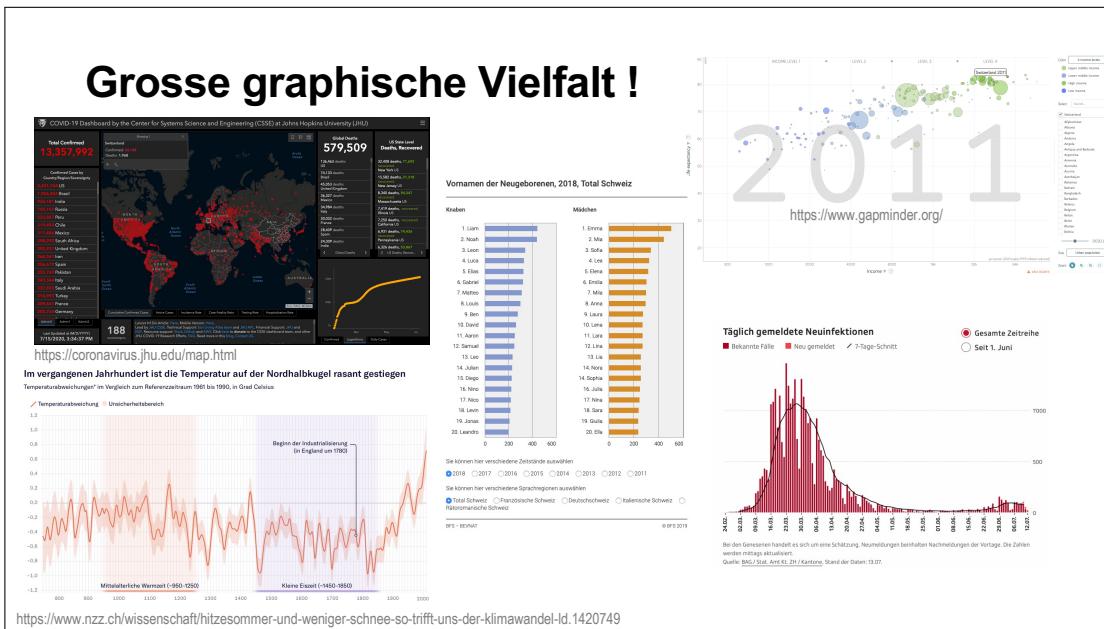


Studium Digitale

Kursbaustein 7: Effektive Informationsvisualisierung

Lektion 1: Einstieg und Überblick





Lernziele

- Sie erkennen, wann es sich lohnt, ihre Zeit in eine Visualisierung zu investieren
- Sie kennen diverse Visualisierungsmethoden und können diese zieltgerecht einsetzen.
- Sie können die für Ihre Aufgabe und Daten angemessene Visualisierungsmethode auswählen und anwenden
- Sie können selbständig attraktive Visualisierungen mit Microsoft Excel erstellen



Überblick

1. Lernziele
2. Struktur: Die **5 Ws**

Die **5 Ws** der Visualisierung

Warum | Was | für Wen | Wozu und Wie wird visualisiert?



Warum visualisiere ich?

Erkenntnisgewinn aus
Datenhaufen

(07.12.17) Die Schweizer Wasserversorgungen stellten 2016 923 Mio. m³ Wasser bereit. Das sind gut 1% weniger als im Vorjahr. Der Pro-Kopf-Verbrauch sank von 307 auf 299 Liter pro Tag und liegt nach dem Trockenjahr 2015 nun wieder etwa auf dem Niveau von 2014. Damit setzt sich der rückläufige Verbrauchstrend fort, der in den Achtzigerjahren begann. Das geht aus der soeben fertiggestellten Wasserstatistik des Branchenverbands SVGW hervor. Eine Imageumfrage des Verbands ergab zudem, dass die Bevölkerung das Schweizer Trinkwasser weiterhin als sehr gut einstuft.

Bewegte sich in den 70er-Jahren der mittlere Tagesverbrauch pro Person noch an der 500-Liter-Marke, sank er seither stetig auf heute rund 300 Liter pro Tag. Auch die gesamte Wasserabgabe sank kontinuierlich und lag 2016 bei 923 Mio. m³, 1% weniger als 2015. Das entspricht rund dem 1.5-fachen Volumen des Murtensees. Hauptursache des markanten Mehrverbrauchs 2015 dürfte die ausgeprägte Trockenphase während der ganzen zweiten Jahreshälfte 2015 gewesen sein. Dafür spricht auch die gegenüber 2014 und 2016 um fast 20% höhere maximale Tagesabgabe.

Kosten seit den Achtzigerjahren um 7 Prozent gestiegen

Aus der neuen Statistik geht auch hervor, dass sich die Betriebs- und Kapitalkosten 2016 auf rund 1.6 Mrd. Fr. oder 188 Fr. pro Einwohner beliefen. Teuerungsbereinigt sind die Betriebs- und Kapitalkosten 7% höher als 1980. Die jährlichen Investitionen in den Wasserversorgungen betrugen 2016 925 Mio. Fr. oder 110 Fr. pro Einwohner. Teuerungsbereinigt werden damit pro Einwohner 7% mehr investiert als noch 1980. In den Wasserversorgungen arbeiteten 2016 rund 6500 Personen, deren Pensum rund 3100 Vollzeitstellen entspricht. Neben der Bereitstellung von Wasser produzierten die Schweizer Wasserversorger 2016 auch 186 GWh Strom, überwiegend mittels Trinkwasserturbinen. Damit deckte die Branche fast die Hälfte ihres Stromverbrauchs aus eigener Produktion.

Quelle:

http://www.svgw.ch/index.php?id=267&no_cache=1&tx_ttnews%5Btt_news%5D=1716&cHash=4b8c803c421031222b5901f65cccd5413/index.php?id=267&no_cache=1&tx_ttnews%5Btt_news%5D=1716&cHash=4b8c803c421031222b5901f65cccd5413



(07.12.17) Die Schweizer Wasserversorgungen stellten 2016 923 Mio. m³ Wasser bereit. Das sind gut 1% weniger als im Vorjahr. Der Pro-Kopf-Verbrauch sank von 307 auf 299 Liter pro Tag und liegt nach dem Trockenjahr 2015 nun wieder etwa auf dem Niveau von 2014. Damit setzt sich der rückläufige Verbrauchstrend fort, der in den Achtzigerjahren begann. Das geht aus der soeben fertiggestellten Wasserstatistik des Branchenverbands SVGW hervor. Eine Imageumfrage des Verbands ergab zudem, dass die Bevölkerung das Schweizer Trinkwasser weiterhin als sehr gut einstuft.

Bewegte sich in den 70er-Jahren der mittlere Tagesverbrauch pro Person noch an der 500-Liter-Marke, sank er seither stetig auf heute rund 300 Liter pro Tag. Auch die gesamte Wasserabgabe sank kontinuierlich und lag 2016 bei 923 Mio. m³, 1% weniger als 2015. Das entspricht rund dem 1.5-fachen Volumen des Murtensees. Hauptursache des markanten Mehrverbrauchs 2015 dürfte die ausgeprägte Trockenphase während der ganzen zweiten Jahreshälfte 2015 gewesen sein. Dafür spricht auch die gegenüber 2014 und 2016 um fast 20% höhere maximale Tagesabgabe.

Kosten seit den Achtzigerjahren um 7 Prozent gestiegen

Aus der neuen Statistik geht auch hervor, dass sich die Betriebs- und Kapitalkosten 2016 auf rund 1.6 Mrd. Fr. oder 188 Fr. pro Einwohner beliefen. Teuerungsbereinigt sind die Betriebs- und Kapitalkosten 7% höher als 1980. Die jährlichen Investitionen in den Wasserversorgungen betrugen 2016 925 Mio. Fr. oder 110 Fr. pro Einwohner. Teuerungsbereinigt werden damit pro Einwohner 7% mehr investiert als noch 1980. In den Wasserversorgungen arbeiteten 2016 rund 6500 Personen, deren Pensum rund 3100 Vollzeitstellen entspricht. Neben der Bereitstellung von Wasser produzierten die Schweizer Wasserversorger 2016 auch 186 GWh Strom, überwiegend mittels Trinkwasserturbinen. Damit deckte die Branche fast die Hälfte ihres Stromverbrauchs aus eigener Produktion.

Quelle:

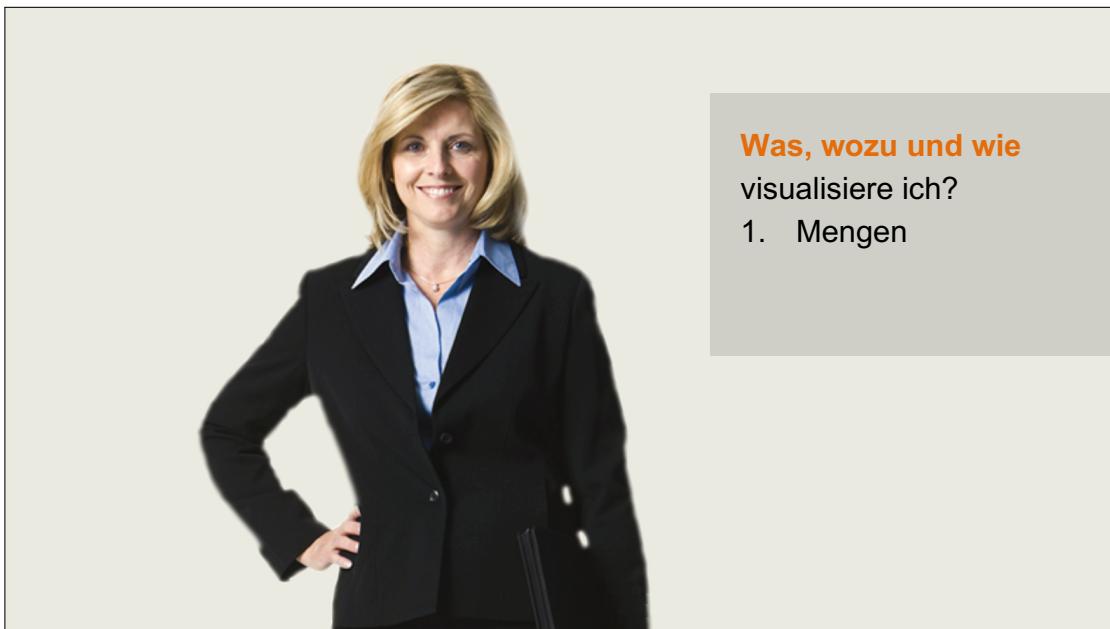
http://www.svgw.ch/index.php?id=267&no_cache=1&tx_ttnews%5Btt_news%5D=1716&cHash=4b8c803c421031222b5901f65cccd5413/index.php?id=267&no_cache=1&tx_ttnews%5Btt_news%5D=1716&cHash=4b8c803c421031222b5901f65cccd5413

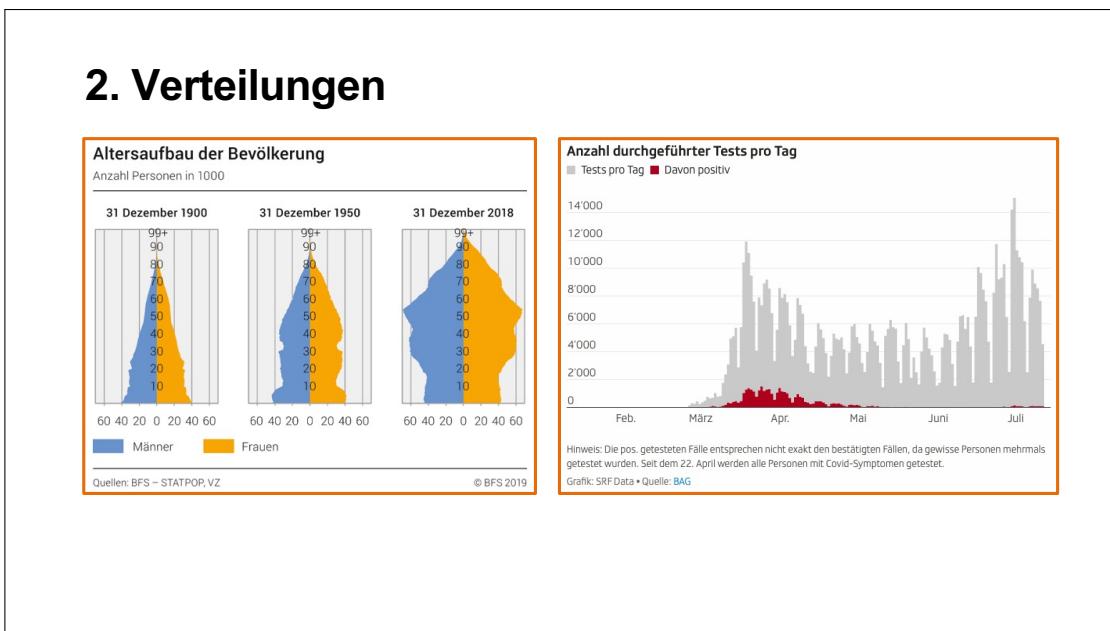


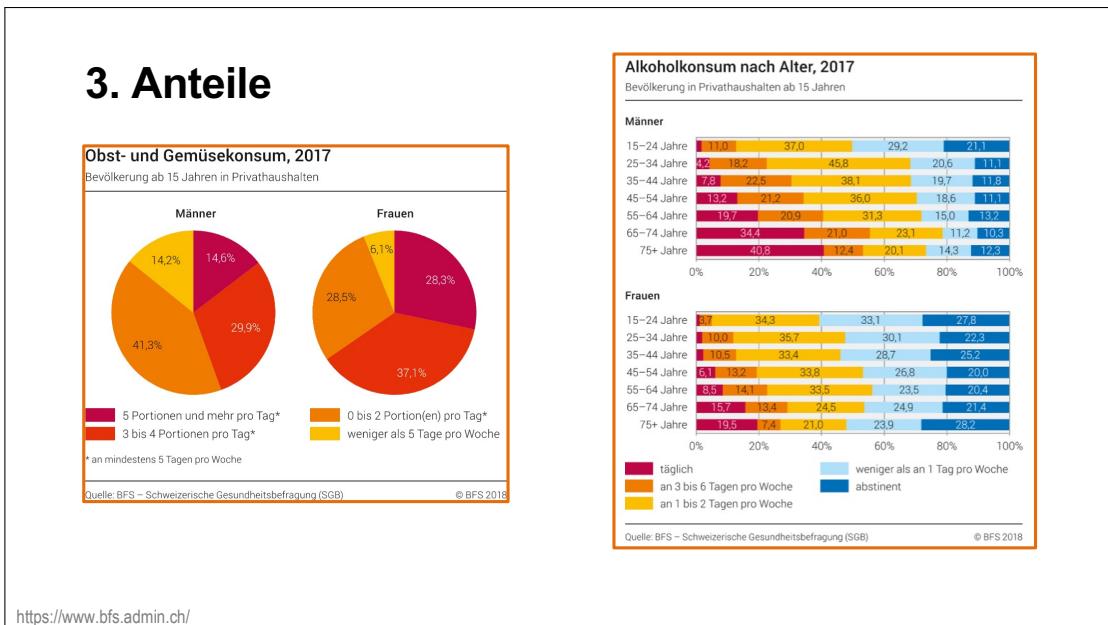
Für **wen** visualisiere ich?

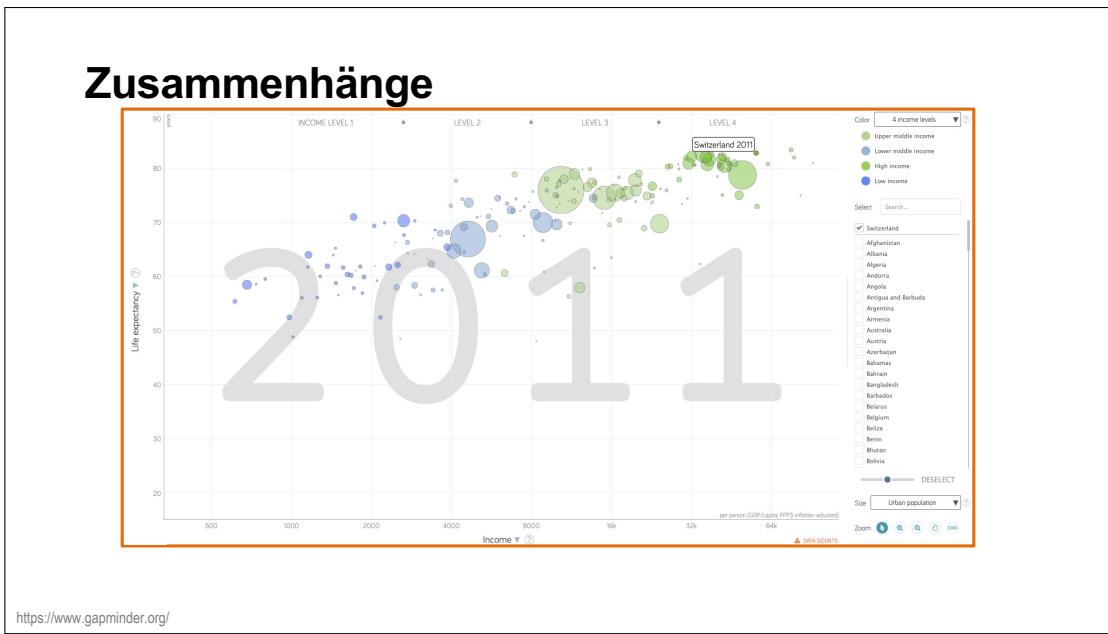
Wahrnehmungseffekte

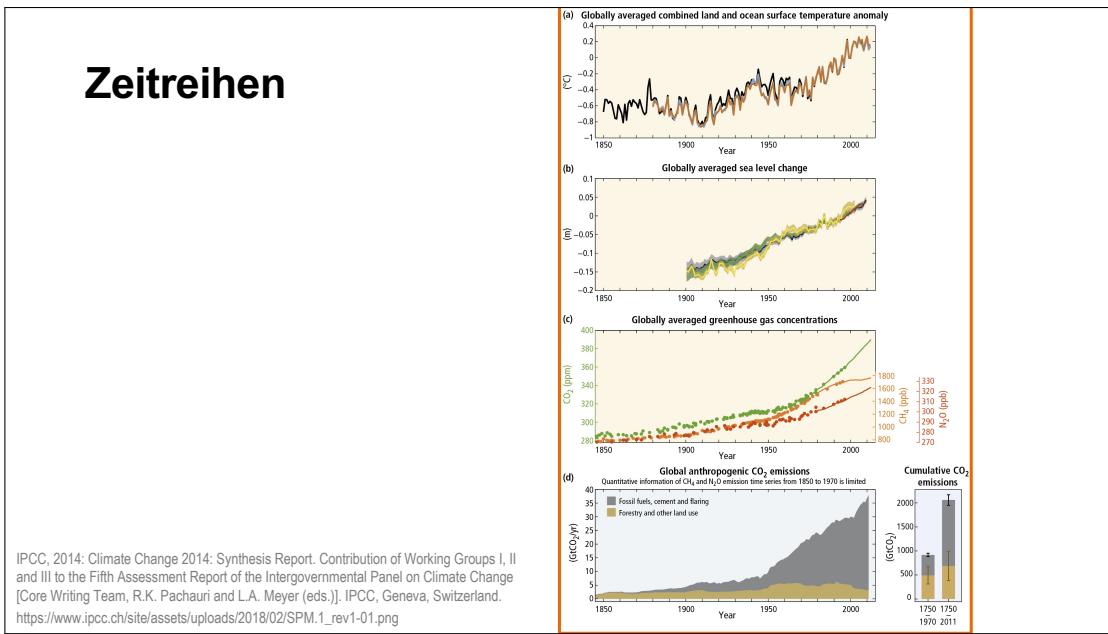
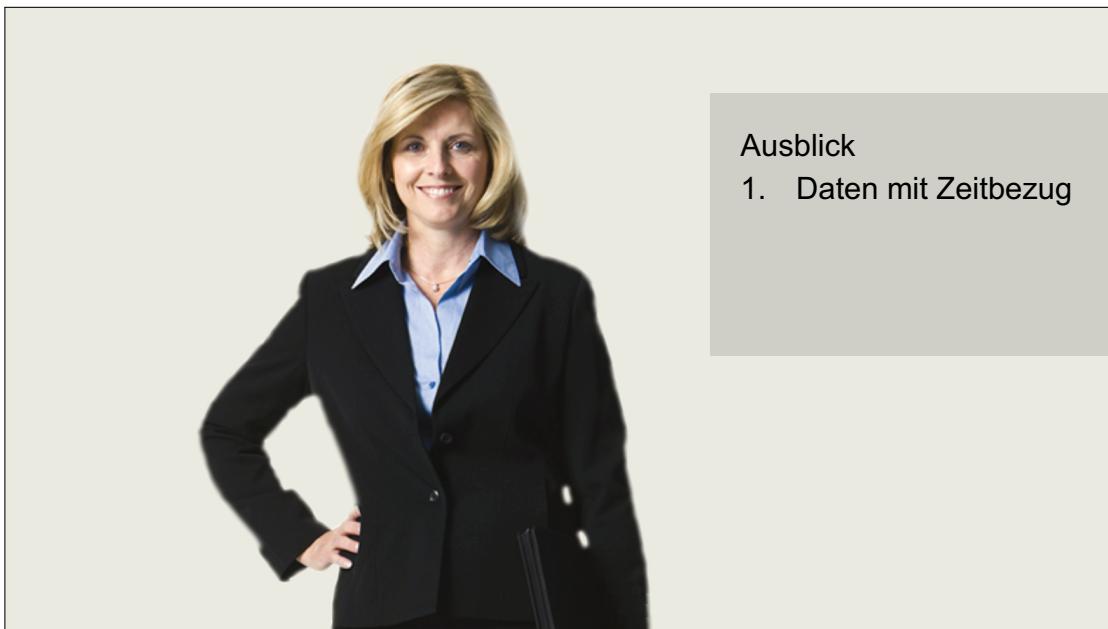


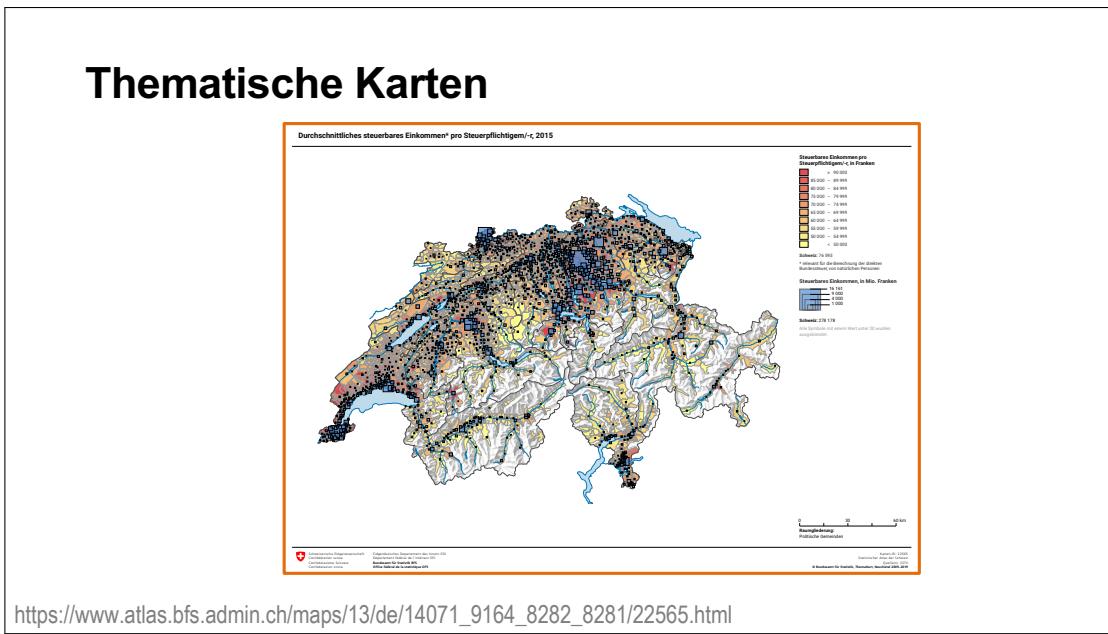
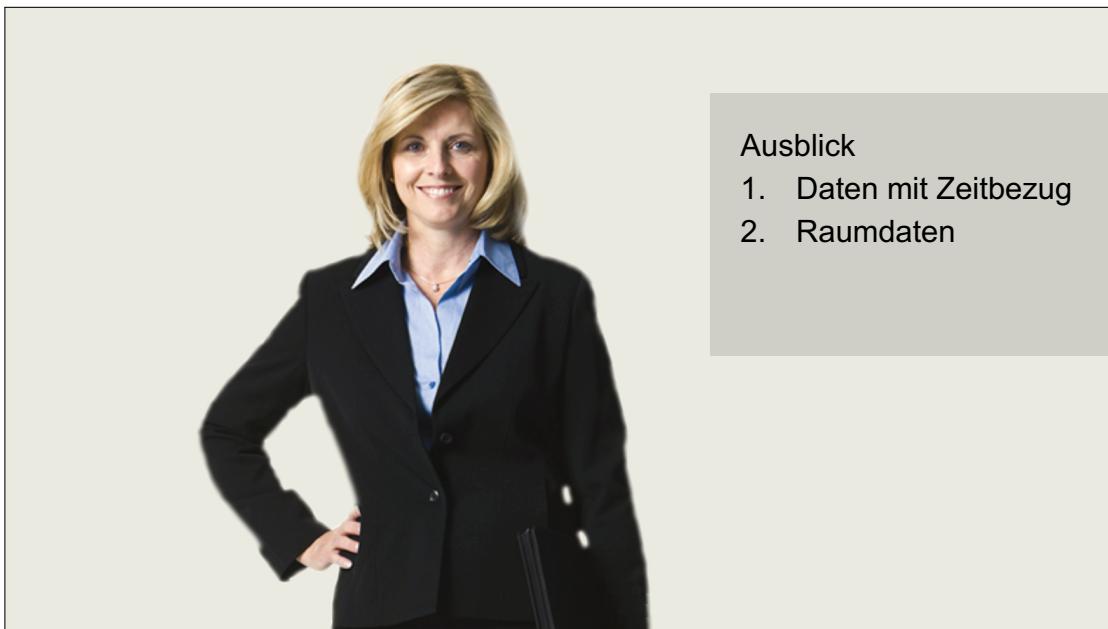












Hilfreiche Bücher über Datenvisualisierung

Claus Wilke

<https://serialmentor.com/dataviz/>

Stephanie Evergreen

<https://stephanieevergreen.com/books/>

Referenzen

Hilfreiche Bücher:

<https://serialmentor.com/dataviz/>

<https://stephanieevergreen.com/books/>

<https://www.amazon.com/gp/product/1492031089>

<https://www.amazon.de/Graphische-Semiolegric-Diagramme-Netze-Karten/dp/3110036606>

https://www.amazon.com/The-Elements-of-Graphing-Data/dp/0963488414/ref=cm_cr_dp_d_rvw_txt?ie=UTF8

<https://www.amazon.com/Visualization-Analysis-Design-AK-Peters/dp/1466508914>

<https://www.amazon.com/Design-Information-Introduction-Histories-Visualizations/dp/1592538061>

Weblinks

Hilfreiche Internetseiten:

<http://albertocairo.com>

<https://www.datawrapper.de>

http://www.visual-literacy.org/periodic_table/periodic_table.html

<https://flowingdata.com/2018/10/17/ask-the-question-visualize-the-answer/>

Referentin

Prof. Sara Irina Fabrikant

Geographisches Institut

© Universität Zürich

Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 7: Effektive Informationsvisualisierung

Lektion 2: Warum will ich visualisieren?

Lernziele

- Sie können informiert entscheiden, wann es sich lohnt, Daten zu visualisieren.
- Sie können selbständig eine statistische Aussage mit einem Bild visualisieren.
- Sie können mit einem Beispiel illustrieren, warum die Visualisierung von Daten wichtig und zielführend ist.

Die 5 Ws in Visualisierung

Warum | Was | für Wen | Wozu und Wie wird visualisiert?



Warum visualisiere ich?

1. Kommunikation
eines Sachverhalts

Der Wasserverbrauch in der Schweiz sinkt weiter

(07.12.17) Die Schweizer Wasserversorgungen stellten 2016 923 Mio. m³ Wasser bereit. Das sind gut 1% weniger als im Vorjahr. Der Pro-Kopf-Verbrauch sank von 307 auf 299 Liter pro Tag und liegt nach dem Trockenjahr 2015 nun wieder etwa auf dem Niveau von 2014. Damit setzt sich der rückläufige Verbrauchstrend fort, der in den Achtzigerjahren begann. Das geht aus der soeben fertiggestellten Wasserstatistik des Branchenverbands SVGW hervor. Eine Imageumfrage des Verbands ergab zudem, dass die Bevölkerung das Schweizer Trinkwasser weiterhin als sehr gut einstuft.

Bewegte sich in den 70er-Jahren der mittlere Tagesverbrauch pro Person noch an der 500-Liter-Marke, sank er seither stetig auf heute rund 300 Liter pro Tag. Auch die gesamte Wasserabgabe sank kontinuierlich und lag 2016 bei 923 Mio. m³, 1% weniger als 2015. Das entspricht rund dem 1.5-fachen Volumen des Murtensees. Hauptursache des markanten Mehrverbrauchs 2015 dürfte die ausgeprägte Trockenphase während der ganzen zweiten Jahreshälfte 2015 gewesen sein. Dafür spricht auch die gegenüber 2014 und 2016 um fast 20% höhere maximale Tagesabgabe.

Kosten seit den Achtzigerjahren um 7 Prozent gestiegen

Aus der neuen Statistik geht auch hervor, dass sich die Betriebs- und Kapitalkosten 2016 auf rund 1.6 Mrd. Fr. oder 188 Fr. pro Einwohner beliefen. Teuerungsbereinigt sind die Betriebs- und Kapitalkosten 7% höher als 1980. Die jährlichen Investitionen in den Wasserversorgungen betrugen 2016 925 Mio. Fr. oder 110 Fr. pro Einwohner. Teuerungsbereinigt werden damit pro Einwohner 7% mehr investiert als noch 1980.

In den Wasserversorgungen arbeiteten 2016 rund 6500 Personen, deren Pensum rund 3100 Vollzeitstellen entspricht. Neben der Bereitstellung von Wasser produzierten die Schweizer Wasserversorger 2016 auch 186 GWh Strom, überwiegend mittels Trinkwasserturbinen. Damit deckte die Branche fast die Hälfte ihres Stromverbrauchs aus eigener Produktion.

Quelle: http://www.svgw.ch/index.php?id=267&no_cache=1&tx_ttnews%5Btt_news%5D=1716&cHash=4b8c803c421031222b5901f65cccd5413/index.php?id=267&no_cache=1&tx_ttnews%5Btt_news%5D=1716&cHash=4b8c803c421031222b5901f65cccd5413

Kommunikation eines Sachverhalts

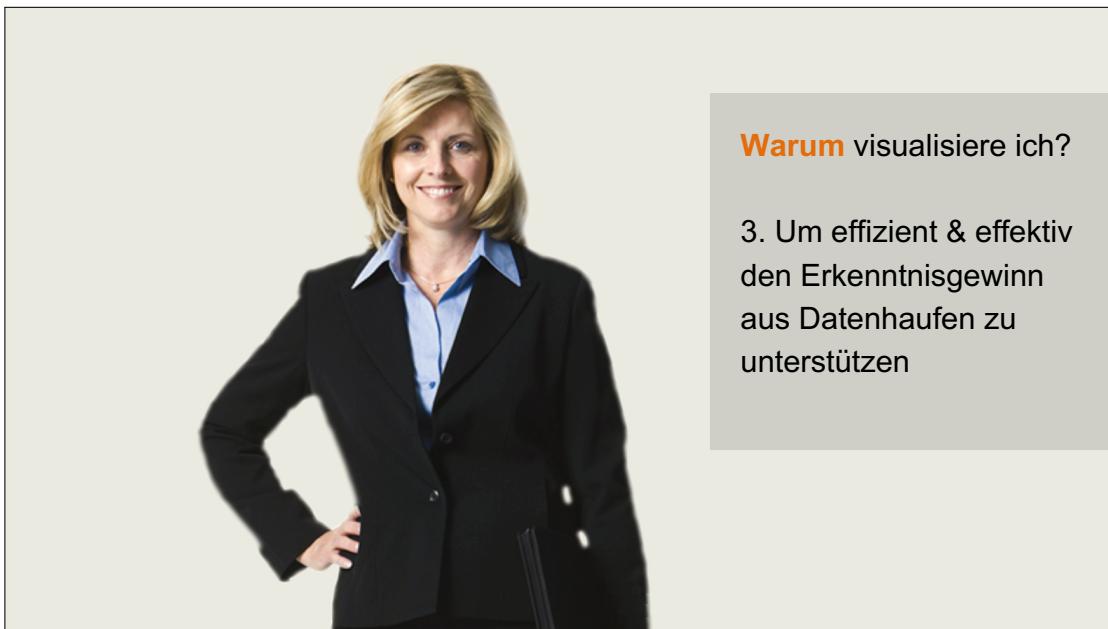
Wie hoch ist der mittlere Tagesverbrauch von Wasser pro Person in der Schweiz?



Warum visualisiere ich?

2. Wenn **ein Bild** mehr
als tausend Wörter sagt





Vergleich von vier überschaubaren Datensätzen

Set A		Set B		Set C		Set D	
X	Y	X	Y	X	Y	X	Y
10	8	10	9	10	7	8	7
8	7	8	8	8	7	8	6
13	8	13	9	13	13	8	8
9	9	9	9	9	7	8	9
11	8	11	9	11	8	8	8
14	10	14	8	14	9	8	7
6	7	6	6	6	6	8	5
4	4	4	3	4	5	19	13
12	11	12	9	12	8	8	6
7	5	7	7	7	6	8	8
5	6	5	5	5	6	8	7

[Anscombe 1973]

Anscombe, F. J. (1973). Graphs in Statistical Analysis. American Statistician. 27 (1): 17–21. doi:10.1080/00031305.1973.10478966.

Was steckt im Datenhaufen?

Welche Information bzw. Muster ist in den vier Datensätzen verborgen?

Vier Datensätze: identisches statistisches Muster !

Set A		Set B		Set C		Set D	
X	Y	X	Y	X	Y	X	Y
10	8	10	9	10	7	8	7
8	7	8	8	8	7	8	6
13	8	13	9	13	13	8	8
9	9	9	9	9	7	8	9
11	8	11	9	11	8	8	8
14	10	14	8	14	9	8	7
6	7	6	6	6	6	8	5
4	4	4	3	4	5	19	13
12	11	12	9	12	8	8	6
7	5	7	7	7	6	8	8
5	6	5	5	5	6	8	7

Mittelwert und Streuung

$$\mu_X = 9.0 \quad \sigma_X = 3.317$$

$$\mu_Y = 7.5 \quad \sigma_Y = 2.03$$

Linerarer Zusammenhang

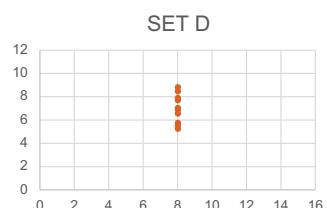
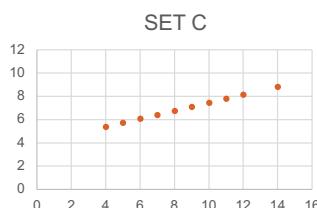
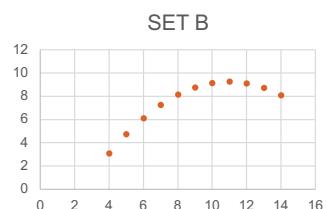
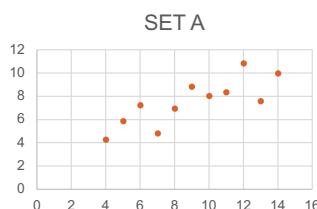
$$Y = 3 + 0.5X$$

$$R^2 = 0.67$$

[Anscombe 1973]

Anscombe, F. J. (1973). Graphs in Statistical Analysis. American Statistician. 27 (1): 17–21. doi:10.1080/00031305.1973.10478966.

Vier Datensätze: unterschiedliche graphische Muster !

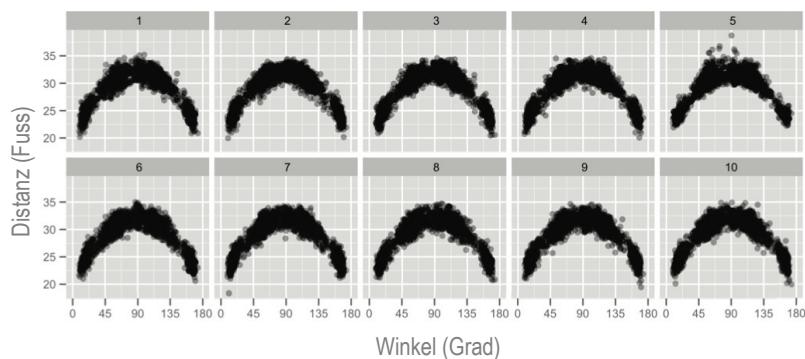


Warum visualisiere ich?

3. Um effizient & effektiv die Mustererkennung in Datenhaufen zu erleichtern.

Vergleich von 10 visualisierten Datensätzen:

Nullhypothese: Es besteht ein quadratischer Zusammenhang zwischen Wurfdistanz und Winkel zum Basketballkorb bei Drei-Punkte Würfen des Los Angeles Lakers Basketball Teams.

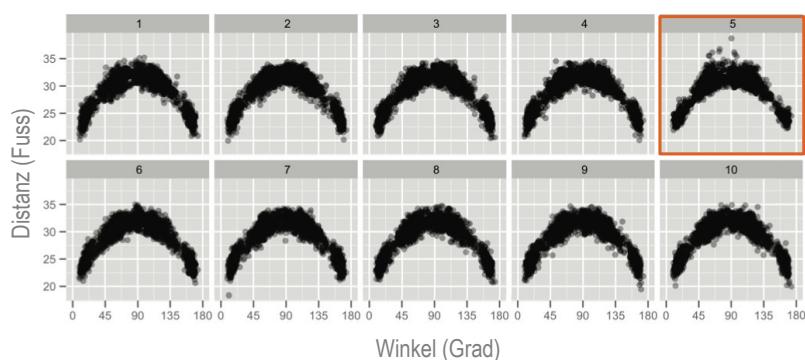


[Wickham et al. 2010]

Wickham, H., Cook, D., Hofmann, H., Buja, A. (2010). Graphical Inference for Infovis. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6: 973-979.

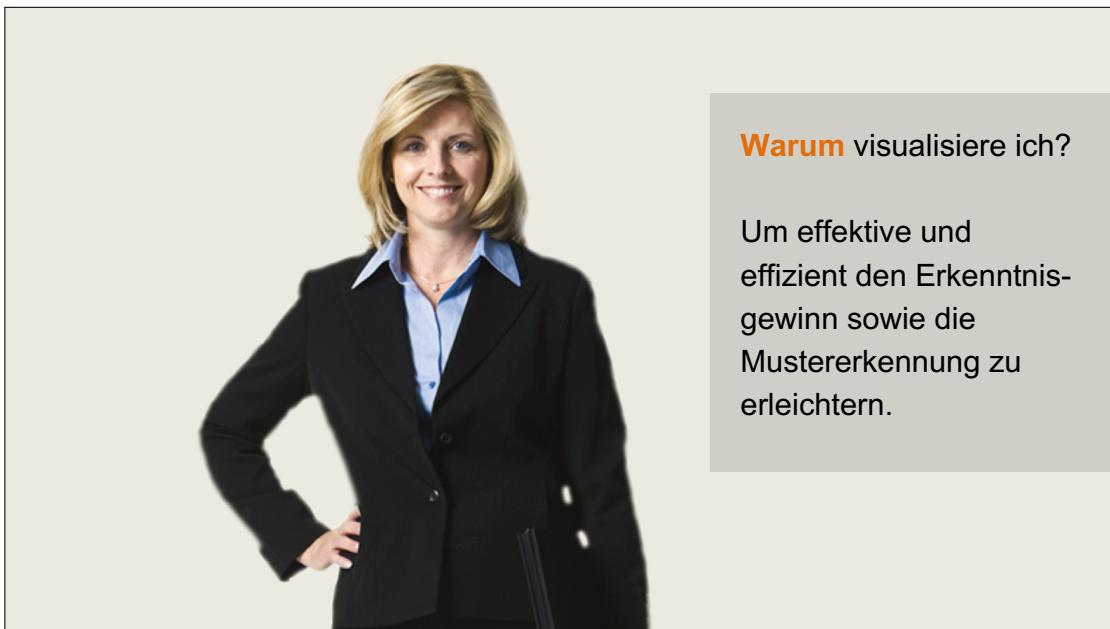
Vergleich von 10 visualisierten Datensätzen:

Nullhypothese: Es besteht ein quadratischer Zusammenhang zwischen Wurfdistanz und Winkel zum Basketballkorb bei Drei-Punkte Würfen des Los Angeles Lakers Basketball Teams.



[Wickham et al. 2010]

Wickham, H., Cook, D., Hofmann, H., Buja, A. (2010). Graphical Inference for Infovis. IEEE Transactions on Visualization and Computer Graphics, vol. 16, no. 6: 973-979.

**Warum** visualisiere ich?

Um effektive und effizient den Erkenntnisgewinn sowie die Mustererkennung zu erleichtern.

Referenzen

- Anscombe, F. J. (1973). Graphs in Statistical Analysis. *American Statistician*. 27 (1): 17–21. doi:10.1080/00031305.1973.10478966.
- Evergreen, Stephanie (2020). Effective Data Visualization. Second Edition, Sage, Thousand Oaks, (CA), U.S.A.
- Wickham, H., Cook, D., Hofmann, H.. Buja, A. (2010). Graphical Inference for Infovis. *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6: 973-979.
- Wilke, Claus (2019). Fundamentals of Data Visualization. O'Reilly Media, Inc., Sebastopol, (CA), U.S.A.

Weblinks

<https://serialmentor.com/dataviz/>

<https://viz.wtf>

<http://www.svgw.ch>

<https://www.autodeskresearch.com/publications/samestats>

Referentin

Prof. Sara Irina Fabrikant

Geographisches Institut

© Universität Zürich

Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 7: Effektive Informationsvisualisierung

Lektion 3: Was will ich visualisieren?

Lernziele

- Sie kennen zentrale Dateneigenschaften (Datentyp, Datenmerkmale, Messniveau, etc.). Siehe dazu auch den Kursbaustein «Statistik».
- Sie kennen die graphischen Variablen für die Visualisierung
- Sie haben einen Überblick über das Visualisierungsinventar

Die 5 Ws in Visualisierung

Warum | Was | für Wen | Wozu und Wie wird visualisiert?



Was visualisiere ich ?

1. Datentyp

wrapped by an almost perfectly symmetrical
volcano. The ceaseless beat of minute waves
the rocks I sit on, make me increasingly
worried and ~~sad~~^{relaxed} I saw my dreams today...
Dying hills, covered with small yellow
flowers, cut off abruptly, as knife in
a sea. I am in awe. Looking off
to the distance... snow capped mts
to the sky, beautifully rolling and smoothly
formed clouds. Vaguely appearing like
monsters in ~~the~~ ^{the} night. The sun
is still out, and is beginning to set. The
now on the Orosio Volcano turns orange, and
the sky on the opposite horizon. As
I walk back through the Plaza, I see the
bus must be the hang-out area for the
young people at night. At the same time it
is where the older generation takes its walks.
I take a deep breath, and continue walking.

<https://pixabay.com/illustrations/pay-digit-number-fill-count-mass-1036469/>



<https://pixabay.com/illustrations/pay-digit-number-fill-count-mass-1036469/>



Was visualisiere ich?

1. Datentyp
2. Datenmerkmale



Welche Daten kann ich visualisieren?

Typ	Merkmal	Wert	Messniveau	Beschreibung
Numerisch	kontinuierlich	1.3, 10 ²	ratio	Distanz in Metern
Numerisch	diskret	1, 10, -25	interval	Temperatur in Grad Celsius
alphabetisch	diskret	niedrig, mittel, hoch	ordinal	Erfahrung in Visualisierung
alphabetisch	diskret	Rot, Grün, Blau	nominal	Farbton
Alpha-numerisch	Datum/Zeit	1. August 1291	Interval/ratio	Nationalfeiertag

Siehe Kurseinheit: "Statistik"

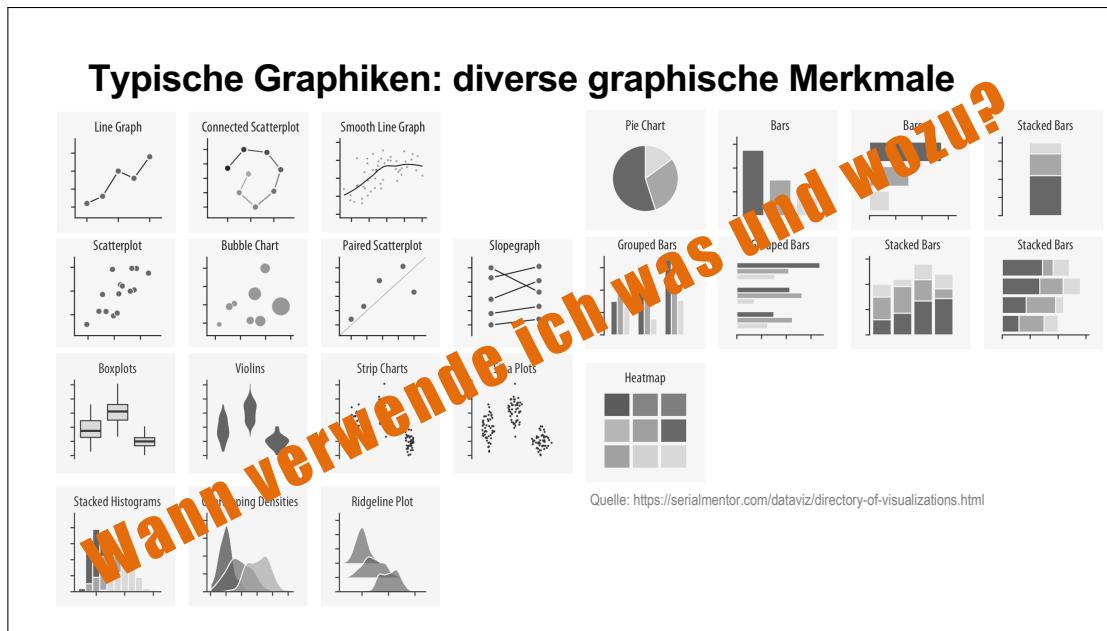
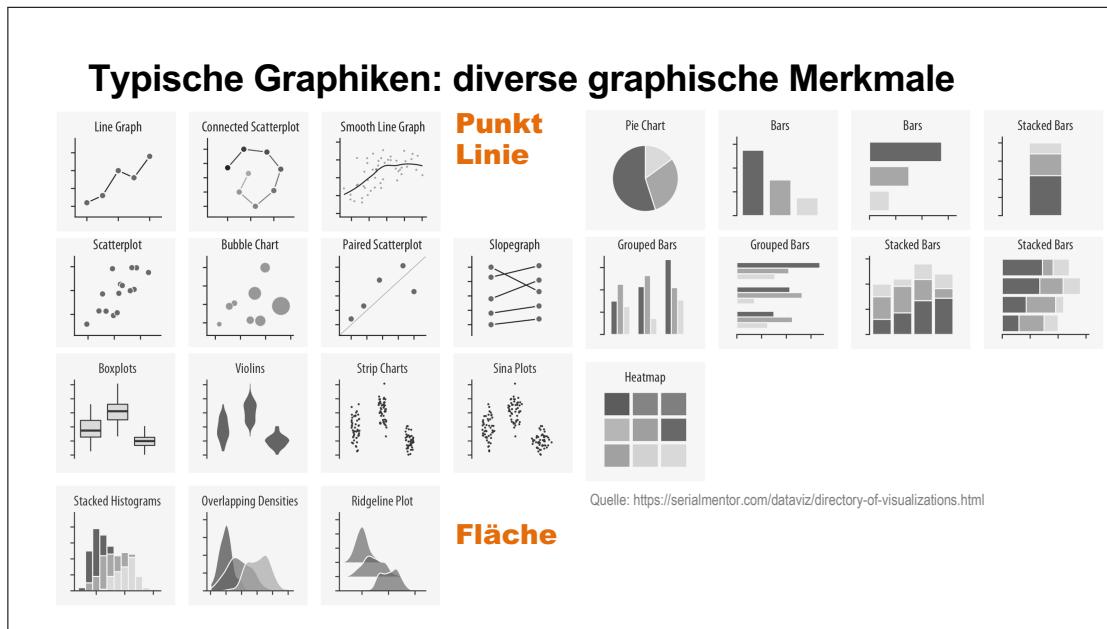
Eine typische Datentabelle

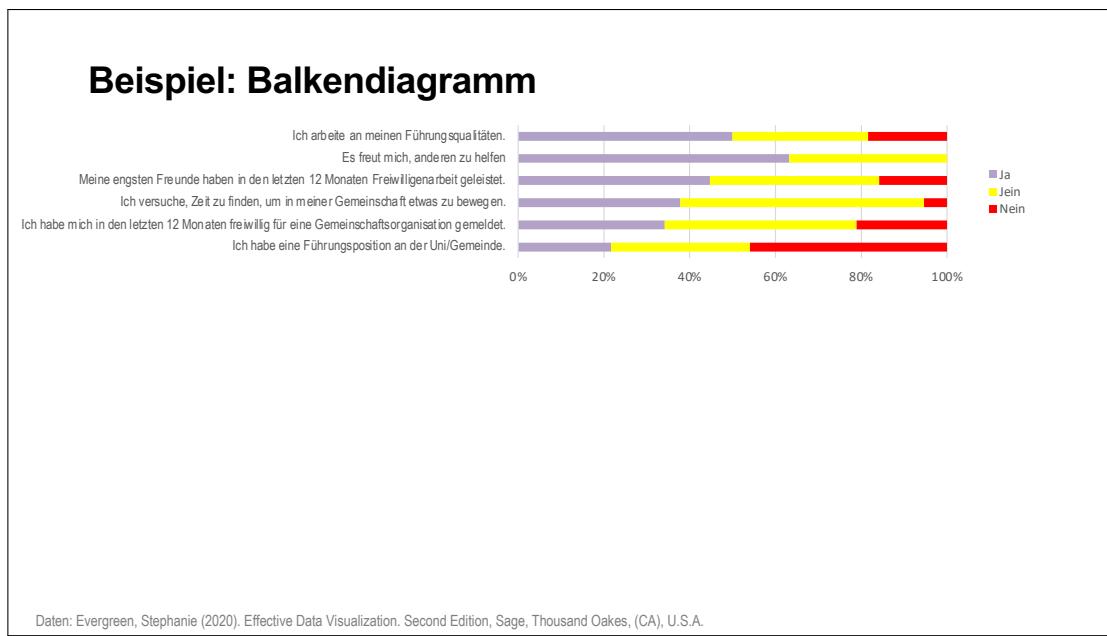
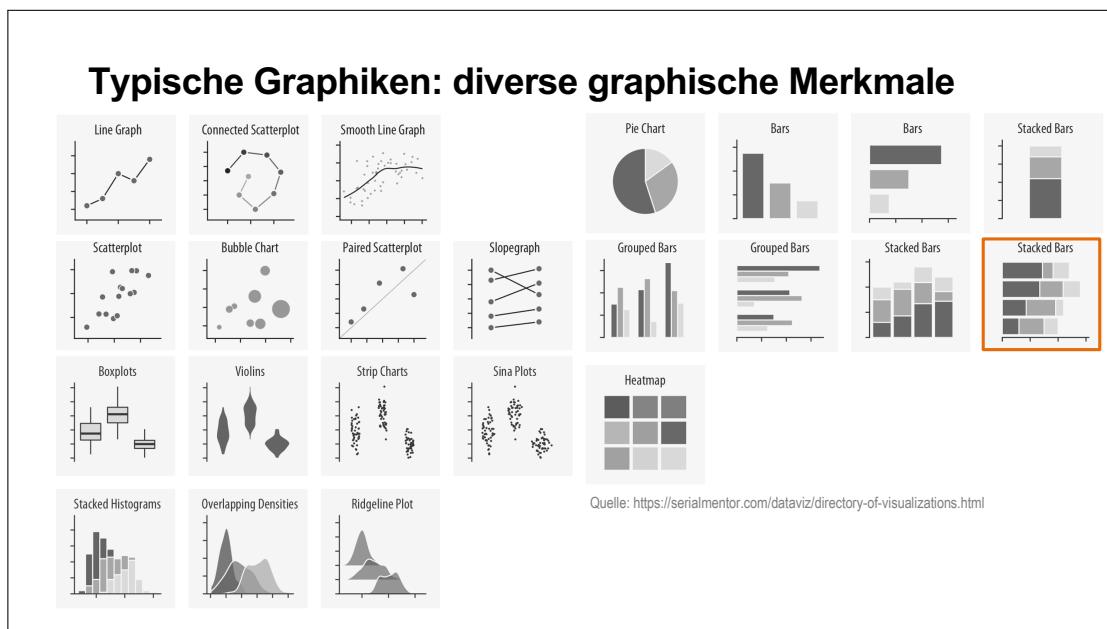
Jahr	Alterskategorie	Geschlecht	Kreis	Aufenthaltsdauer
1993	0-19	F	1	1.7
1993	0-19	F	2	5.1
1993	0-19	F	3	3.9
1993	0-19	F	4	3.4
1993	0-19	F	5	3.5

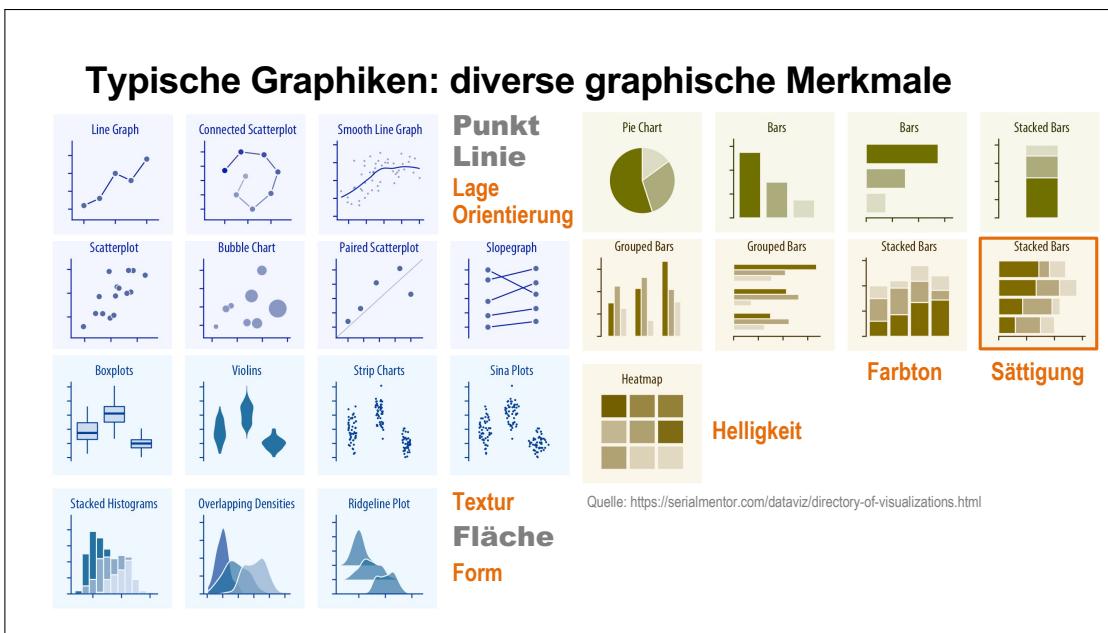
Quelle: <https://www.stadt-zuerich.ch/prd/de/index/statistik/themen/bevoelkerung/aufenthaltsdauer/aufenthaltsdauer-beim-wegzug.html>



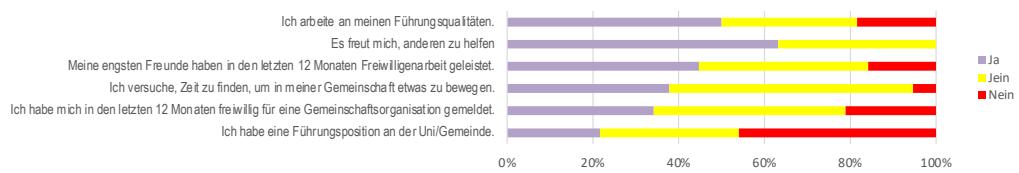
Was visualisiere ich?
1. Geometrie



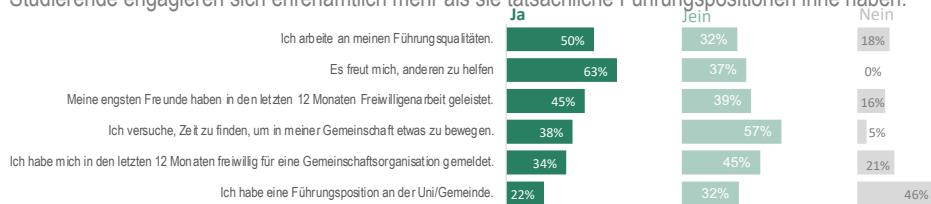




Gleicher Datensatz: unterschiedliche Darstellungen



Es besteht eine Lücke zwischen dem Erkennen von Führungschancen und dem Ergreifen dieser Chancen.
Studierende engagieren sich ehrenamtlich mehr als sie tatsächliche Führungspositionen inne haben.



Referenzen

Evergreen, Stephanie (2020). Effective Data Visualization. Second Edition, Sage, Thousand Oaks, (CA), U.S.A.

Wilke, Claus (2019). Fundamentals of Data Visualization. O'Reilly Media, Inc., Sebastopol, (CA), U.S.A.

Weblinks

<https://serialmentor.com/dataviz/>

<https://flowingdata.com/2018/10/17/ask-the-question-visualize-the-answer/>

Referentin

Prof. Sara Irina Fabrikant

Geographisches Institut

© Universität Zürich

Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 7: Effektive Informationsvisualisierung

Lektion 4: Für **wen** wird visualisiert?

Lernziele

- Sie kennen typische visuelle Wahrnehmungseffekte
- Sie wissen um das Prinzip der graphischen Prägnanz
- Sie können die Bausteine des Gesetzes der guten Gestalt nennen
- Sie können die Gestaltbausteine selbstständig in Visualisierungen anwenden

Die 5 Ws in Visualisierung

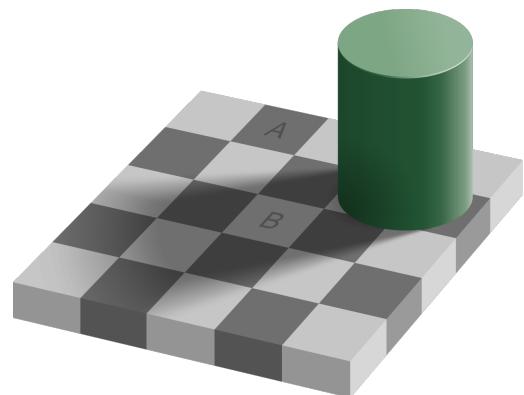
Warum | Was | für Wen | Wozu und Wie wird visualisiert?



Für **Wen** visualisiere ich?

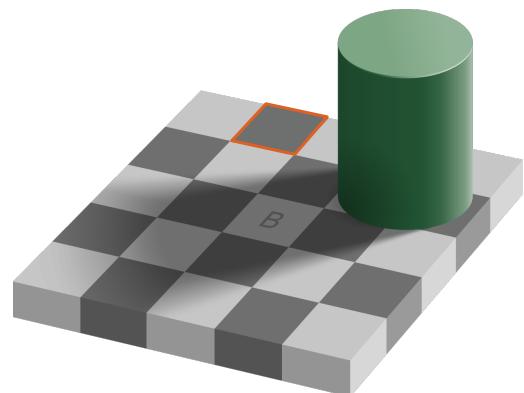
Für Personen mit einem Informationsbedürfnis unter Berücksichtigung deren visuellen Wahrnehmung.

Visuelle Realität...



Edward H. Adelson (1995)
<http://persci.mit.edu/gallery/checkershadow>

Visuelle Realität und Illusion!



Edward H. Adelson (1995)
<http://persci.mit.edu/gallery/checkershadow>

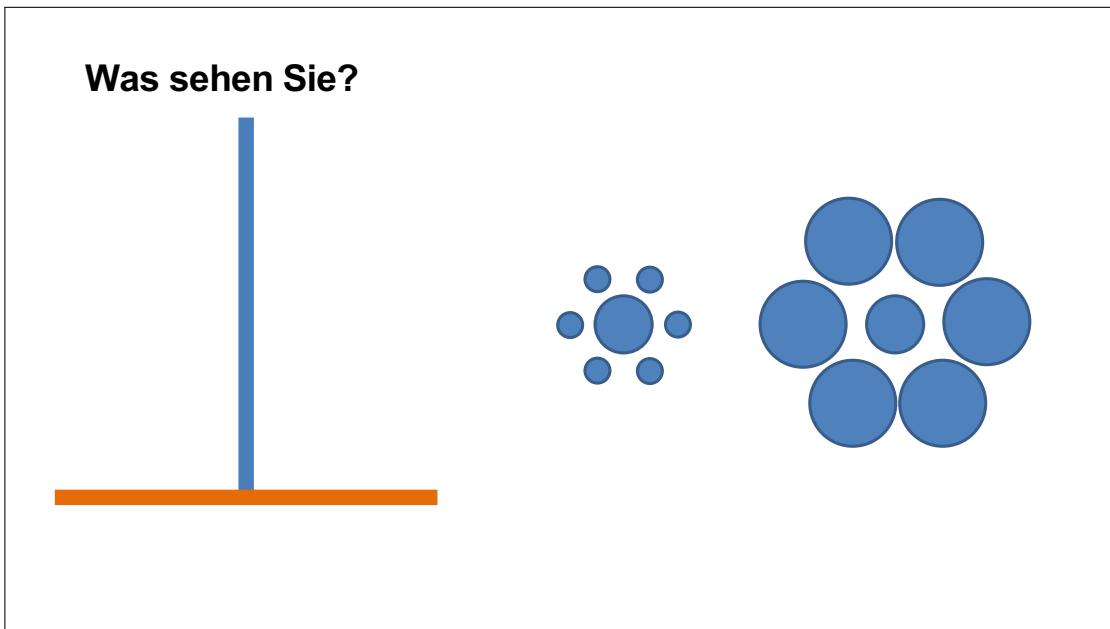


Menschliche Wahrnehmung
1. Perzeption

Was sehen Sie?

A diagram illustrating the Opel illusion. It consists of a short blue vertical line on a horizontal orange line. The blue line appears longer than the orange line due to the way the brain processes visual information.

[Oppel 1815–1894]
<https://www.illusionsindex.org/ir/vertical-horizontal-illusion>



Was sehen Sie?

The slide features four color palettes arranged in a 2x2 grid. The top row consists of a grayscale palette (left) and a dark palette (right). The bottom row consists of a green palette (left) and a yellow palette (right). Each palette is a 4x6 grid of color swatches, with each swatch labeled with a name such as 'Burg', 'Orange', 'Teal', etc. Below the palettes, there is a link: Farblegenden: <https://carto.com/carto-colors/>.



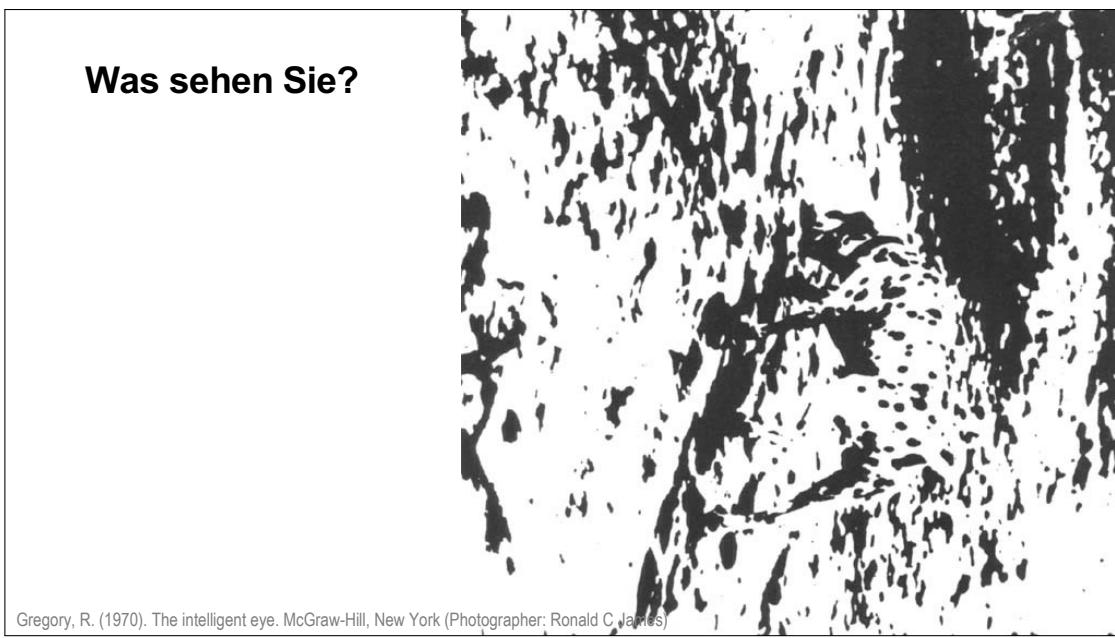
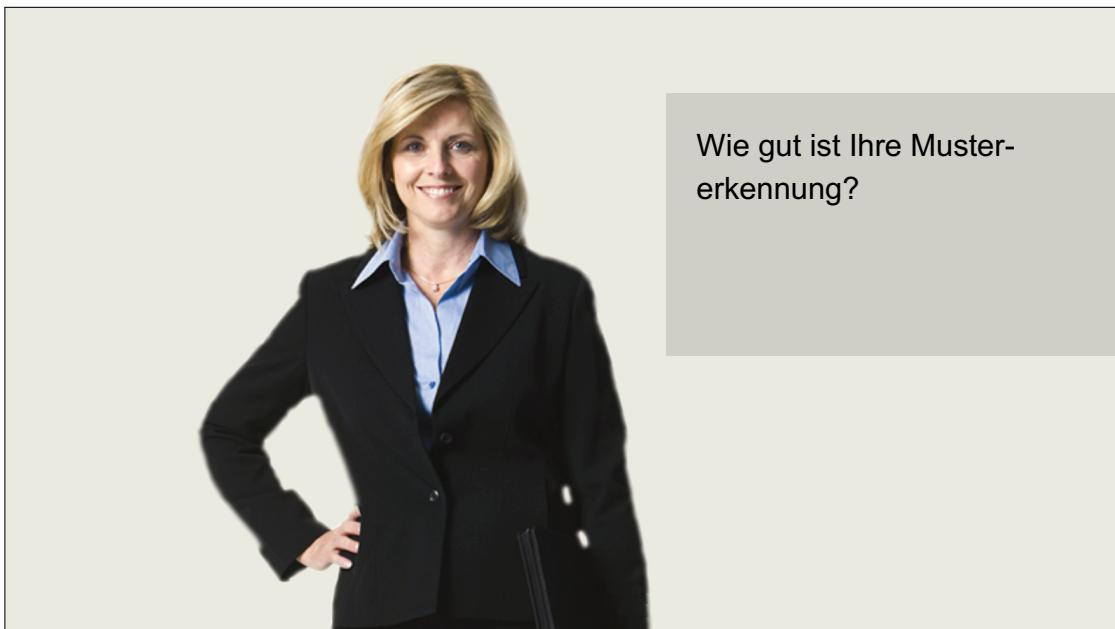
Menschliche Wahrnehmung

1. Perzeption
2. Kognition

Lesen Sie den Text laut vor...

**ROT
GRÜN
BLAU
ROT
GELB
GRÜN
BLAU
ROT
BLAU
GELB**

Stroop, R. (1935) Studies of interference in serial verbal reactions. In: Journal of Experimental Psychology. Band 18, 1935, S. 643–662.



Was sehen Sie jetzt?



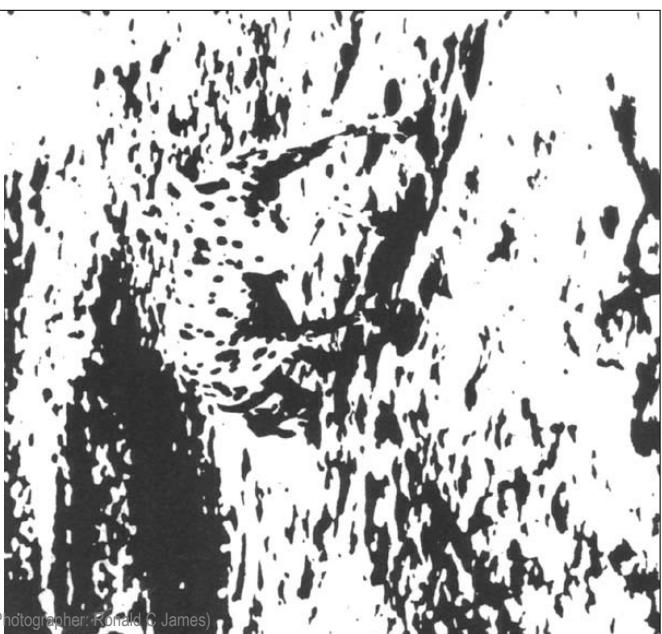
Gregory, R. (1970). The intelligent eye. McGraw-Hill, New York (Photographer: Ronald C James)

Was sehen Sie nun?



Gregory, R. (1970). The intelligent eye. McGraw-Hill, New York
(Photographer: Ronald C James)

Was sehen Sie?



Gregory, R. (1970). The intelligent eye. McGraw-Hill, New York (Photographer: Ronald O James)

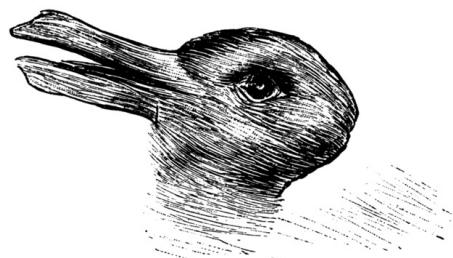
Was sehen Sie?



[anon. ca. 1888]

Was sehen Sie?

Welche Thiere gleichen ein-
ander am meisten?

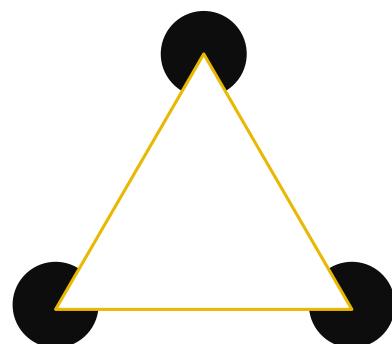


[Fliegende Blätter 23 October 1892]

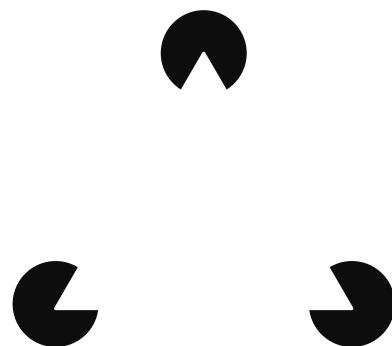
Was sehen Sie?



Ein weisses Dreieck?



Ein weisses Dreieck?



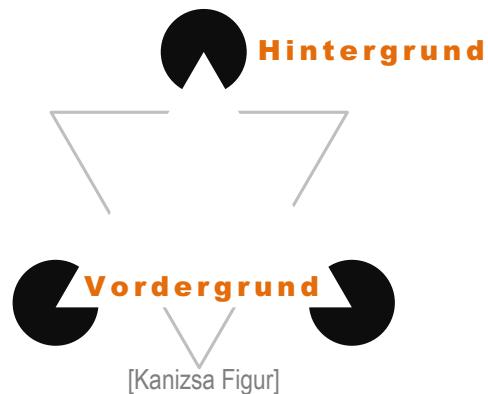


Was ist visuell prägnant?

The diagram illustrates the Kanizsa figure, a classic example of visual perception. It consists of several black and white shapes: two black circles at the top, two white triangles pointing downwards from behind them, and two black circles at the bottom. The text "Hintergrund" (Background) is written next to the top circles, and "Vordergrund" (Foreground) is written next to the bottom circles. A bracket below the bottom circles is labeled "[Kanizsa Figur]".

Kanizsa, G., 1955. Margini quasi-percettivi in campi con stimolazione omogenea, Rivista di Psicologia, 49 (1): 7–30.

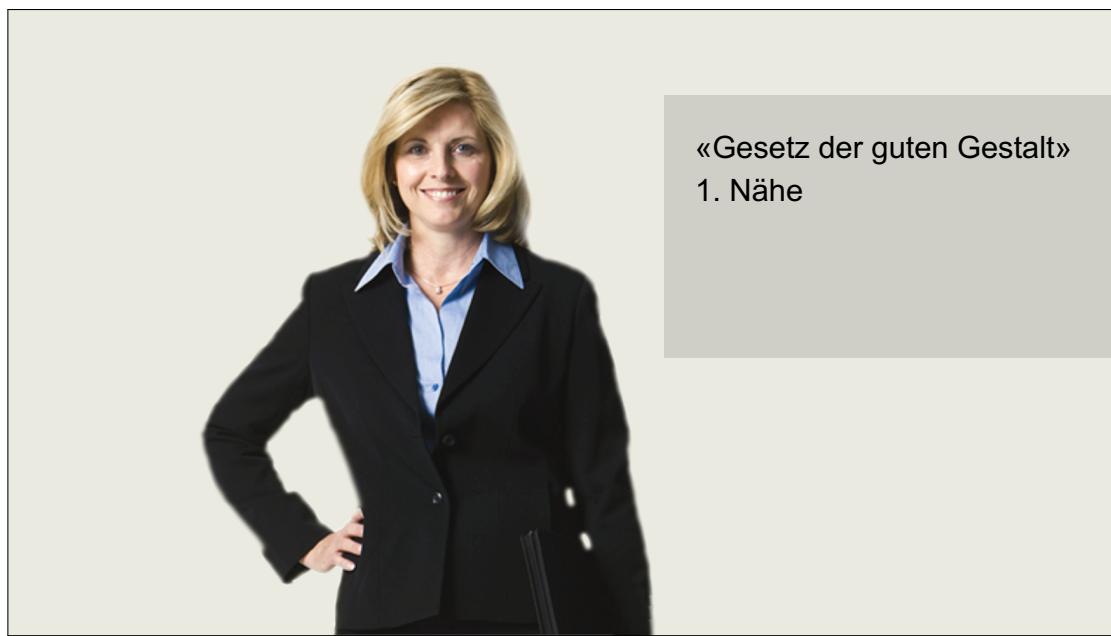
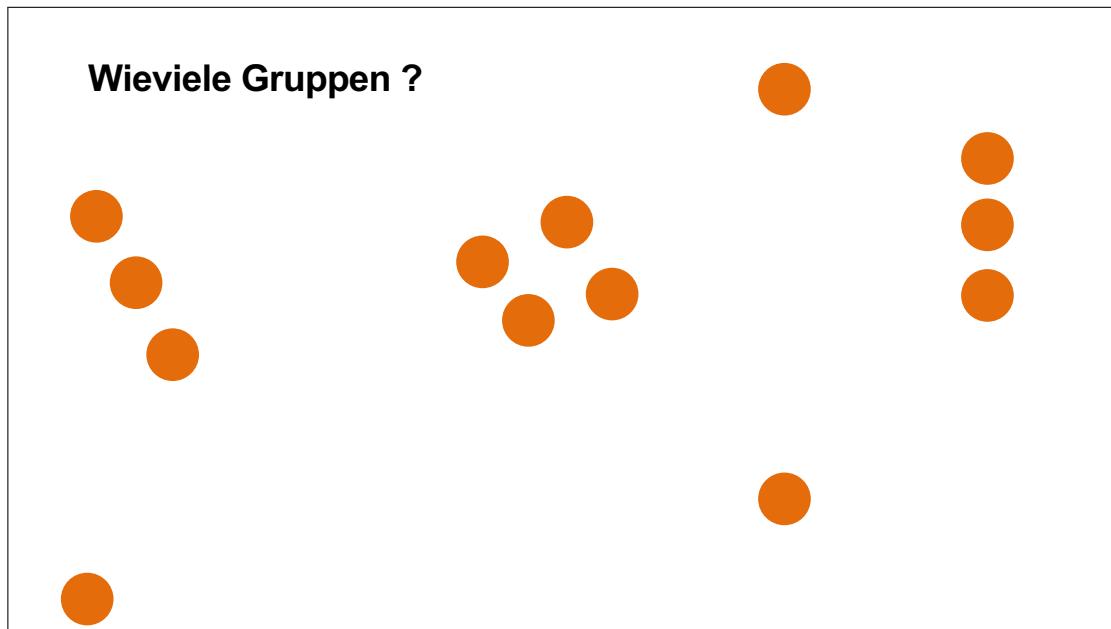
Was ist visuell prägnant?



Kanizsa, G., 1955. Margini quasi-percettivi in campi con stimolazione omogenea, Rivista di Psicologia, 49 (1): 7–30.

Wie erreiche ich visuelle Prägnanz?



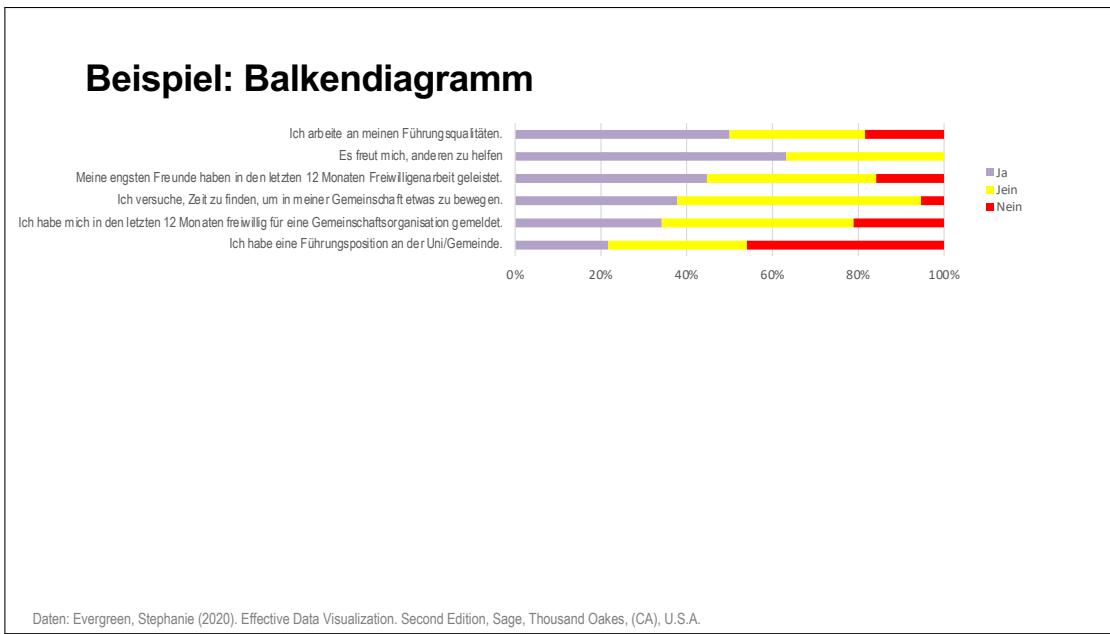
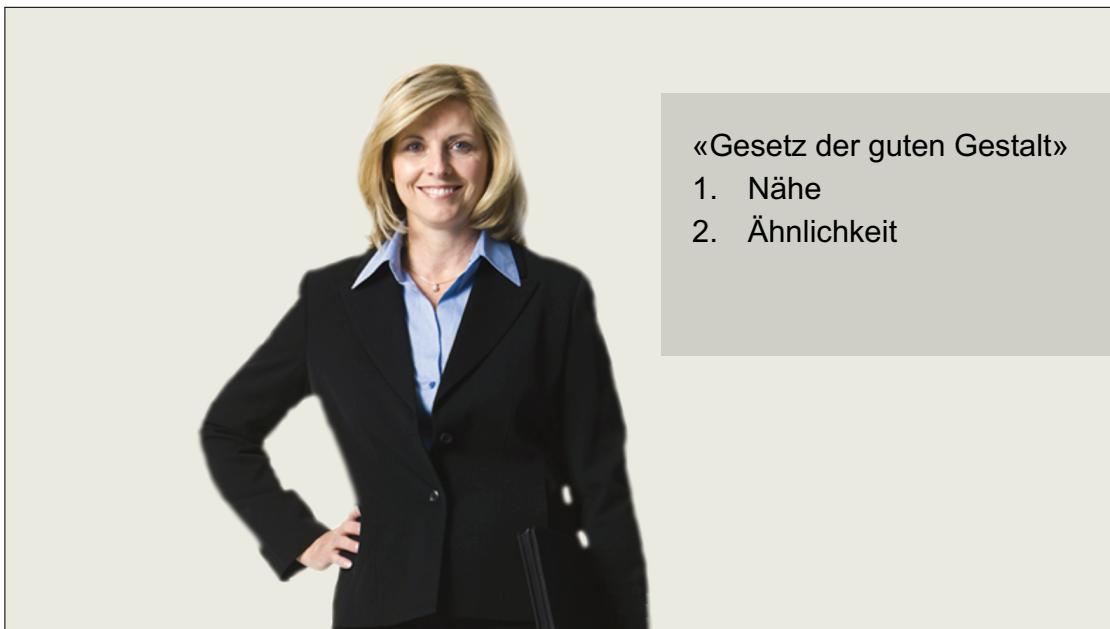


Wieviele Gruppen ?

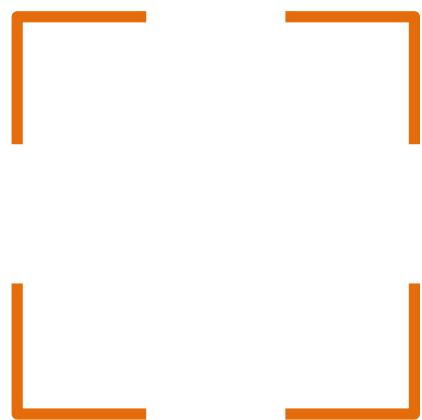
O X O X O X O X
O X O X O X O X
O X O X O X O X
O X O X O X O X

Wieviele Gruppen ?

O X O X O X O X
O X O X O X O X
O X O X O X O X
O X O X O X O X



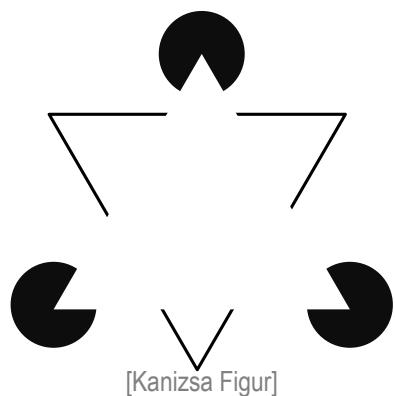
Wieviele Figuren?



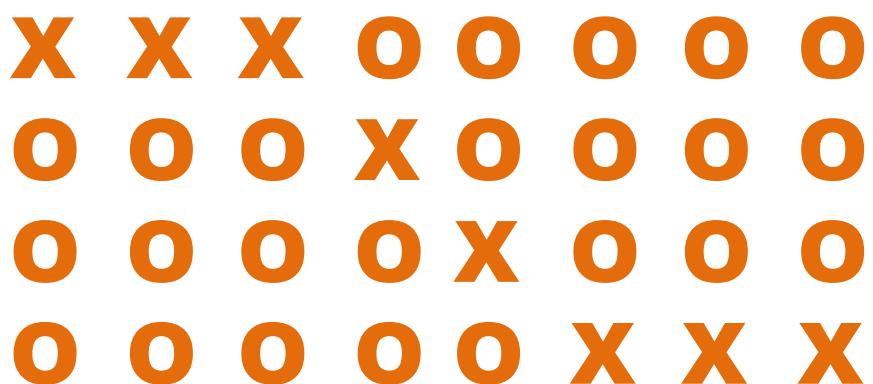
«Gesetz der guten Gestalt»

1. Nähe
2. Ähnlichkeit
3. Geschlossenheit

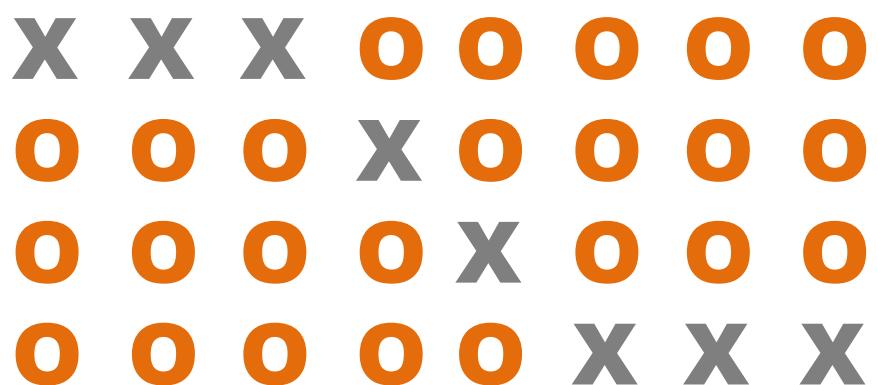


Wieviele Figuren

Kanizsa, G., 1955. Margini quasi-percettivi in campi con stimolazione omogenea, Rivista di Psicologia, 49 (1): 7–30.

Wieviele Gruppen ?

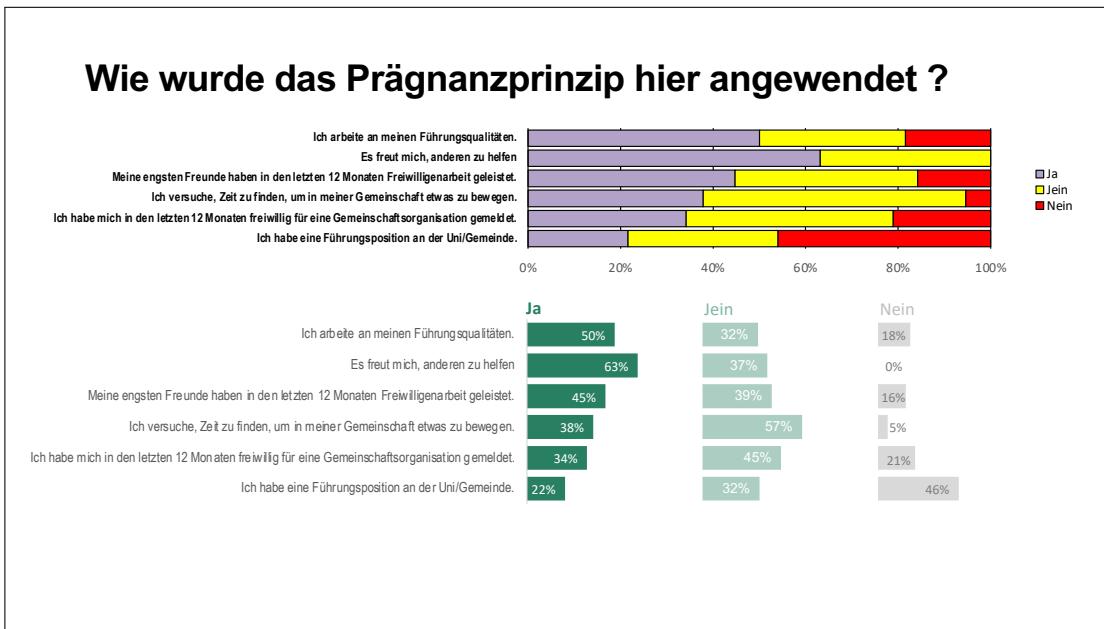
Wieviele Gruppen ?



«Gesetz der guten Gestalt»

1. Nähe
2. Ähnlichkeit
3. Kontinuität





Referenzen

- Evergreen, Stephanie (2020). Effective Data Visualization. Second Edition, Sage, Thousand Oaks, (CA), U.S.A.
- Gregory, R.L. (1997). Eye and Brain: The Psychology of Seeing (5th ed.). Oxford University Press, Oxford, U.K.
- Ware, C. (2008). Visual Thinking for Design. Morgan Kaufmann, San Francisco, U.S.A.
- Wilke, Claus (2019). Fundamentals of Data Visualization. O'Reilly Media, Inc., Sebastopol, (CA), U.S.A.

Weblinks

<https://serialmentor.com/dataviz/>

<http://persci.mit.edu/gallery/checkershadow>

<https://www.e-study-psychologie.de/mod/book/view.php?id=2929&chapterid=3300>

Referentin

Prof. Sara Irina Fabrikant

Geographisches Institut

© Universität Zürich

Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 7: Effektive Informationsvisualisierung

Lektion 5: **Wozu** visualisiere ich?

Lernziele

- Sie können gezielte Fragen stellen um Mengen, Verteilungen, Anteile, oder Zusammenhänge zu visualisieren.
- Sie sind befähigt den geeigneten Diagrammtyp für Ihre Fragestellung auszuwählen.
- Sie können diverse Visualisierungslösungen vorschlagen, um Mengen, Verteilungen, Anteile oder Zusammenhänge effektiv darzustellen.

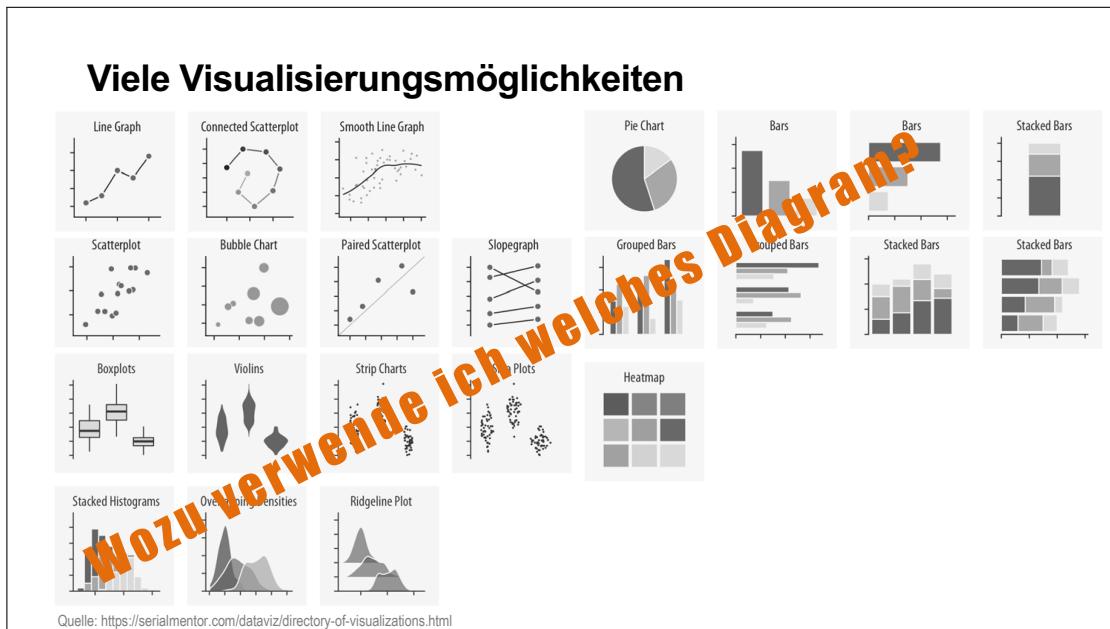
Die 5 Ws in Visualisierung

Warum | Was | für Wen | **Wozu** und Wie wird visualisiert?



Wozu visualisiere ich?

Um die richtige Antwort
auf mein
Informationsbedürfnis zu
erhalten.





Wozu visualisiere ich?

1. Mengen
2. Verteilungen
3. Anteile
4. Zusammenhänge

Visualisierung einer Menge

Wie **hoch** ist der mittlere globale Wasserverbrauch, um 100 Gramm Schokolade herzustellen?



Visualisierung mehrerer Mengen

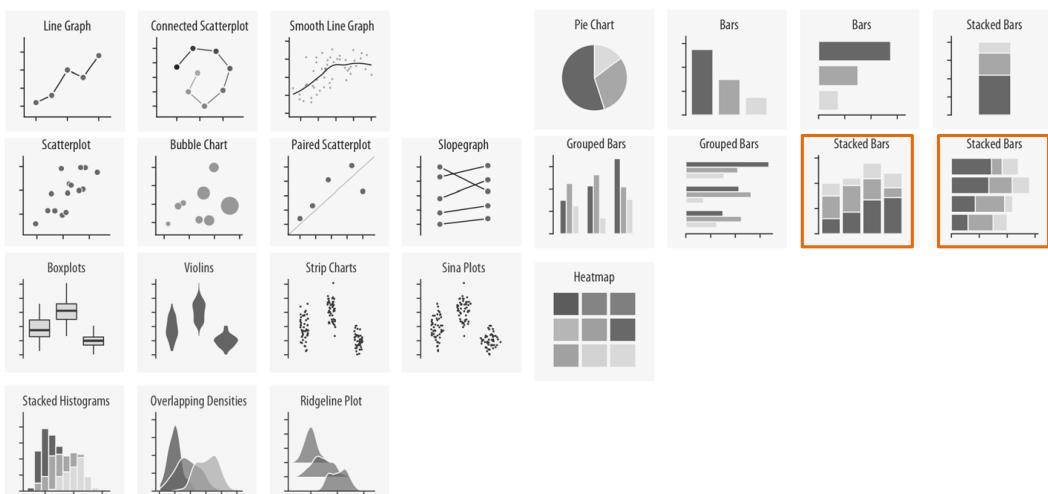
Wie **gross** ist der durchschnittliche virtuelle Wasserfussabdruck von Ländern **im globalen Vergleich**?

Wassermengenvergleich

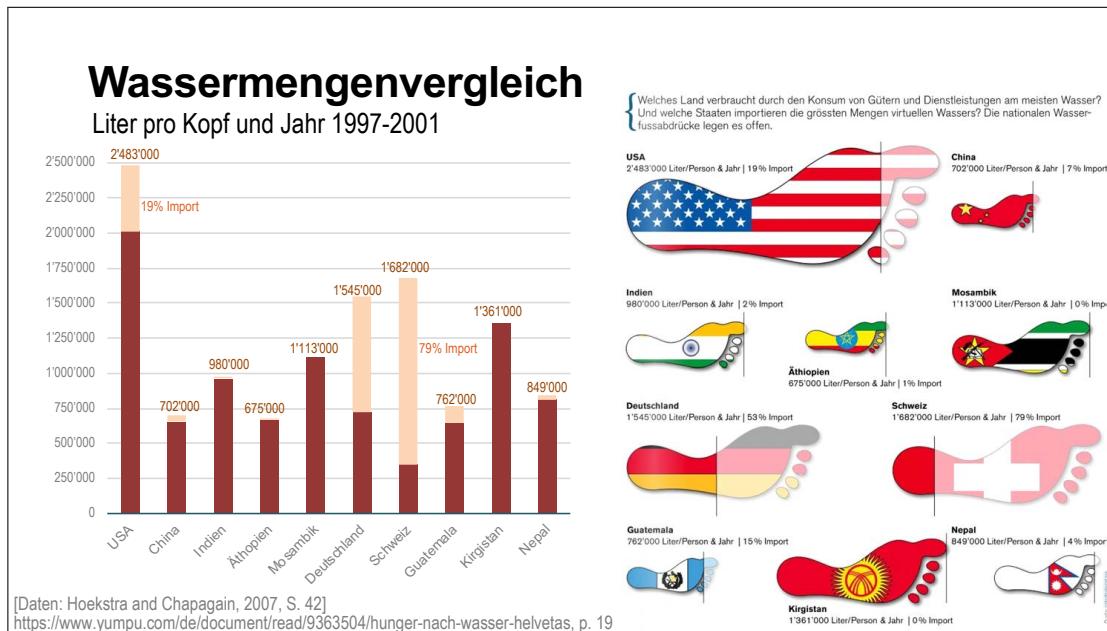
Land	Verbrauch (Liter/Person und Jahr)	Import (%)
USA	2'483'000	19
China	702'000	7
Indien	980'000	2
Äthiopien	675'000	1
Mosambik	1'113'000	0
Deutschland	1'545'000	53
Schweiz	1'682'000	79
Guatemala	762'000	15
Kirgistan	1'361'000	0
Nepal	849'000	4

<https://www.yumpu.com/de/document/read/9363504/hunger-nach-wasser-helvetas>

Viele Visualisierungsmöglichkeiten



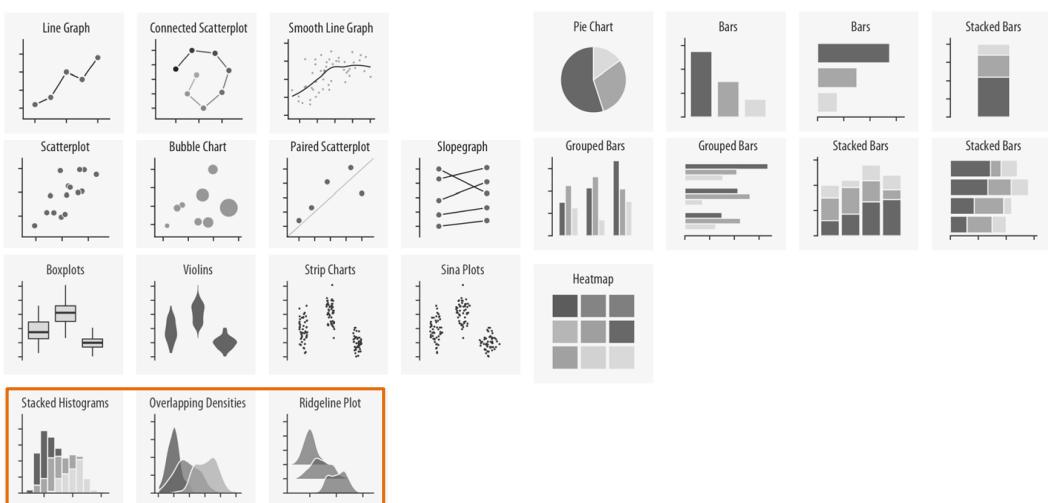
Quelle: <https://serialmentor.com/dataviz/directory-of-visualizations.html>

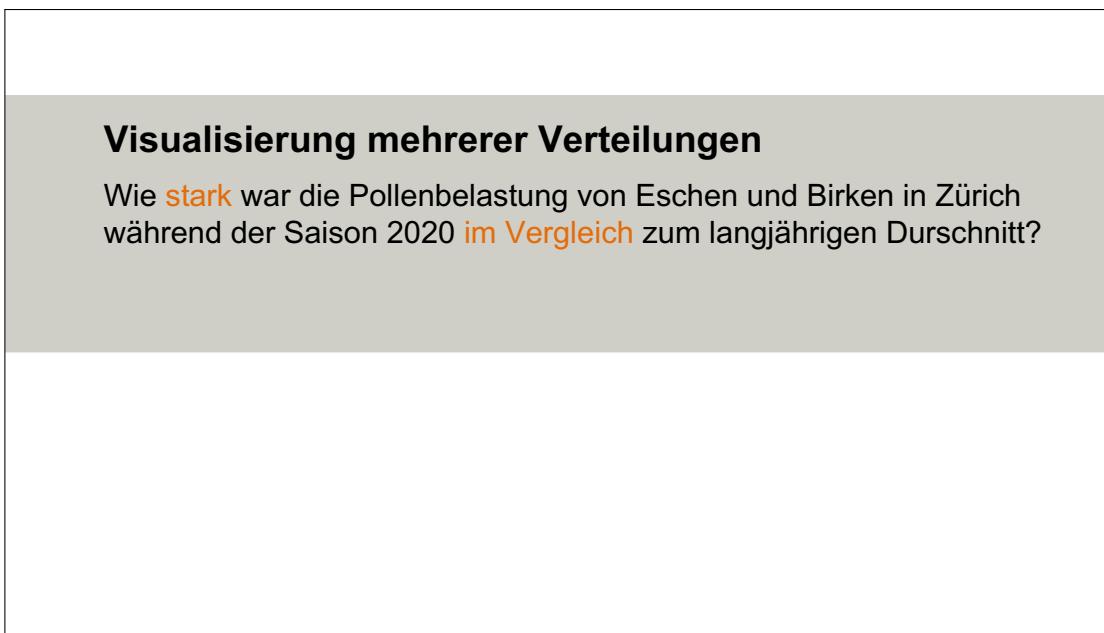
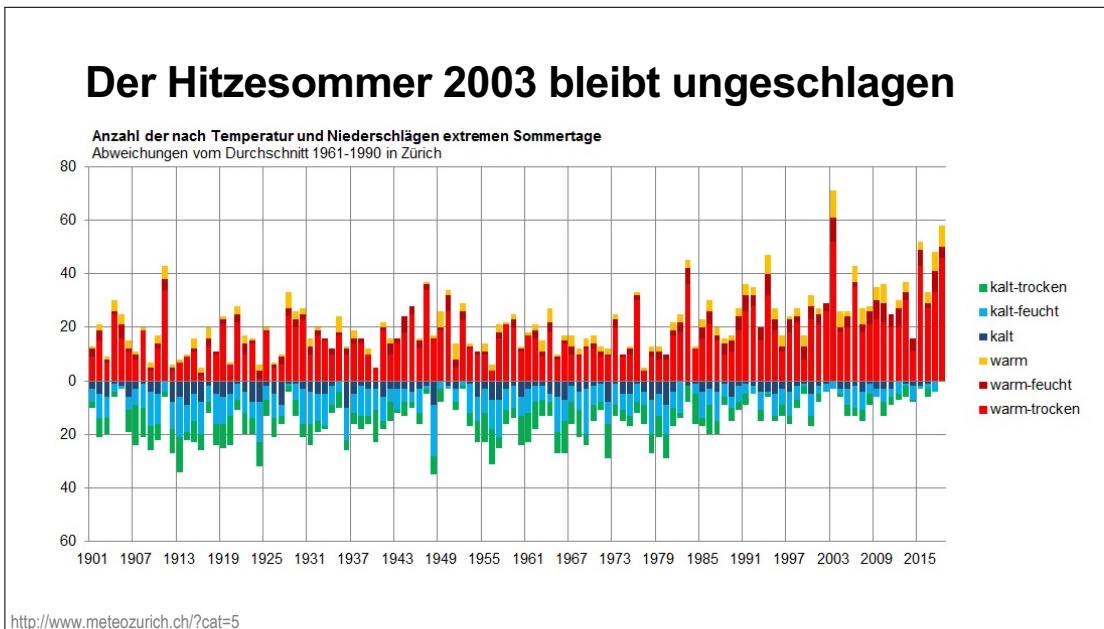


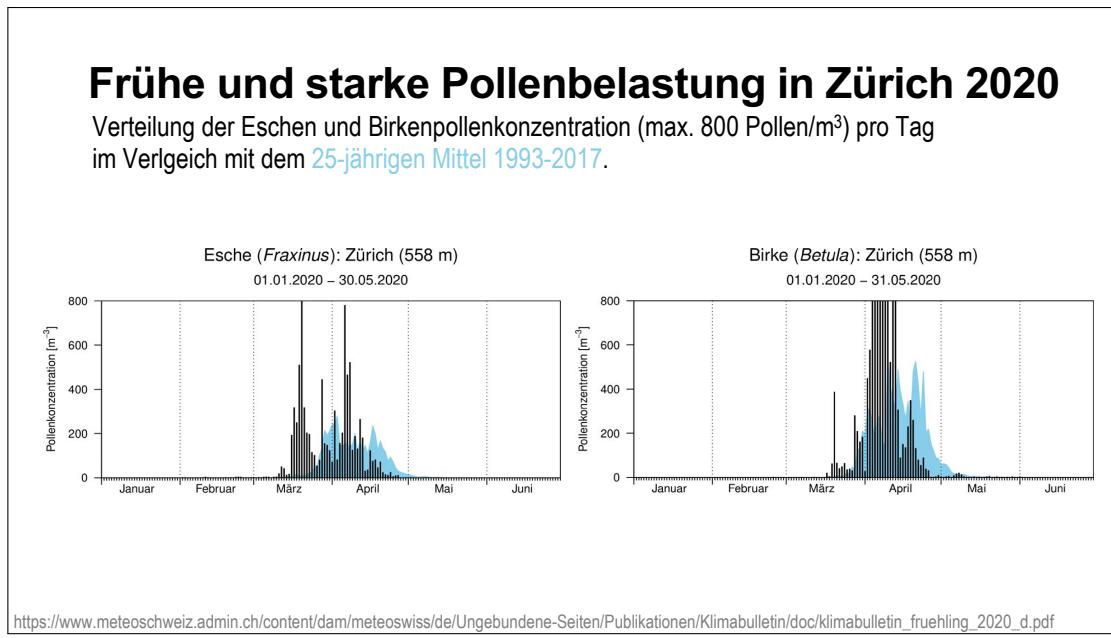
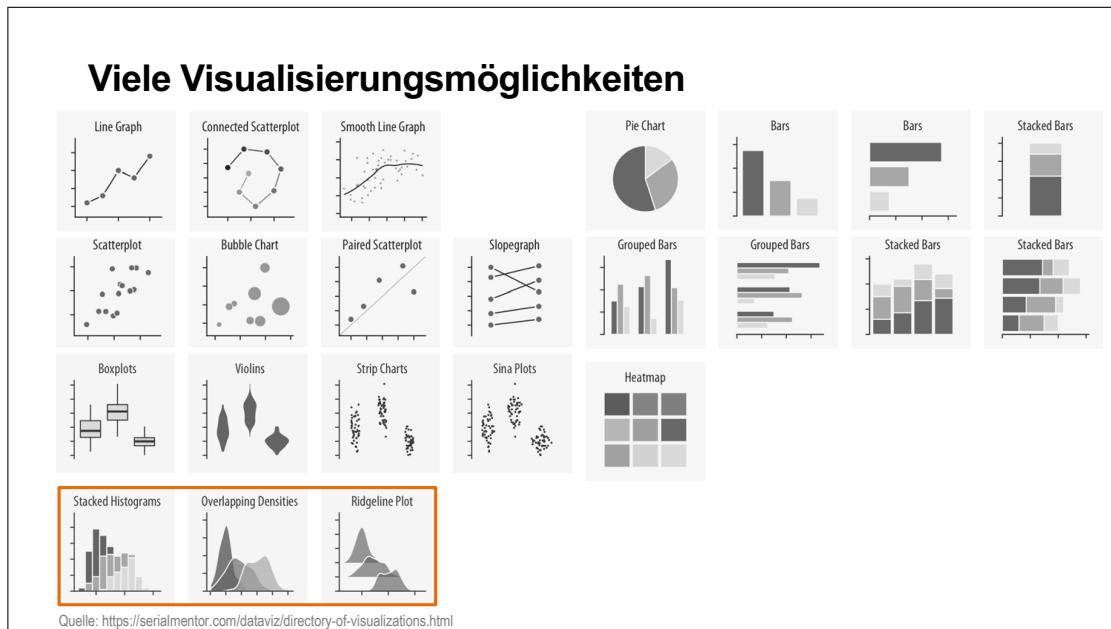
Visualisierung einer Verteilung

Wird der Sommer immer **heisser** in der Stadt Zürich?

Viele Visualisierungsmöglichkeiten







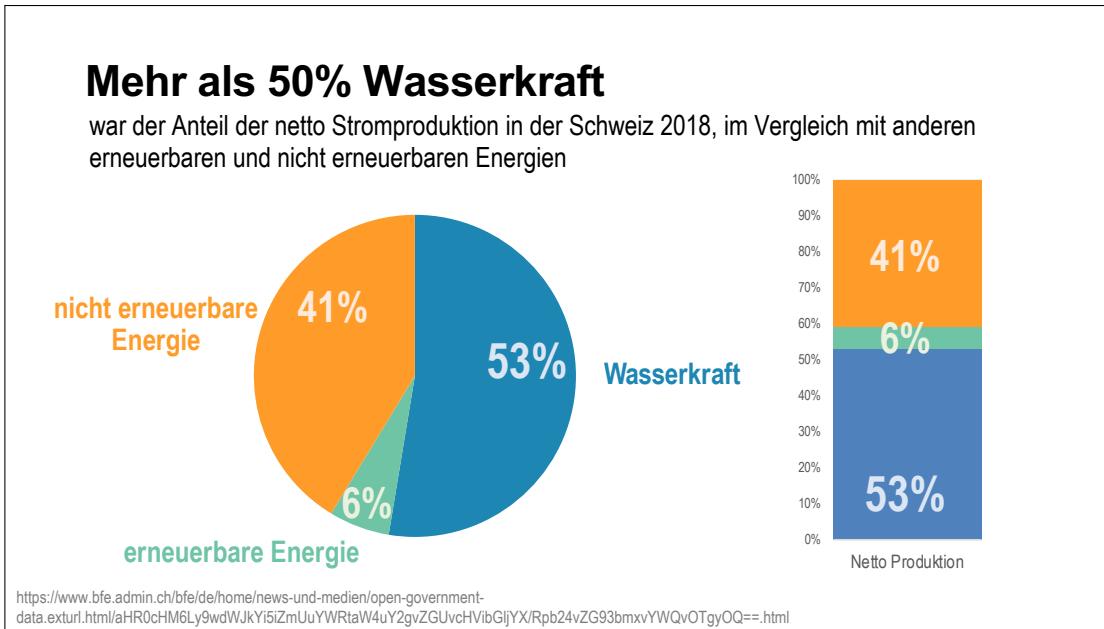
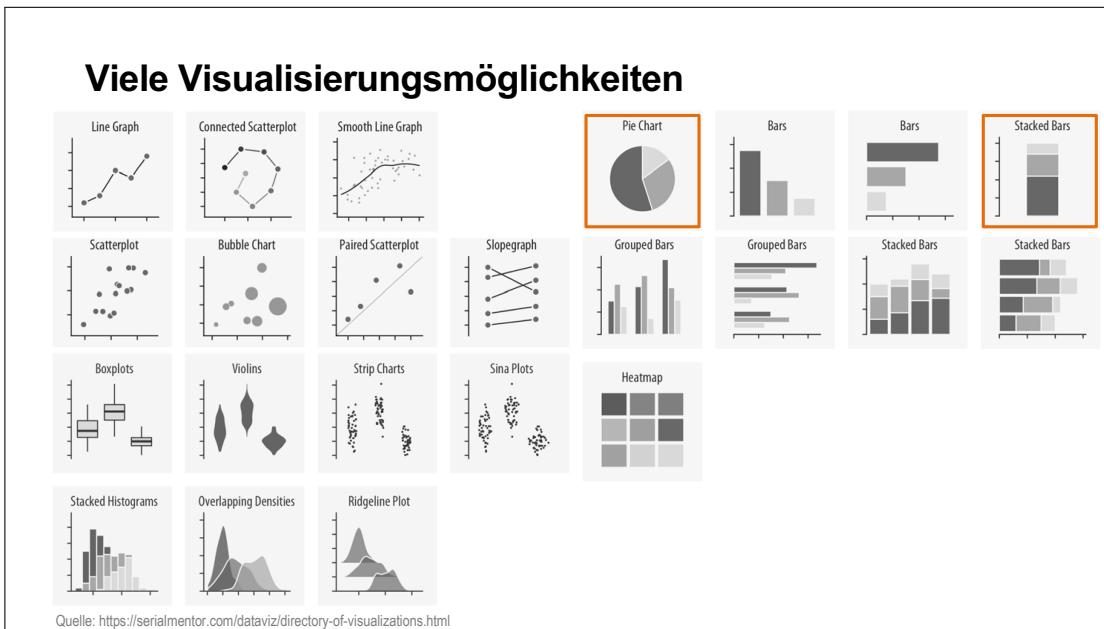


Wozu visualisiere ich?

- 1. Mengen
- 2. Verteilungen
- 3. Anteile**
- 4. Zusammenhänge

Visualisierung eines Anteils

Wie **gross** ist der **Anteil** der Wasserkraft an der **gesamten** Stromproduktion in der Schweiz?

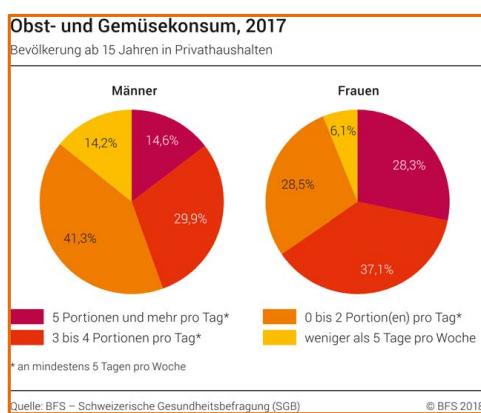


Visualisierung mehrerer Anteile

Wie gross ist der **Anteil** der Personen in der Schweiz, die mindestens fünf Portionen Obst oder Gemüse pro Woche konsumieren? Gibt es Geschlechterunterschiede?

Doppelt soviele Frauen wie Männer

konsumierten in der Schweiz mindestens 5 Portionen Obst- und Gemüse pro Woche in 2017.



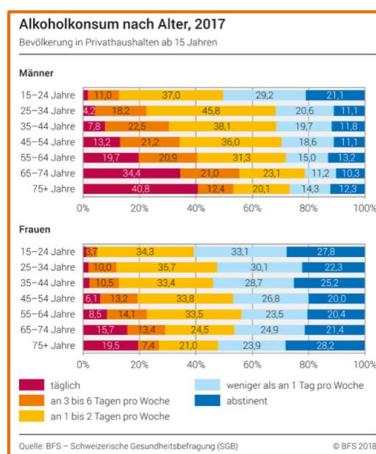
<https://www.bfs.admin.ch/bfs/de/home/statistiken/gesundheit/determinanten/ernaehrung.html>

Visualisierung mehrerer Anteile

Welche Altersklasse hat den **höchsten Anteil** am täglichen Alkoholkonsum in der Schweiz. Gibt es Geschlechterunterschiede?

Doppelt soviele Männer ab 75 J. wie Frauen

trinken täglich Alkohol in der Schweiz 2017



<https://www.bfs.admin.ch/bfs/de/home/statistiken/gesundheit/determinanten/alkohol.html>

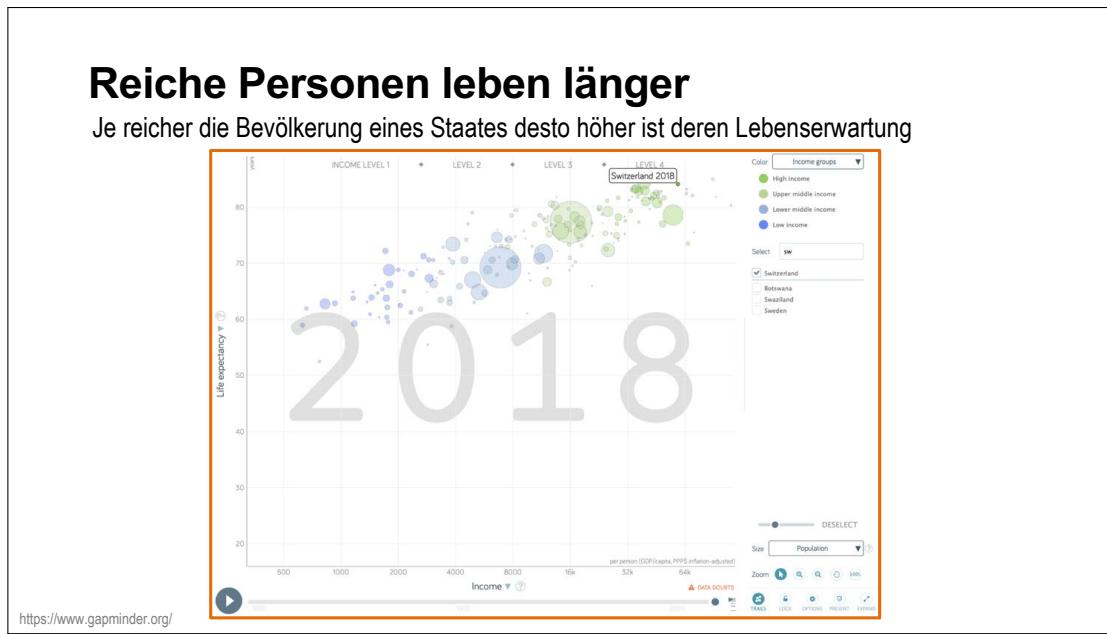
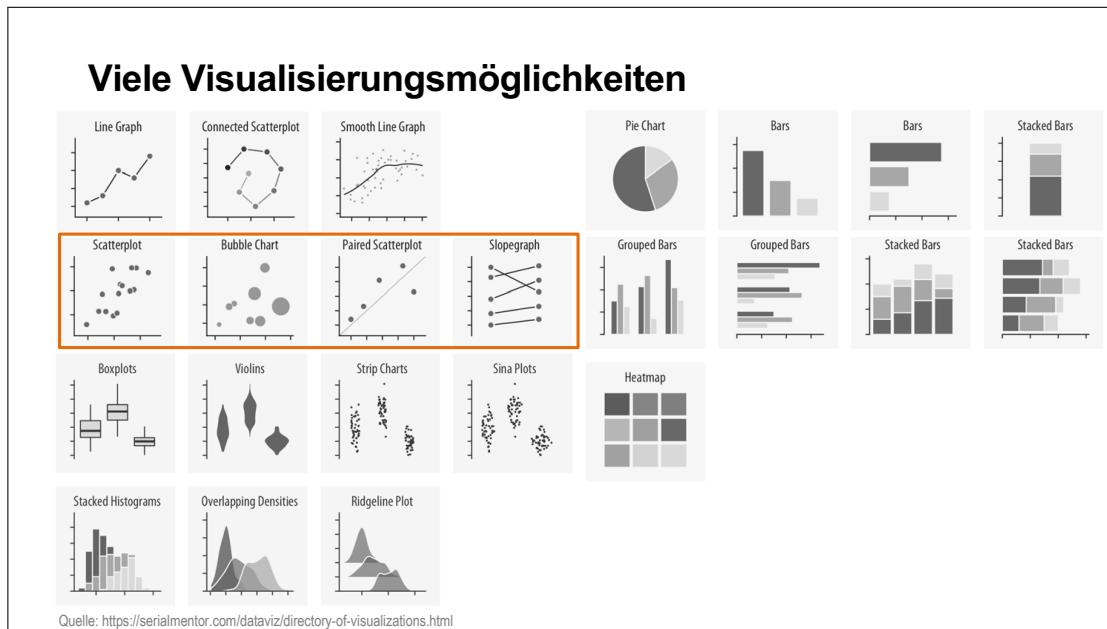


Wozu visualisiere ich?

1. Mengen
2. Verteilungen
3. Anteile
4. **Zusammenhänge**

Visualisierung eines Zusammenhangs

Besteht ein **Zusammenhang** zwischen dem Einkommen und der Lebenserwartung in der Weltbevölkerung?

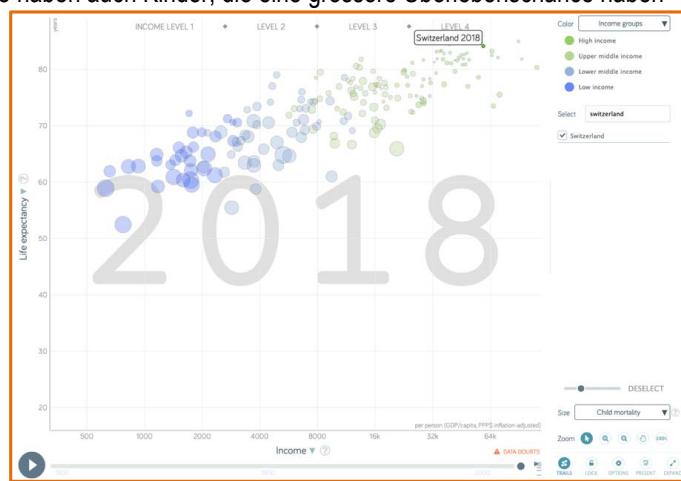


Visualisierung von mehreren Zusammenhängen

Besteht ein **Zusammenhang** zwischen dem Einkommen, der Lebenserwartung und der Kindersterblichkeit der Weltbevölkerung?

Reiche Personen leben nicht nur länger

sondern sie haben auch Kinder, die eine grössere Überlebenschance haben



<https://www.gapminder.org/>

Referenzen

Evergreen, Stephanie (2020). Effective Data Visualization. Second Edition, Sage, Thousand Oaks, (CA), U.S.A.

Hoekstra, A.Y., Chapagain, A.K. Water footprints of nations: Water use by people as a function of their consumption pattern. *Water Resour Manage* 21, 35–48 (2007).
<https://doi.org/10.1007/s11269-006-9039-x>, Table 3, p. 42

Wilke, Claus (2019). Fundamentals of Data Visualization. O'Reilly Media, Inc., Sebastopol, (CA), U.S.A.

Weblinks

<https://serialmentor.com/dataviz/>

https://www.pseau.org/outils/ouvrages/waterfootprint_comprehensive_introduction_to_water_footprints_en.pdf

http://www.education21.ch/sites/default/files/uploads/pdf-d/Mystery_Virtuelles-Wasser_Sek-II_de.pdf

Rosling TED Talk 2006:

https://www.ted.com/talks/hans_rosling_the_best_stats_you_ve_ever_seen#t-267880

Referentin

Prof. Sara Irina Fabrikant
Geographisches Institut

© Universität Zürich
Digital Society Initiative



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 7: Effektive Informationsvisualisierung

Lektion 5: Wie visualisiere ich?

Lernziele

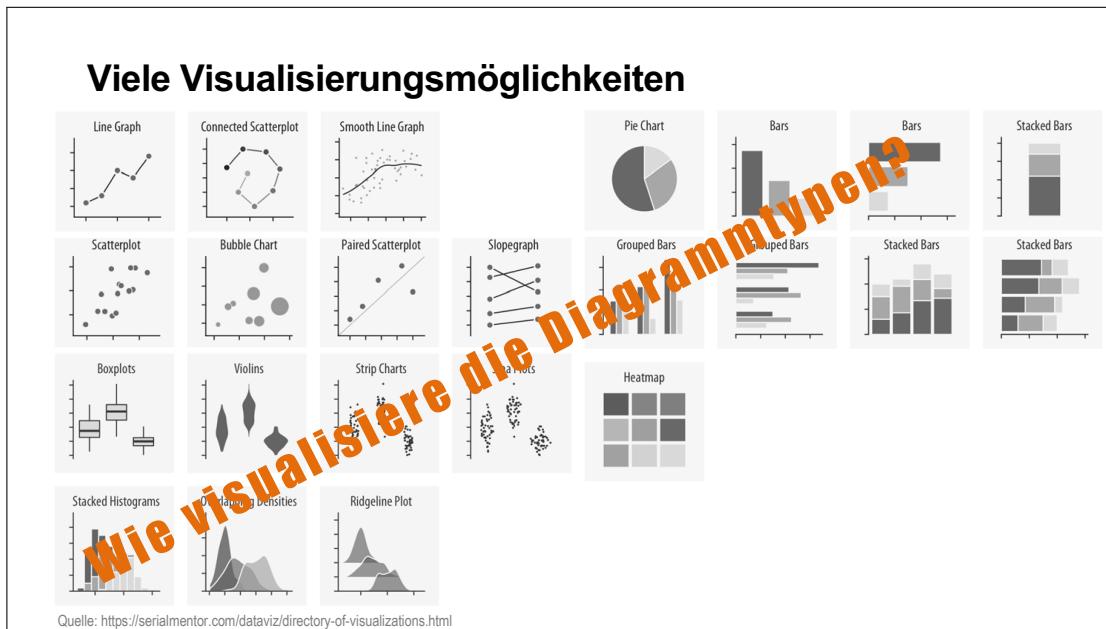
- Sie wissen welche Diagrammkomponenten bearbeitet werden sollten, um die Lesbarkeit der Visualisierung zu erhöhen.
- Sie kennen die graphischen Variablen, um Diagrammtypen attraktiv und gut lesbar zu gestalten
- Sie können diverse graphische Lösungen vorschlagen, um Mengen, Verteilungen, Anteile oder Zusammenhänge prägnant darzustellen.

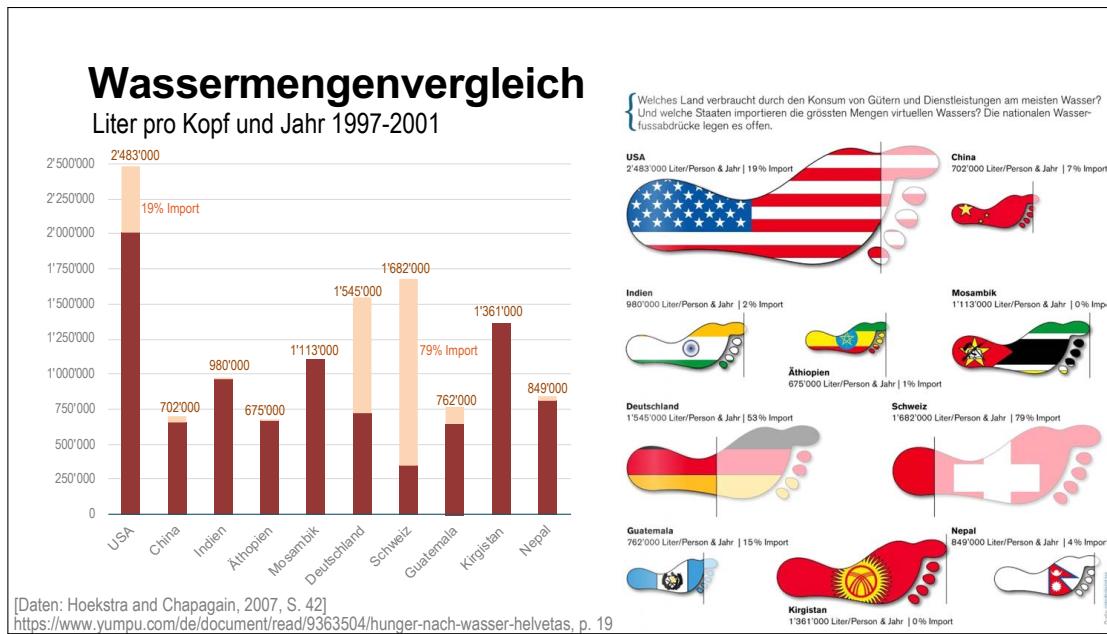
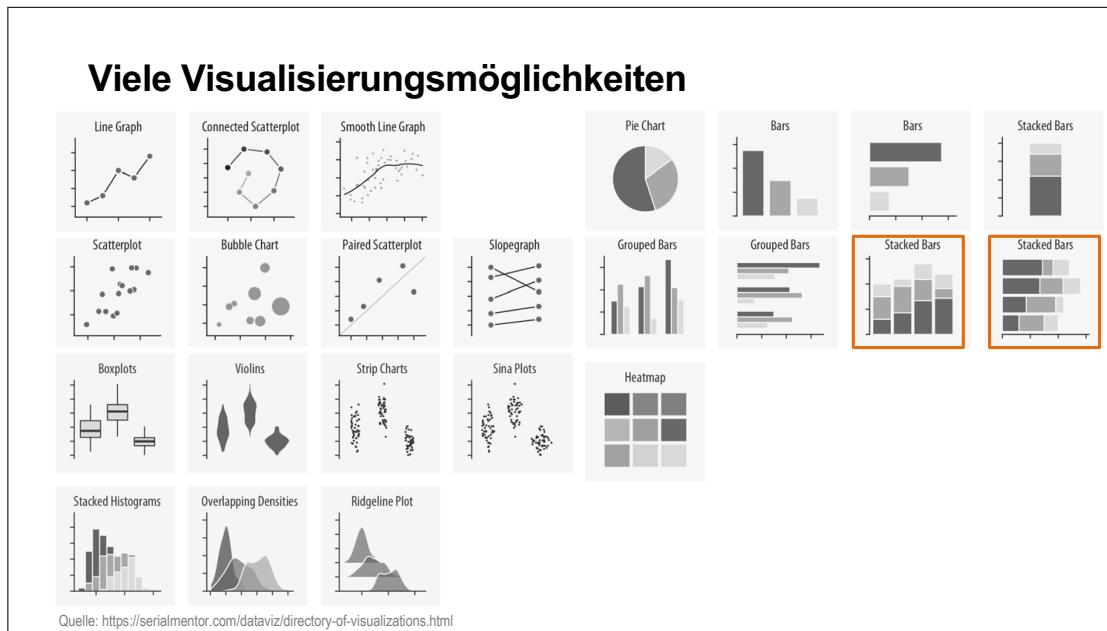
Die 5 Ws in Visualisierung

Warum | Was | für Wen | Wozu und Wie wird visualisiert?

Wie visualisiere ich?

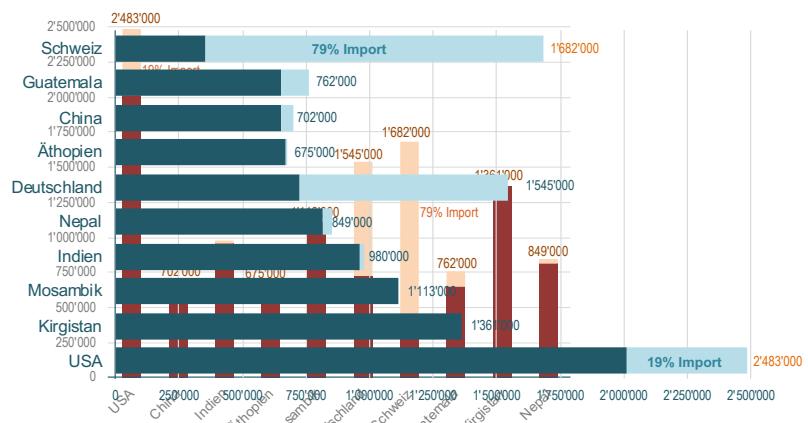






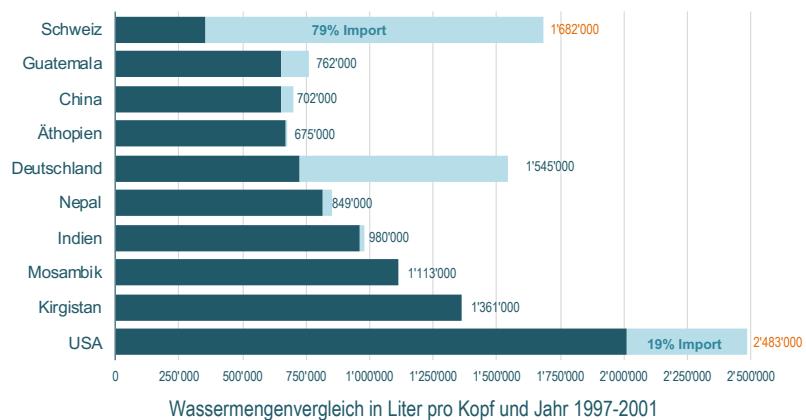
Wassermengenvergleich

Liter pro Kopf und Jahr 1997-2001



[Daten: Hoekstra and Chapagain, 2007, S. 42]
<https://www.yumpu.com/de/document/read/9363504/hunger-nach-wasser-helvetas, p. 19>

Die Schweiz importiert 79% ihres Wasserfußabdrucks



[Daten: Hoekstra and Chapagain, 2007, S. 42]
<https://www.yumpu.com/de/document/read/9363504/hunger-nach-wasser-helvetas, p. 19>

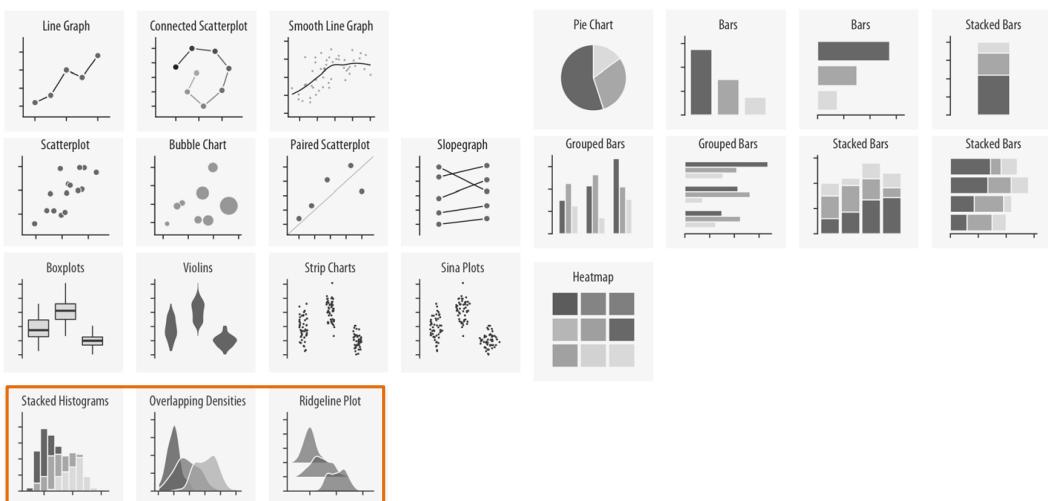
Nun kommen wir zu graphischen Verbesserungsvorschlägen, um Verteilungen prägnant zu visualisieren



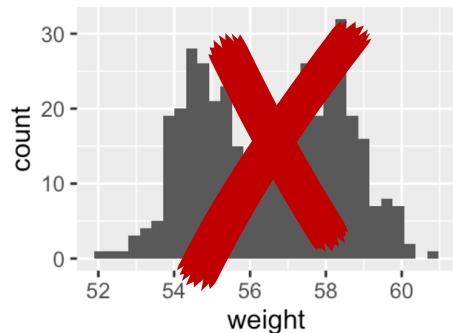
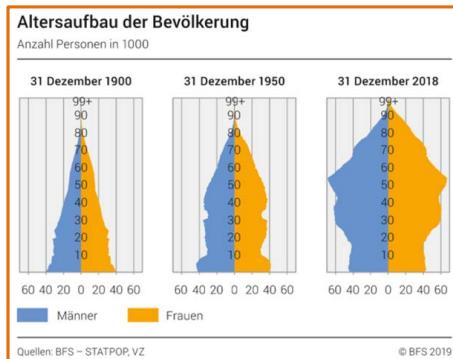
Wozu visualisiere ich?

1. Mengen
2. **Verteilungen**
3. Anteile
4. Zusammenhänge

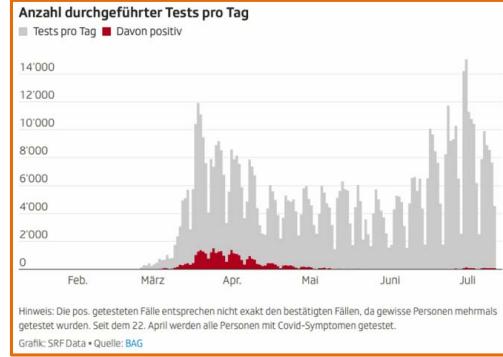
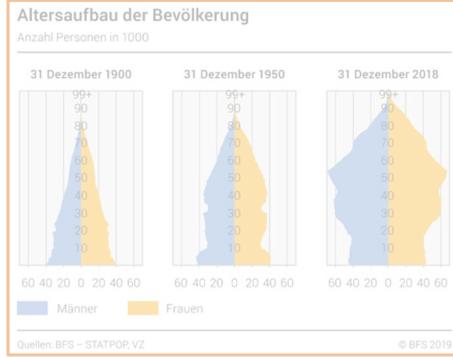
Viele Visualisierungsmöglichkeiten

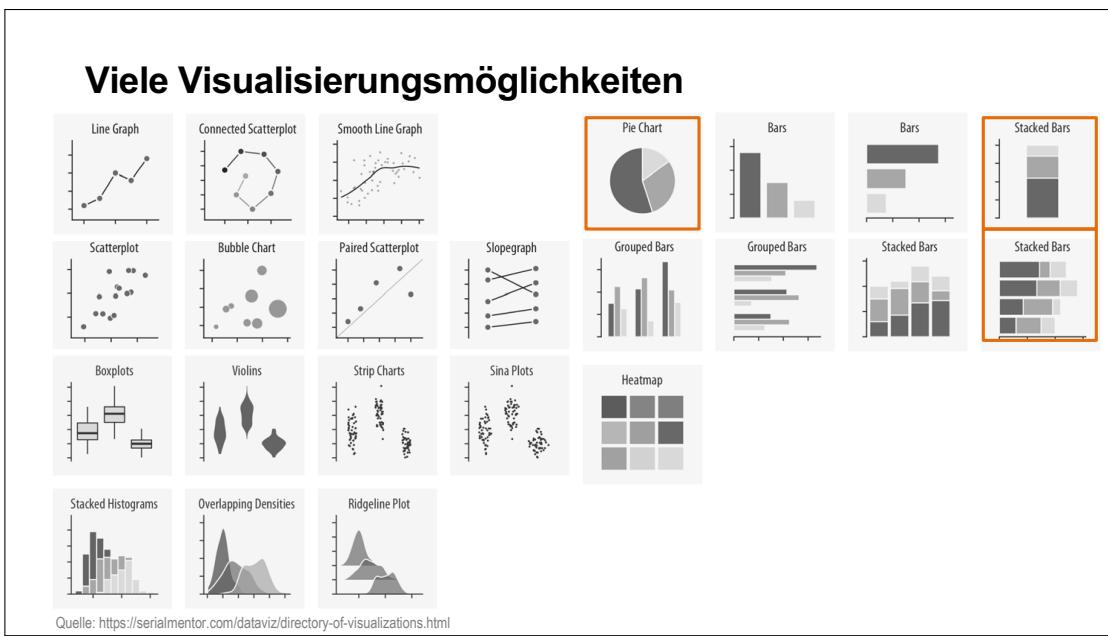


Die Schweizer Bevölkerung wird immer älter



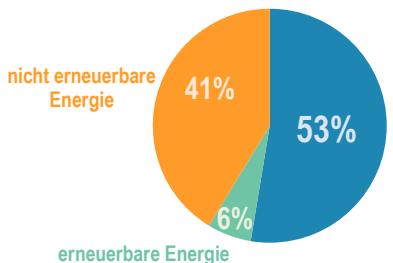
Die positiv getesteten Corona Fälle sind stark gesunken, obwohl insgesamt mehr getestet wird





Mehr als 50% Wasserkraft

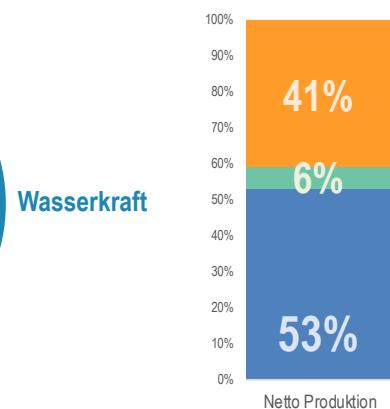
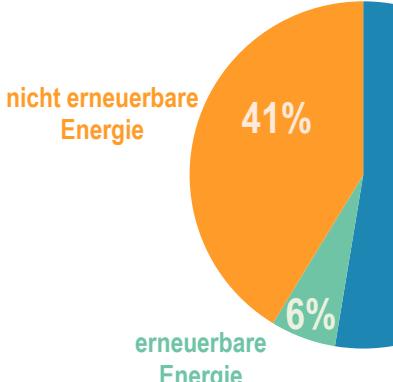
war der Anteil der netto Stromproduktion in der Schweiz 2018, im Vergleich mit anderen erneuerbaren und nicht erneuerbaren Energien



<https://www.bfe.admin.ch/bfe/de/home/news-und-medien/open-government-data/exturi.html/aHR0cHM6Ly9wdWJkYi5IZmUuYWRTaW4uY2gvZGUvcHViGjYX/Rpb24vZG93bmvxYWQvOTgyOQ==.html>

Mehr als 50% Wasserkraft

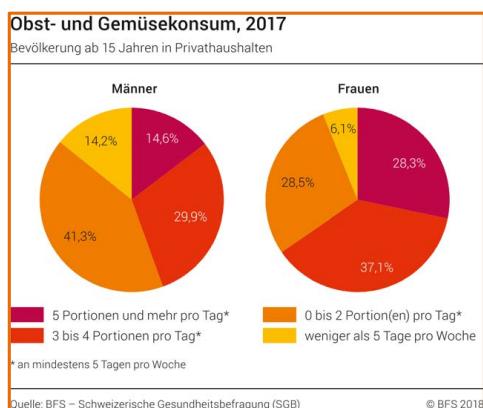
war der Anteil der netto Stromproduktion in der Schweiz 2018, im Vergleich mit anderen erneuerbaren und nicht erneuerbaren Energien



<https://www.bfe.admin.ch/bfe/de/home/news-und-medien/open-government-data/exturi.html/aHR0cHM6Ly9wdWJkYi5IZmUuYWRTaW4uY2gvZGUvcHViGjYX/Rpb24vZG93bmvxYWQvOTgyOQ==.html>

Doppelt soviele Frauen wie Männer

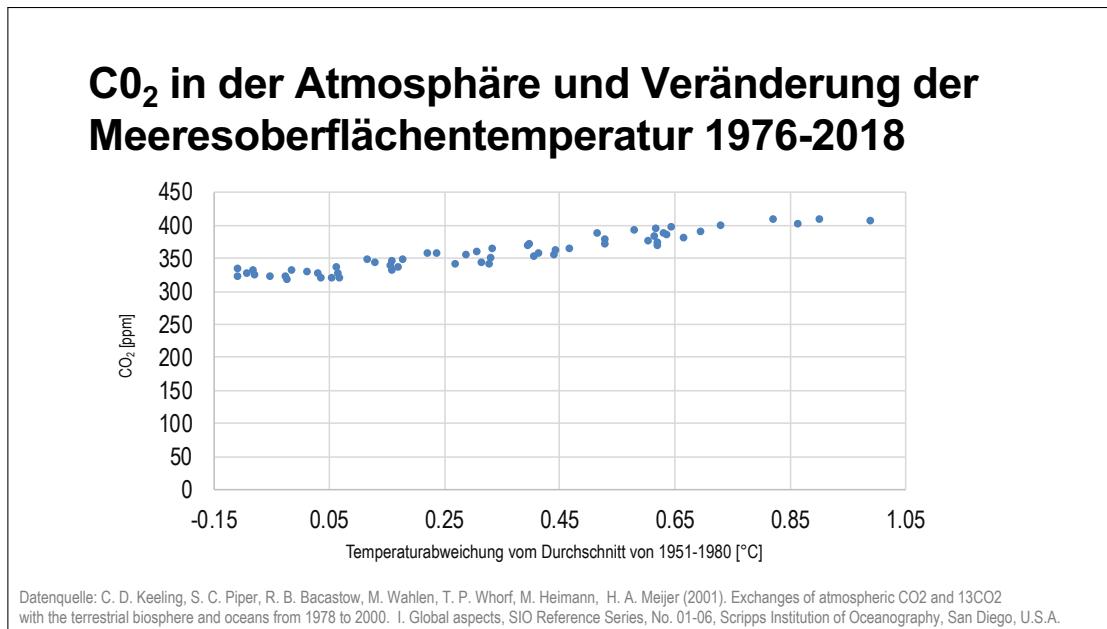
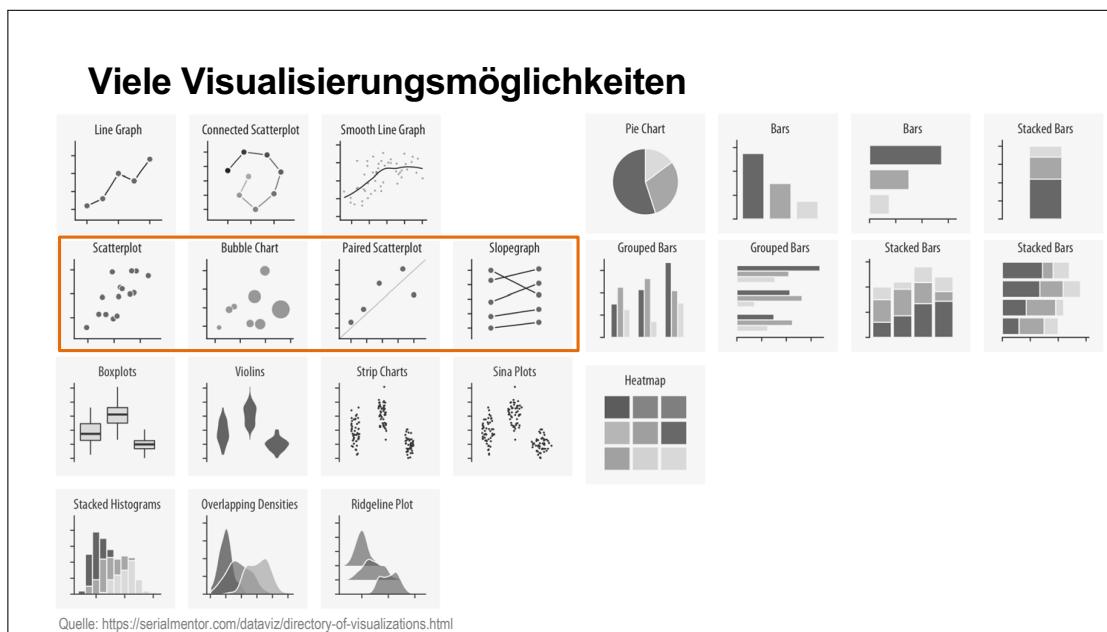
konsumierten in der Schweiz mindestens 5 Portionen Obst- und Gemüse pro Woche in 2017.



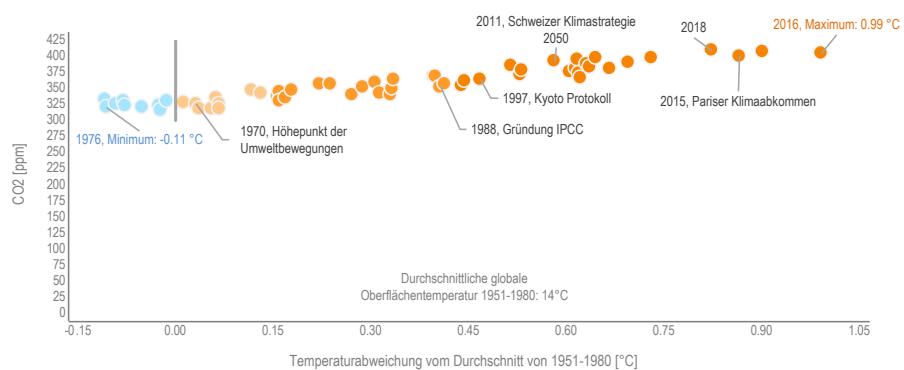
<https://www.bfs.admin.ch/bfs/de/home/statistiken/gesundheit/determinanten/ernaehrung.html>

Wozu visualisiere ich?

1. Mengen
2. Verteilungen
3. Anteile
4. Zusammenhänge

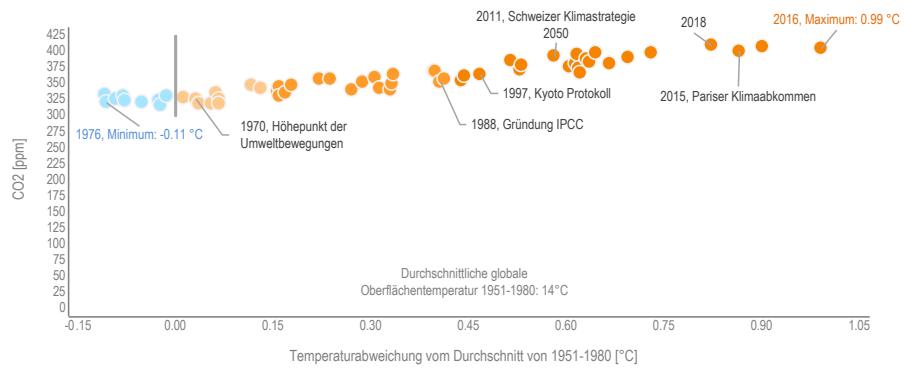


Je mehr CO₂ in der Luft, desto wärmer das Meer



Datenquelle: C. D. Keeling, S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, H. A. Meijer (2001). Exchanges of atmospheric CO₂ and ¹³CO₂ with the terrestrial biosphere and oceans from 1978 to 2000. I. Global aspects, SIO Reference Series, No. 01-06, Scripps Institution of Oceanography, San Diego, U.S.A.

Je mehr CO₂ in der Luft, desto wärmer das Meer



Datenquelle: C. D. Keeling, S. C. Piper, R. B. Bacastow, M. Wahlen, T. P. Whorf, M. Heimann, H. A. Meijer (2001). Exchanges of atmospheric CO₂ and ¹³CO₂ with the terrestrial biosphere and oceans from 1978 to 2000. I. Global aspects, SIO Reference Series, No. 01-06, Scripps Institution of Oceanography, San Diego, U.S.A.

Referenzen

Evergreen, Stephanie (2020). Effective Data Visualization. Second Edition, Sage, Thousand Oaks, (CA), U.S.A.

Hoekstra, A.Y., Chapagain, A.K. Water footprints of nations: Water use by people as a function of their consumption pattern. *Water Resour Manage* 21, 35–48 (2007). <https://doi.org/10.1007/s11269-006-9039-x>, Table 3, p. 42

Wilke, Claus (2019). Fundamentals of Data Visualization. O'Reilly Media, Inc., Sebastopol, (CA), U.S.A.

Weblinks

<https://serialmentor.com/dataviz/>

<https://stephanieevergreen.com/books/>

https://www.pseau.org/outils/ouvrages/waterfootprint_comprehensive_introduction_to_water_footprints_en.pdf

http://www.education21.ch/sites/default/files/uploads/pdf-d/Mystery_Virtuelles-Wasser_Sek-II_de.pdf

<https://vega.github.io>

Referentin

Prof. Sara Irina Fabrikant
Geographisches Institut

© Universität Zürich
Digital Society Initiative

5 Datenmanagement

SVEN HELMER

Studium Digitale Kursbaustein 3: Datenmanagement

Begleit-Skript Dieses Skript bereitet die Inhalte des Kursbaustein 3 schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

5.1 Lektion 1: Wieso eigentlich Datenmanagement?

Die Menschheit produziert tagtäglich Unmengen von Daten, sei es auf Social-Network-Plattformen, in Logdateien auf Webservern (die unser Surfverhalten mitprotokollieren), in Sensoren die inzwischen an vielen Maschinen und Fahrzeugen angebracht sind oder in der Protokollierung von Lieferketten. Schätzungen gehen davon aus, dass wir im Jahr 2025 jeden Tag ca. 463 Exabytes an Daten erzeugen werden, das entspricht dem Inhalt von ungefähr 212 Milliarden DVDs.

Eine derart grosse Menge an Daten speichert man nicht mehr einfach in Dateien auf einem Rechner, das Ganze muss professioneller organisiert werden. Um dieses Thema geht es in diesem Kursbaustein: wie verwalte ich Daten, so dass ich sie später effektiv einsetzen kann?

Datenmanagement kann kurz so definiert werden:

- Es geht um das Sammeln, Speichern und Verarbeiten von Daten
- und das Ganze sollte zuverlässig, sicher und effizient geschehen.

Leider wird mit Daten oft nicht sorgfältig umgegangen, was dazu führt, dass sie später entweder unbrauchbar werden oder mit viel Aufwand rekonstruiert und aufbereitet werden müssen. Eventuell gehen sie sogar ganz verloren.

5.2 Lektion 2: Lebenszyklus von Daten

Die meisten Daten durchlaufen einen typischen Lebenszyklus und leben oft länger als man denkt. Oft werden die Daten auch von anderen Benutzern übernommen und weiterverwendet. Wenn unsere Daten schlecht organisiert und beschrieben sind, wird das schwierig. Mehr über die korrekte Beschreibung von

Daten und Konventionen zur Namensgebung erfahren Sie in einem späteren Kursbaustein.

Wie sieht ein typischer Lebenszyklus von Daten (englisch: data lifecycle) eigentlich aus? Wir schauen uns die wichtigsten Phasen einmal an:

- Phase 1 (Erzeugen): zunächst werden die Daten gesammelt bzw. erzeugt. Das können Logdateien auf einem Webserver sein der Zugriffe mitprotokolliert oder experimentelle Daten die von Sensoren ausgegeben werden. Hier macht es Sinn sich Gedanken zu machen, welche Daten konkret gebraucht werden und wie die Daten organisiert und strukturiert sind (wir kommen auf die Strukturierung von Daten in der Lektion 5 über Datenmodellierung zurück).
- Phase 2 (Vorverarbeiten): oft gibt es Probleme mit der Qualität der gesammelten Daten, z.B. durch falsche Eingaben oder gestörte Sensoren. Die Daten müssen erst einmal gesichtet und eventuell überarbeitet werden, um ihre Qualität zu verbessern. Sie lernen Verfahren aus der Datenanalyse und der Statistik die beim Aufspüren von Fehlern helfen können in einem späteren Kursbaustein kennen.
- Phase 3 (Speichern): wie speichern wir die Daten möglichst geschickt ab, damit sie optimal genutzt werden können? Das bedeutet konkret, dass ich genau das finde was ich suche und das möglichst schnell. In Lektion 4 schauen wir uns an, wie Datenbanksysteme zur Speicherung eingesetzt werden können.
- Phase 4 (Nutzung): Jetzt kann die eigentliche Nutzung der Daten erfolgen. Da dies stark von der Anwendung abhängt, schauen wir uns hier nur ein paar Beispiele an. Lektion 6 zeigt, wie wir auf Daten die in einem Datenbanksystem abgespeichert sind, zugreifen können.
- Phase 5 (Archivierung/Löschen): wenn die Daten nicht mehr aktiv verwendet werden, steht die Archivierung an. Falls benötigt können sie dann zu einem späteren Zeitpunkt wieder abgerufen werden. Es kann aber auch sein, dass die Daten gelöscht werden müssen. Aus rechtlichen Gründen dürfen bestimmte Daten nur eine gewisse Zeit aufbewahrt werden. Mehr zu diesem Thema gibt es in einem anderen Kursbaustein.

5.3 Lektion 3: Spreadsheets

Spreadsheets, neudeutsch für Tabellenkalkulation, war eine der ersten Killer-Apps die für PCs entwickelt wurden. Im Prinzip ist es eine elektronische Umset-

zung von Vorgängen die man aus der Buchhaltung kennt und es ist ein Werkzeug für die Eingabe, Speicherung, Analyse und Visualisierung von Daten. Dabei besteht ein Spreadsheet aus einer Tabelle mit Zeilen und Spalten:

- Die oberste Zeile enthält Beschreibungen der Spalten
- Jede Zeile darunter enthält einen Datensatz
- Jede Spalte steht dabei für eine Variable
- Jede Zelle enthält einen Wert. Dieser Wert kann auch aus anderen Zellwerten berechnet werden.

Die folgende Abbildung zeigt ein Beispiel eines Spreadsheets mit Informationen über die Passagierfrequenz verschiedener Bahnhöfe der SBB.

	A	B	C	D	E	F	G	H
1	Code	Bahnhof Haltestelle	Bezugsjahr	Kanton	DTV	DWV	DNWV	
2	AD	Aadorf	2018	TG	1700	2000	1000	
3	AA	Aarau	2018	AG	37900	44800	22700	
4	ABO	Aarburg-Ottringen	2018	AG	2500	3000	1300	
5	AAT	Aathal	2018	ZH	740	800	610	
6	ACLA	Acla da Fontauna	2018	GR	90	90	80	
7	AE	Aesch	2018	BL	2000	2400	1300	
8	AF	Affoltern am Albis	2018	ZH	6800	8100	4000	
9	AIG	Aigle	2018	VD	6800	7700	5000	
10	AI	Airolo	2018	TI	420	370	510	

Die erste Zeile im Spreadsheet sagt uns, dass die Spalten Informationen über den Bahnhofscode, den Namen des Bahnhofs, das Jahr der Datenerhebung, den Kanton in dem sich der Bahnhof befindet und die Passagierfrequenzen enthält. Dabei steht DTV für durchschnittlicher täglicher Verkehr, DWV für durchschnittlicher Wochentagsverkehr und DNWV für durchschnittlicher Nichtwochentagsverkehr. Die Information über den ersten Bahnhof in der zweiten Zeile des Spreadsheets kann man in natürlicher Sprache folgendermassen zusammenfassen: der Bahnhof Aadorf mit dem Code AD im Kanton Thurgau hatte für das Jahr 2018 folgendes Passagieraufkommen: DTV – 1700, DWV – 2000, DNWV – 1000.

Die Best Practices für den Umgang mit Daten die in einem späteren Kursbaustein detaillierter vorgestellt werden (wie z.B. sinnvolle Namensgebung von Dateien, die Speicherung von Metadaten mit den eigentlichen Daten, das Anlegen von Sicherungskopien, sowie die Verwendung von nicht-proprietären Dateiformaten), gelten auch für Spreadsheets. Zusätzlich dazu sollte man bei Spreadsheets noch folgendes beachten:

- Nur einen Wert pro Zelle eintragen.
- Zellen nicht einfach leer lassen. Sonst weiss man später nicht ob der Wert fehlt, weil es ihn nicht gibt oder ob er nicht bekannt ist.
- Man sollte keine Berechnungen auf den Rohdaten ausführen die die Rohdaten ändern. Solche Berechnungen immer auf Kopien ausführen.
- Um unsinnige Dateneingabe zu verhindern, sollte man die Datenvalidierungsfunktionen von Spreadsheets nutzen: z.B. Datentypen festlegen oder Wertebereiche überprüfen.
- Schauen, ob die Modellierung sinnvoll ist indem man eine Zeile wie oben für den Bahnhof Aadorf in natürlicher Sprache ausdrückt.

Spreadsheets haben eine Reihe von Vorteilen und Nachteilen, schauen wir uns erst die Vorteile an:

- Es ist relativ einfach Daten einzutragen und abzuspeichern.
- Daten können in gängigen Formaten importiert und exportiert werden.

Es gibt aber auch eine Reihe von Nachteilen (wie die ausgebessert werden, sehen wir in der nächsten Lektion über Datenbanksysteme):

- Spreadsheets skalieren nicht besonders gut, d.h. sie haben Schwierigkeiten bei sehr grossen Datenmengen.
- Die Qualität der Daten hängt stark von der Selbstdisziplin der Nutzer ab.
- Es gibt keine Mehrbenutzersynchronisation. Wenn mehrere verschiedene Versionen eines Spreadsheets im Umlauf sind, hat man schnell ein Durcheinander.

5.4 Lektion 4: Datenbanksysteme

Anwendungsentwickler in den 60ern und 70ern haben irgendwann gemerkt, dass sie immer wieder die gleichen Probleme lösen. Einige Entwickler überlegten sich, ein System zu bauen, das diese wiederkehrenden Probleme löst. Auf diese Weise muss nicht jeder Entwickler das Rad neu erfinden. Das senkt die Kosten und verkürzt die Entwicklungszeit von Anwendungen. Damit war die Idee von Datenbanksystemen geboren. Was macht ein Datenbanksystem für uns? Im Folgenden sind die wichtigsten Punkte aufgeführt:

- Erst einmal kann komfortabler auf die Daten zugegriffen werden. Ich muss nicht wissen wie und wo die Daten konkret abgespeichert sind. Ein Datenbanksystem schirmt uns vor den technischen Details ab. Ich kann ja auch Auto fahren, ohne genau zu wissen, wie der Motor funktioniert. Dieses Konzept nennt sich Datenunabhängigkeit.
- Datenbanksysteme stellen auch Anfragesprachen zur Verfügung, die mir helfen, auf die Daten zuzugreifen. Oft sind diese Sprachen deklarativ, d.h. ich sage dem System welche Daten ich brauche. Ich muss nicht wissen, wo und wie diese Daten geholt werden müssen. Darum kümmert sich das System.
- Ausserdem beinhaltet ein Datenbanksystem Verfahren für die Fehlerbehandlung. Wenn ein Datenbanksystem abstürzt, gehen keine Daten verloren. Es wird ständig im Hintergrund mitprotokolliert, was gerade gemacht wird. Dies passiert automatisch, ich muss mich als Benutzer nicht darum kümmern.
- Wenn mehrere Benutzer gleichzeitig Daten verändern kann das zu inkonsistenten Zuständen führen. Wenn z.B. zwei Benutzer eine Datei gleichzeitig öffnen, Änderungen machen und dann die geänderte Datei zurückschreiben, überleben nur die Änderungen die als letztes zurückgeschrieben werden. Ein Datenbanksystem synchronisiert die Zugriffe der verschiedenen Benutzer automatisch und vermeidet inkonsistente Zustände.
- Auch kann ich angeben, wer welche Daten sehen oder gar ändern darf. Ich kann sogar Regeln angeben, auf welche Weise bestimmte Daten geändert werden dürfen. Das Datenbanksystem überwacht auf diese Weise die Integrität der Daten.
- Datenbanksysteme können mit grossen Datenmengen effizient umgehen. Datenbanksysteme können die Daten indexieren, um so schnell

die gewünschte Information zu finden. Dies ist vergleichbar mit einem Index in einem Buch in dem ich Schlagworte und deren Seitennummer nachschlagen kann.

Zusammenfassend zur Funktionalität von Datenbanksystemen lässt sich sagen, dass sie viele der Probleme von Spreadsheets vermeiden.

Datenbanksysteme werden in den verschiedensten Anwendungen eingesetzt, laut Gartner hat der weltweite Markt für Datenbanksysteme eine Grösse von 46 Milliarden US\$. Datenbanksysteme sind oft im Hintergrund tätig, aber fast überall dort zu finden, wo grössere Mengen von Daten systematisch verwaltet werden müssen:

- Bei Banken für die Kontoverwaltung und Transaktionsabwicklung
- Auf Webseiten und Services, wie z.B. Social Networks
- Bei Transportunternehmen
- Selbst auf Smartphones verwenden viele Apps eine Datenbank im Hintergrund

5.5 Lektion 5: Datenmodellierung

Die im Moment am gebräuchlichste Art von Datenbanken sind relationale Datenbanksysteme. Sie speichern die Daten gut strukturiert und sehr systematisch in Tabelle mit Zeilen und Spalten ab. Dabei werden oft auch mathematische Begriffe aus der Relationentheorie verwendet:

- Tupel = Zeile
- Attribut = Spalte
- Tabelle = Relation

Ein weiterer wichtiger Punkt ist, dass eine Relation aus zwei Teilen besteht:

- einem Schema R , das sind die Metadaten
- und einer Instanz R , das sind die eigentlichen Daten, also der Inhalt der Relation

Das folgende Diagramm veranschaulicht das Ganze mit dem SBB-Datensatz. Das Ganze sieht zwar ähnlich wie ein Spreadsheet aus, aber ein entscheidender Unterschied ist, dass das Datenbanksystem die Einhaltung des Schemas durchsetzt.

Schema

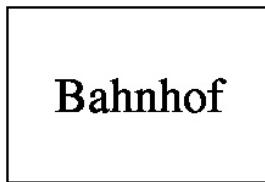
	code character (4)	bahnhof character varying (40)	bezugsjahr integer	kanton character (3)	dtv integer	dwv integer	dnmv integer
1	AD	Aadorf	2018	TG	1700	2000	1000
2	AA	Aarau	2018	AG	37900	44800	22700
3	ABO	Aarburg-Oftringen	2018	AG	2500	3000	1300
4	AAT	Aathal	2018	ZH	740	800	610
5	ACLA	Acla da Fontauna	2018	GR	90	90	80
6	AE	Aesch	2018	BL	2000	2400	1300
7	AF	Affoltern am Albis	2018	ZH	6800	8100	4000
8	AIG	Aigle	2018	VD	6800	7700	5000
9	AI	Airolo	2018	TI	420	370	510
10	ALL	Allaman	2018	VD	3400	4100	1700

Instanz

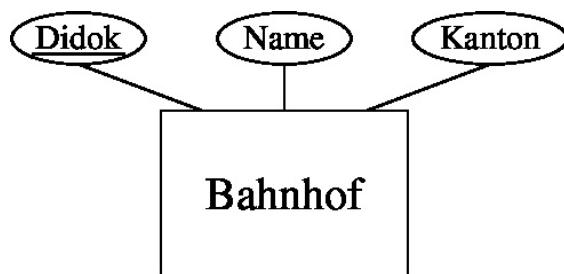
Bevor man anfängt eine Relation anzulegen und mit Daten zu füllen, sollte man sich zuerst einmal Gedanken über die Struktur der Daten machen. Im ersten Schritt entwirft man ein sogenanntes konzeptuelles Schema, was dann in eine oder mehrere Relationen umgesetzt wird. Ein wichtiges konzeptuelles Schema ist das Entity-Relationship-Modell, oder kurz ER-Modell. Dieses Modell stellt grafische Komponenten zur Modellierung zur Verfügung. Das macht es einfacher den Überblick zu wahren und sich mit den Anwendungsexperten auszutauschen. Die gebräuchlichsten Komponenten im ER-Modell sind:

- Entitäten (Entities)
- Attribute
- Schlüssel
- Beziehungen (Relationships)

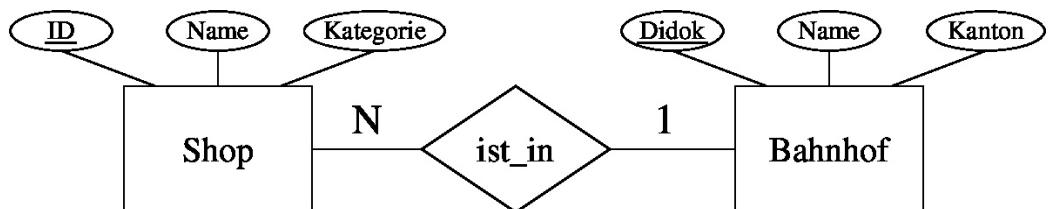
Wir spielen das Ganze nun mit den SBB-Daten durch. Eine Entität ist ein Ding, eine Person, oder ein Konzept. Eine Sammlung von gleichen Entitäten bildet eine (Entitäts-)Menge, die durch ein Rechteck dargestellt wird:



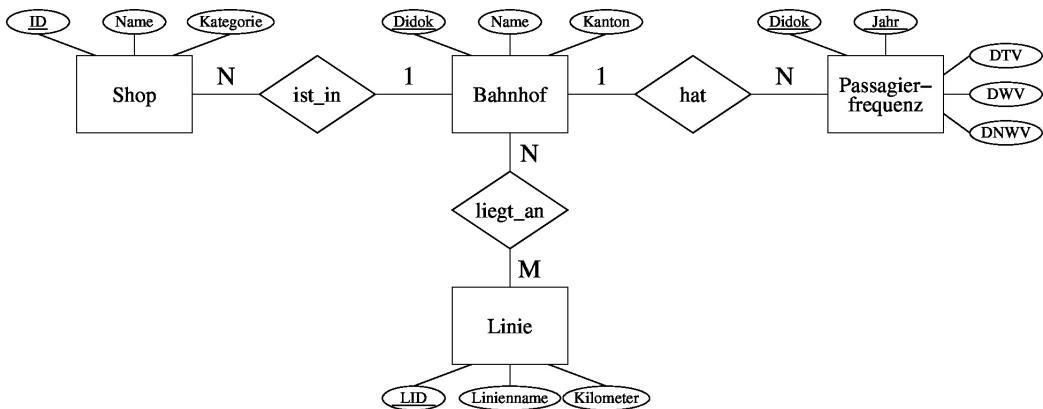
Eigenschaften von Entitäten werden mit Attribut en beschrieben. Ein Attribut wird durch eine Ellipse repräsentiert. Attribute, die eine Entität eindeutig beschreiben, sind Schlüssel und werden im Diagramm unterstrichen:



Eine Beziehung verbindet zwei oder mehr Entitätsmengen und wird mit Hilfe einer Raute dargestellt. Ein Beispiel für eine binäre Beziehung ist die Beziehung zwischen Shop und Bahnhof. Die Symbole N und 1 stehen für Funktionalitäten und sagen aus, dass in einem Bahnhof N, also beliebig viele, Shops sein können, ein (konkreter) Shop aber nur in einem Bahnhof ist:



Das gesamte Modell umfasst noch Linien und Passagierfrequenzen, die über die Beziehungen “hat” und “liegt_an” mit der Entitätsmenge Bahnhof verbunden sind:



Die Umsetzung in Relationen wird in zwei Schritten vorgenommen. Zuerst werden die Entitäten umgesetzt und danach die Beziehungen. Jede Entität wird dabei in eine eigene Relation übersetzt (die jeweiligen Schlüssel sind dabei unterstrichen):

- Bahnhof(Didok, Name, Kanton)
- Shop(ID, Name, Kategorie)
- Passagierfrequenz(Didok, Jahr, DTV, DWV, DNWV)
- Linie(LID, Linienname, Kilometer)

Bei den Beziehungen wird zwischen 1:N und N:M Beziehungen unterschieden. Bei den 1:N Beziehungen wird der Schlüssel der Relation auf die verwiesen wird (die Entität auf der 1-Seite) als Fremdschlüssel in die Relation übernommen (Fremdschlüssel sind kursiv dargestellt):

- Shop(ID, Name, Kategorie, Didok)
- Passagierfrequenz(Didok, Jahr, DTV, DWV, DNWV)
- Die N:M Beziehungen bilden eine eigene Relation:
- Bahnhoflinien(Didok, LID)

5.6 Lektion 6: Datenbankanfragen

SQL (steht für Structured Query Language, auf deutsch: strukturierte Anfrage-sprache) ist eine standardisierte Anfragesprache für relationale Datenbanksysteme. Das bedeutet, dass, von kleineren Ausnahmen abgesehen, alle relationalen Systeme die gleiche Sprache verwenden. SQL ist eine deklarative Sprache, d.h. als Benutzer gebe ich nur an welchen Daten ich interessiert bin, das Datenbanksystem kümmert sich darum, wie die Anfrage verarbeitet wird.

Das Kernstück einer Anfrage ist ein Select-From-Where-Block, in dem die Details der Anfrage spezifiziert werden:

```
select Attributliste  
from Relationenliste  
where Filter;
```

Dabei spielen die drei Teile folgenden Rollen:

- Die Attributliste in der Select-Klausel gibt an, welche Attribute im Ergebnis ausgegeben werden.
- Die Relationenliste in der From-Klausel gibt an, aus welchen Relationen die Daten kommen.
- Mit dem Filter in der Where-Klausel kann spezifiziert werden, welche Daten mich genau interessieren.

Wir spielen das Ganze einmal mit verschiedenen Anfragen an den SBB-Datensatz durch. Eine sehr einfache Anfrage gibt den gesamten Inhalt einer Relation aus:

```
select *  
from Bahnhof;
```

Der Stern (*) ist eine Abkürzung für “gib alle Attribute aus”. Die obige Anfrage gibt den kompletten Inhalt der Relation Bahnhof aus:

Didok	Name	Kanton
3000	Zürich HB	ZH
23	Liestal	BL
1026	Genève Aéroport	GE
...

Anstatt alle Attribute auszugeben, können wir uns auch auf bestimmte Attribute beschränken:

```
select Name, Kanton
from Bahnhof;
```

Im Ergebnis dieser Anfrage fehlt nun die Spalte Didok:

Name	Kanton
Zürich HB	ZH
Liestal	BL
Genève Aéroport	GE
...	...

Die Where-Klausel ist der interessanteste Teil einer Anfrage. Hier können wir z.B. ungewollte Tupel ausfiltern:

```
select *
from Bahnhof
where Kanton = "ZH";
```

Hier sind jetzt wieder alle Spalten dabei, allerdings werden nur die Tupel der Bahnhöfe im Kanton Zürich in das Ergebnis übernommen:

Didok	Name	Kanton
3000	Zürich HB	ZH
3006	Zürich Oerlikon	ZH
...

Bisher haben wir nur Anfragen betrachtet die sich auf eine Relation beziehen. Es kann aber sein, dass die benötigten Daten über mehrere Tabellen verteilt sind. Wir müssen spezifizieren wie Tupel aus verschiedenen Tabellen miteinander verknüpft werden. Auch das geschieht in der Where-Klausel. In der nächsten Anfrage verknüpfen wir die Passagierfrequenzdaten mit den Namen der Bahnhöfe an denen die Statistiken erhoben wurden:

```
select B.Name, P.Jahr, P.DTV
from Bahnhof B, Passagierfrequenz P
where B.Didok = P.Didok;
```

Wir sagen dem Datenbanksystem, dass Tupel aus der Relation Bahnhof nur mit Tupeln aus der Relation Passagierfrequenz verbunden werden dürfen, wenn ihr Wert für Didok übereinstimmt. Die Abkürzungen B und P für die Relationen wurden eingeführt um sich Tipparbeit zu sparen (ansonsten müsste man an allen Stellen an denen B und P auftauchen, den vollen Namen angeben). Das Ergebnis dieser Anfrage sieht folgendermassen aus:

B.Name	P.Jahr	P.DTV
Zürich HB	2018	423600
Liestal	2017	17500
...

Die Filter in einer Where-Klausel können beliebig komplex werden. So kann man das Ergebnis der Verknüpfung der Daten aus den beiden Tabellen Bahnhof und Passagierfrequenz weiter einschränken. In der nächsten Anfrage sind wir nur an den Daten aus dem Kanton Zürich für das Jahr 2018 interessiert:

```
select B.Name, P.Jahr, P.DTV
from Bahnhof B, Passagierfrequenz P
```

```
where B.Didok = P.Didok
and P.Jahr = 2018
and B.Kanton = "ZH";
```

Die einzelnen Filterausdrücke werden mit dem Schlüsselwort "and" verknüpft. Es ist auch möglich, Bedingungen mit den logischen Operatoren "or" oder "not" zu formulieren. Für diese Anfrage sähe das Ergebnis folgendermassen aus:

B.Name	P.Jahr	P.DTV
Zürich HB	2018	423600
Zürich Oerlikon	2018	80000
...

Das war jetzt nur ein kurzer Überblick über die Funktionalität von SQL. Es ist noch viel mehr möglich, z.B. das Gruppieren von Daten und das Erstellen einfacher Statistiken. Als einfaches Beispiel wird hier noch das Sortieren von Daten mit Hilfe des Schlüsselworts "order by" gezeigt:

```
select B.Name, P.Jahr, P.DTV
from Bahnhof B, Passagierfrequenz P
where B.Didok = P.Didok
and P.Jahr = 2018
order by P.DTV desc;
```

"desc" steht für descending, auf deutsch absteigend. Man kann auch aufsteigend sortieren: dies geschieht mit dem Schlüsselwort "asc". Wenn man weder "asc" noch "desc" angibt, wird automatisch aufsteigend sortiert. Hier noch das Ergebnis der Beispielanfrage:

B.Name	P.Jahr	P.DTV
Zürich HB	2018	423600
Bern	2018	184000
...

Bisher haben wir nur Anfragen betrachtet. Bevor wir aber Anfragen an ein Datenbanksystem stellen können, müssen wir zuerst Relationen anlegen und mit Inhalt füllen. Eventuell wollen wir später auch wieder Daten ändern oder sogar löschen. Auch das ist mit SQL möglich.

Das Schema einer Relation wird mit dem Kommando “create table” angelegt. Um z.B. die Relation für Bahnhöfe anzulegen, wird folgender Befehl abgesetzt:

```
create table Bahnhof (
    Didok integer,
    Name varchar(80),
    Kanton char(2),
    primary key (Didok)
)
```

Dabei wird für jedes Attribut angegeben, welchen Datentyp es hat: Ganzzahlen für Didok, eine Zeichenkette variabler Länge (maximal 80 Zeichen) für Name und eine Zeichenkette der Länge 2 für Kanton. Es kann ausserdem angegeben werden, welches Attribut den Schlüssel bildet.

Um Daten einzufügen wird der Befehl “insert” verwendet. Man gibt den Namen der Relation in die man etwas einfügen möchte an, gefolgt von den Werten:

```
insert into Bahnhof
values (3000, 'Zürich', 'ZH');
```

Um Daten wieder zu löschen verwendet man den Befehl “delete”. Dieser Befehl hat, ähnlich wie eine Anfrage, eine From- und eine Where-Klausel, mit denen man angibt welche Daten gelöscht werden sollen, hier z.B. den Bahnhof mit Didok 3000:

```
delete from Bahnhof
where Didok = 3000;
```

Man sollte beim Löschen allerdings vorsichtig sein. Bei einer leeren Where-Klausel wird der gesamte Inhalt der Relation gelöscht. Wenn man sich nicht ganz sicher ist, kann man die From- und Where-Klausel eines Delete-Befehls

einfach vorher mit einer Select-Klausel ausführen. Dann sieht man, welche Daten gelöscht würden.

Zu guter Letzt können Daten in einer Relation auch geändert werden. Der folgende Befehl weist dem Attribut Name des Tupels mit der Didok 3000 den Wert "Zürich HB" zu:

```
update Bahnhof  
set Name = "Zürich HB"  
where Didok = 3000;
```

5.7 Lektion 7: Big Data und die Cloud

Ein Laptop- oder Desktopcomputer ist gut geeignet um Spreadsheets zu erstellen und zu analysieren, um einen Text zu editieren oder um einen Film zu schauen. Wenn die Datenmengen allerdings etwas grösser werden, stösst man schnell an die Grenzen. Um mit grossen Datenmengen, Schlagwort "Big Data", zurechtzukommen gibt es im Prinzip zwei Möglichkeiten:

- Scale-Up: man kauft sich einen grösseren Rechner mit mehr Speicherplatz und einem stärkeren Prozessor. Das wird auf deutsch auch gelegentlich als vertikale Skalierung bezeichnet.
- Scale-Out: hier verteilt man die Arbeit auf viele (kleinere) Maschinen. Oft kommen handelsübliche PCs mit Linux zum Einsatz. Diese Methode wird auch horizontale Skalierung genannt.

Der Vorteil von Scale-Up ist, dass kaum etwas geändert werden muss. Im Prinzip läuft die gleiche Software einfach auf einem grössseren Rechner. Ein Nachteil ist, dass dies nicht beliebig weit getrieben werden kann. Irgendwann hat man die grösste Ausbaustufe erreicht. Beim Scale-Out gibt es (theoretisch) keine Grenzen: wenn man mehr Kapazitäten benötigt, kauft man einfach Geräte dazu. So gibt es Datenzentren mit zehn- oder sogar hunderttausenden von Maschinen. Ein Nachteil ist, dass man die Software anpassen und parallelisieren muss: die Rechenprozesse auf den verschiedenen Computern müssen synchronisiert werden, was bei vielen Rechnern zu einem erhöhten Kommunikationsaufwand führt. Außerdem benötigt man ausgefeilte Mechanismen zur Ausfallsicherheit: bei so vielen Maschinen geht eigentlich ständig etwas kaputt.

Anfang der 2000er-Jahre entwickelten Firmen wie Google und Yahoo Systeme, wie z.B. das Google File System, Map-Reduce und den Vorläufer von Hadoop,

um mit extrem grossen Datenmengen zurechtzukommen. Viele dieser System, wie z.B. Hadoop, gibt es inzwischen in einer frei erhältlichen Open-Source-Variante. Inzwischen gibt es auch Systeme die einfacher und komfortabler zu bedienen sind, wie z.B. Apache Spark. Auch Datenbanksysteme durchlaufen im Moment einige Änderungen, mit den sogenannten NoSQL-Systemen wird versucht Datenbanksysteme zu entwickeln, die für einen Scale-Out geeignet sind.

Auch wenn man selbst kein Rechenzentrum zuhause hat, kann man Datenanalysen auf grossen Datenmengen durchführen. Viele Cloudanbieter haben Plattformen entwickelt, auf denen man sich Ressourcen mieten kann. Beispiele sind AWS (Amazon Cloud Service) und EC2 (Elastic Compute Cloud) von Amazon, DataProc von Google und Azure von Microsoft.

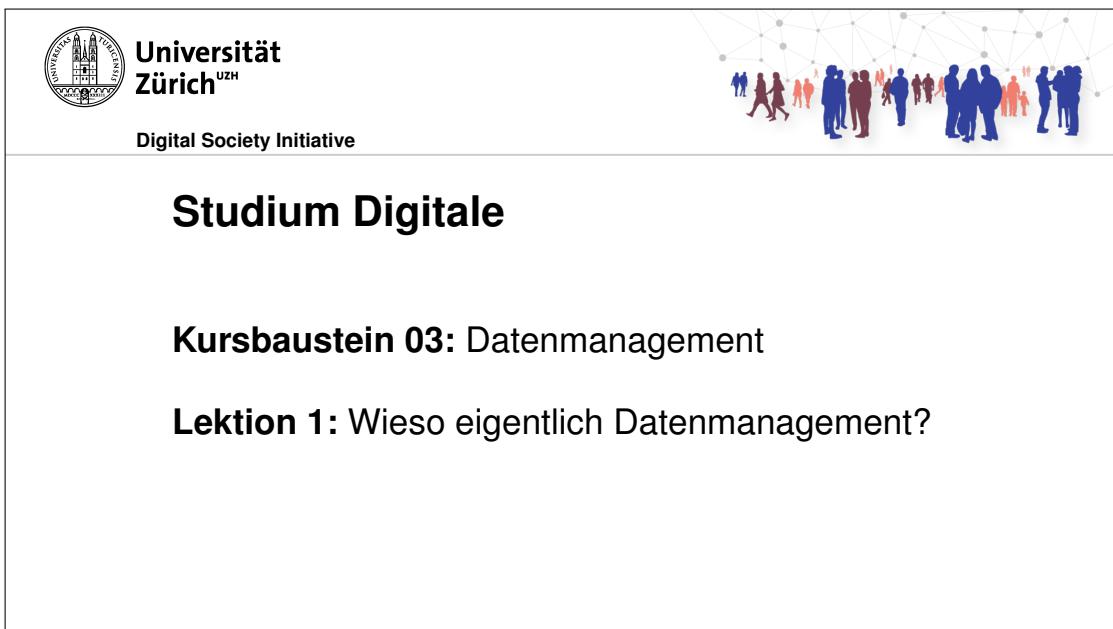
5.8 Lektion 8: Rückblick und Ausblick

In diesem Kursbaustein haben Sie einen Überblick über den Themenbereich Datenmanagement erhalten und Sie wissen jetzt:

- mit welchen Datenmengen wir heutzutage konfrontiert werden und dass es wichtig ist, mit diesen Daten sorgfältig umzugehen,
- welchen Lebenszyklus Daten durchlaufen,
- was Spreadsheets sind und wie sie eingesetzt werden,
- welche Vorteile Datenbanksysteme haben,
- wie Daten modelliert werden,
- wie Daten in einem Datenbanksystem abgefragt werden
- und was es mit Big-Data auf sich hat.

Es gibt noch viel mehr zu erzählen, aber wir haben nur begrenzt Zeit. Eventuell hören Sie im Rahmen Ihres Studiums noch weitere, tiefergehende Vorlesungen zu diesem Thema. Auch wenn im Moment nicht klar ist, wie zukünftige Technologien in diesem Bereich aussehen werden, eins ist sicher: das Thema Datenmanagement wird noch eine ganze Weile relevant sein.

5.9 Folien



The banner features the University of Zurich logo (a circular seal with a building and text) next to the text "Universität Zürich UZH". Below this is the "Digital Society Initiative" logo, which consists of a network of colored dots (grey, red, blue) connected by lines, with silhouettes of people of different colors (red, blue, grey) standing on top of the network.

Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 1: Wieso eigentlich Datenmanagement?

Produzierte Daten pro Tag

- 500 Mill. Tweets
- 294 Mrd. E-Mails
- Facebook: 4PByte Daten ~ 850'000 DVDs

(Raconteur 2019)

Schätzungen für 2025

- 463 Exabytes pro Tag
- knapp 100 Mrd. DVDs

Um was geht es bei Datenmanagement?

Kurz gesagt:

- Es geht um das Sammeln, Speichern und Verarbeiten von Daten
- Das Ganze sollte zuverlässig, sicher und effizient geschehen
- Macht auch schon Sinn für weniger als Millionen von DVDs...

Was geht mich das an?

- Forschungsprojekte
- Industrie 4.0

Was gibt es zu beachten?

- Wie finde ich, was ich suche?
- Hilfe, mein Rechner stürzt ab!

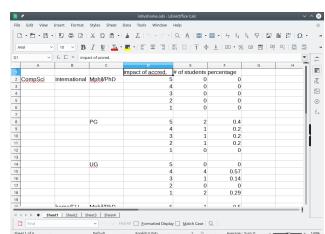
Dateiformate

- Textdateien
- Worddateien
- Spreadsheets
- PDFs
- JSON
- XML
- HTML
- ...

Was mache ich hiermit?

2	5.3	$\frac{1}{2}$	z	δ
3	4.6	—	a	γ
2	10.2	$\frac{3}{4}$	r	δ
...

Mehrere Benutzer



Sicherheit

- Datenschutz
- Privatsphäre

Übersicht über diesen Kursbaustein:

- Was gibt es zu beachten?
- Lebenszyklus von Daten
- Professionelle Werkzeuge
- Datenmodellierung
- Datenbankanfragen
- Big Data und die Cloud

Referenzen

Raconteur (2019). *A Day in Data*.

<https://www.raconteur.net/infographics/a-day-in-data>.

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



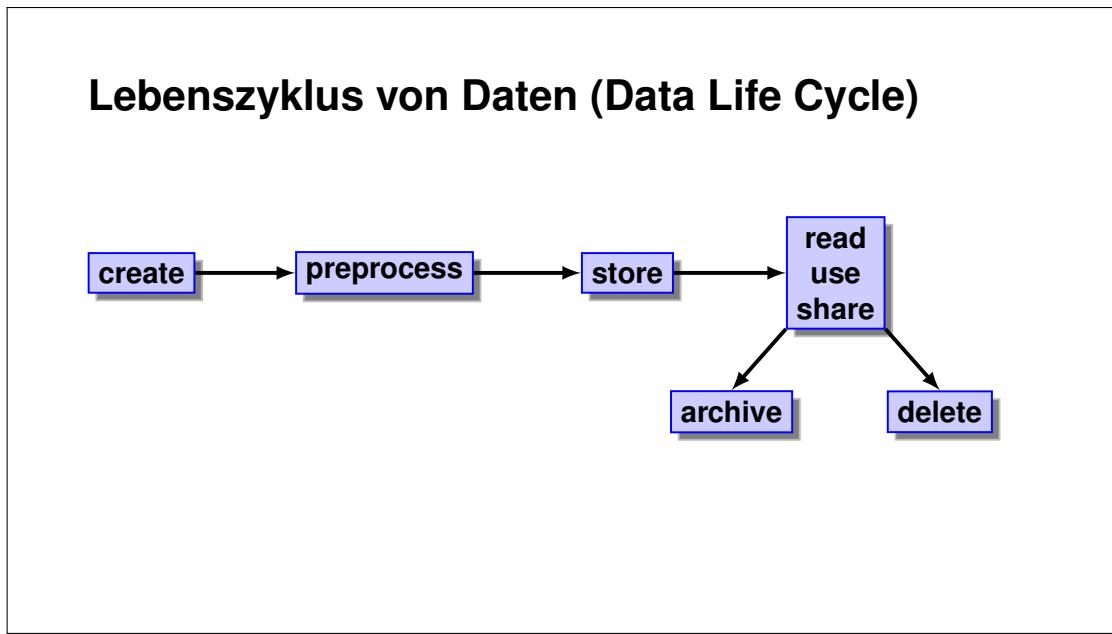
Universität
Zürich UZH

Digital Society Initiative

Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 2: Lebenszyklus von Daten



1. Daten erzeugen

- Digitalisierung
- Logdateien
- Sensoren

2. Daten vorverarbeiten

- Qualitätsprobleme

3. Daten speichern

- müssen auffindbar sein
- und möglichst schnell

4. Daten nutzen**5. Archivierung/Lösung**

- keine aktive Verwendung
- rechtliche Gründe

Viele Daten leben lange

- Wegen Langlebigkeit macht sorgfältiges Datenmanagement Sinn
- Daten können für andere Leute interessant sein:
 - SBB-Daten: <https://data.sbb.ch>
 - Open-Data-Initiativen: <https://opendata.swiss>
 - Interessante Kombinationen

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich^{UZH}

Digital Society Initiative



Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 3: Spreadsheets

Erster Spreadsheet

VisiCalc (siehe auch Grad 2007)

- Eine Idee Ende 1978
- von D. Bricklin und B. Frankston
- Vorläufer von Excel

Was ist eigentlich ein Spreadsheet?

Kurz gesagt: ein Werkzeug für die

- Eingabe,
- Speicherung,
- Analyse und
- Visualisierung

von Daten.

Aufbau eines Spreadsheets

Ein Spreadsheet besteht aus einer Tabelle mit Zeilen und Spalten:

- oberste Zeile: Beschreibungen
- jede Zeile darunter: ein Datensatz
- jede Spalte: eine Variable
- jede Zelle: ein Wert (kann auch berechnet werden)

Ein Beispiel

	A	B	C	D	E	F	G	H
1	Code	Bahnhof	Haltestelle	Bezugsjahr	Kanton	DTV	DWV	DNWV
2	AD	Aadorf		2018	TG	1700	2000	1000
3	AA	Aarau		2018	AG	37900	44800	22700
4	ABO	Aarburg-Ottringen		2018	AG	2500	3000	1300
5	AAT	Aathal		2018	ZH	740	800	610
6	ACLA	Acia da Fontauna		2018	GR	90	90	80
7	AE	Aesch		2018	BL	2000	2400	1300
8	AF	Affoltern am Albis		2018	ZH	6800	8100	4000
9	AIG	Aigle		2018	VD	6800	7700	5000
10	AI	Airolo		2018	TI	420	370	510

Generelle Best Practices

- Namensgebung
- Metadaten anlegen
- Backups
- Als Text abspeichern

Best Practices für Spreadsheets

- Nur ein Wert pro Zelle
- Zellen nicht einfach leer lassen
- Keine Berechnungen auf den Rohdaten
- Datenvielfältigung benutzen
- Einfach einmal einen Datensatz in einem Satz zusammenfassen

(aus Broman und Woo 2018)

Beispiel

Code	Bahnhof	Haltestelle	Bezugsjahr	Kanton	DTV	DWV	DNWV
AD	Aadorf		2018	TG	1700	2000	1000
AA	Aarau		2018	AG	37900	44800	22700
ABO	Aarburg-Ottringen		2018	AG	2500	3000	1300
AAT	Aathal		2018	ZH	740	800	610
ACLA	Acla da Fontana		2018	GR	90	90	80
AE	Aesch		2018	BL	2000	2400	1300
AF	Affoltern am Albis		2018	ZH	6800	8100	4000
AIG	Aigle		2018	VD	6800	7700	5000
AI	Airolo		2018	TI	420	370	510

Vorteile und Grenzen von Spreadsheets

- + Es ist einfach Daten einzutragen und loszulegen
- + Daten können auch importiert und exportiert werden
- Bei Millionen von Datensätzen wird es mühsam
- Probleme mit der Datenqualität
- Keine Mehrbenutzersynchronisation

Referenzen

- Broman, Karl W. und Kara H. Woo (2018). "Data Organization in Spreadsheets". In: *The American Statistician* 72.1, S. 2–10.
- Grad, B. (2007). "The Creation and the Demise of VisiCalc". In: *IEEE Annals of the History of Computing* 29.3, S. 20–31.

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich^{UZH}

Digital Society Initiative



Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 4: Datenbanksysteme

Ursprung von Datenbanksystemen

60er und 70er: Immer wieder die gleichen Probleme

Warum kein System bauen, das diese Probleme löst?

- Senkt Kosten und Entwicklungszeit
- Die Idee von Datenbanksystemen war geboren

Was macht ein Datenbanksystem für uns?

- Datenunabhängigkeit
- Anfragesprachen
- Fehlerbehandlung
- Mehrbenutzersynchronisation
- Datenintegrität
- Skalierbarkeit

(siehe auch Kemper und Eickler 2015)

Bekannte Systeme

- Oracle
- Microsoft SQLServer
- IBM DB2
- MySQL
- PostgreSQL

Wo stecken diese ganzen Systeme?

- Banken und andere Finanzunternehmen
- Verkehrsunternehmen
- Im Backend von Apps
- eigentlich überall dort wo Daten verarbeitet werden
- Datenbanken sind eine Selbstverständlichkeit geworden

Referenzen

Kemper, Alfons und André Eickler (2015). *Datenbanksysteme – Eine Einführung*. 10. Aufl. Walter de Gruyter.

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich^{UZH}

Digital Society Initiative



Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 5: Datenmodellierung

Relationale Datenbanken

In einem relationalen Datenbanksystem

- werden die Daten in Tabellen abgespeichert.
- Die Tabellen bestehen aus Zeilen und Spalten.

Es werden oft auch mathematische Begriffe verwendet:

- Tupel = Zeile
- Attribut = Spalte
- Relation = Tabelle

(siehe auch Kemper und Eickler 2015)

Beispiel (SBB):

	code character (4)	bahnhof character varying (40)	bezugsjahr integer	kanton character (3)	dtv integer	dwv integer	dnmv integer
1	AD	Aadorf	2018	TG	1700	2000	1000
2	AA	Aarau	2018	AG	37900	44800	22700
3	ABO	Aarburg-Oftringen	2018	AG	2500	3000	1300
4	AAT	Aathal	2018	ZH	740	800	610
5	ACLA	Acla da Fontauna	2018	GR	90	90	80
6	AE	Aesch	2018	BL	2000	2400	1300
7	AF	Affoltern am Albis	2018	ZH	6800	8100	4000
8	AIG	Aigle	2018	VD	6800	7700	5000
9	AI	Airolo	2018	TI	420	370	510
10	ALL	Allaman	2018	VD	3400	4100	1700

Aufbau von Relationen

Eine Relation besteht aus zwei Teilen:

- einem Schema \mathcal{R} , das sind die Metadaten
- einer Instanz R , das ist der Inhalt der Relation

Beispiel:

Schema

	code character (4)	bahnhof character varying (40)	bezugsjahr integer	kanton character (3)	dtv integer	dwv integer	dnmv integer
1	AD	Aadorf	2018	TG	1700	2000	1000
2	AA	Aarau	2018	AG	37900	44800	22700
3	ABO	Aarburg-Oftringen	2018	AG	2500	3000	1300
4	AAT	Aathal	2018	ZH	740	800	610
5	ACLA	Acla da Fontauna	2018	GR	90	90	80
6	AE	Aesch	2018	BL	2000	2400	1300
7	AF	Affoltern am Albis	2018	ZH	6800	8100	4000
8	AIG	Aigle	2018	VD	6800	7700	5000
9	AI	Airolo	2018	TI	420	370	510
10	ALL	Allaman	2018	VD	3400	4100	1700

Instanz

Datenbankentwurf

- Zuerst konzeptuelles Schema
- Dann Umsetzung in Relationen
- Ein wichtiges konzeptuelles Modell:
 - Entity-Relationship-Modell

Entity-Relationship-Modell

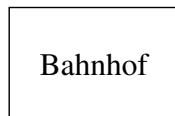
Das Entity-Relationship-Modell hat folgende Grundbausteine:

- Entitäten (Entities)
- Attribute
- Schlüssel
- Beziehungen (Relationships)

Wir spielen das einmal mit dem SBB-Beispiel durch

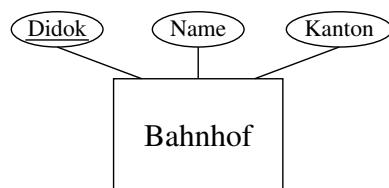
Entitäten

- Eine *Entität* ist ein Ding, eine Person, ein Ort oder ein Konzept
- Eine Sammlung von gleichen Entitäten bildet eine *Menge*
- Entitätsmengen werden durch Rechtecke dargestellt:



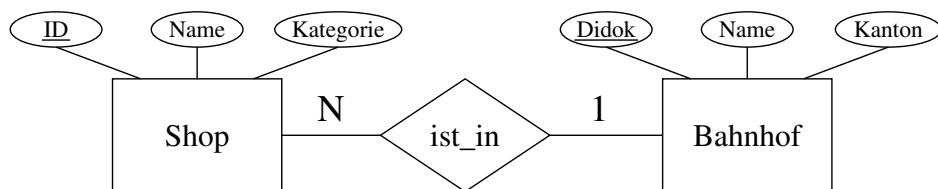
Attribute

- Eigenschaften von Entitäten werden mit *Attributen* beschrieben
- Ein Attribut, das eine Entität eindeutig beschreibt, ist ein *Schlüssel*
- Attribute werden durch Ellipsen repräsentiert:

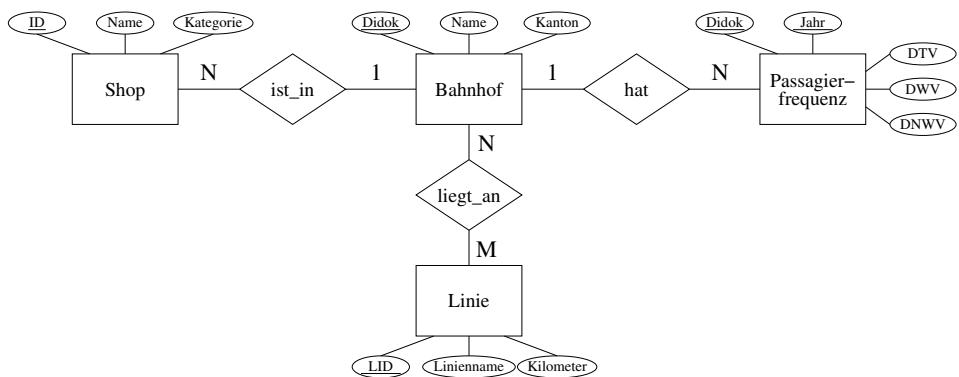


Beziehungen

- Eine *Beziehung* verbindet zwei oder mehr Entitätsmengen
- Darstellung durch eine Raute mit Funktionalitäten
- Beispiel für eine binäre Beziehung



Und nun alles zusammen



Umsetzung in Tabellen

- Bahnhof (Didok, Name, Kanton)
- Shop (ID, Name, Kategorie)
- Passagierfrequenz (Didok, Jahr, DTV, DWV, DNWV)
- Linie (LID, Liniename, Kilometer)

Was machen wir mit den Beziehungen?

Beziehungen

1:N

- Shop (ID, Name, Kategorie, *Didok*)
- Passagierfrequenz (*Didok*, Jahr, DTV, DWV, DNWV)

N:M

- Bahnhoflinien (*Didok*, LID)

Und nun alles zusammen

- Bahnhof (Didok, Name, Kanton)
- Shop (ID, Name, Kategorie, *Didok*)
- Passagierfrequenz (*Didok*, Jahr, DTV, DWV, DNWV)
- Linie (LID, Liniename, Kilometer)
- Bahnhoflinien (*Didok*, LID)

Referenzen

Kemper, Alfons und André Eickler (2015). *Datenbanksysteme – Eine Einführung*. 10. Aufl. Walter de Gruyter.

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich^{UZH}

Digital Society Initiative



Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 6: Datenbankanfragen

SQL

- SQL steht für Structured Query Language
- und ist **die** Anfragesprache für relationale Datenbanksysteme

SQL-Anfragen

SQL-Anfragen bestehen im Wesentlichen aus drei Teilen:
Select, From, und Where

select	Attributliste
from	Relationenliste
where	Filter;

Ein einfaches Beispiel

Wir möchten die gesamte Bahnhofrelation ausgeben:

```
select      *
from       Bahnhof;
```

Didok	Name	Kanton
3000	Zürich HB	ZH
23	Liestal	BL
1026	Genève-Aéroport	GE
...

Attribute selektieren

Wir können uns auch auf bestimmte Attribute beschränken:

```
select      Name, Kanton  
from       Bahnhof;
```

Name	Kanton
Zürich HB	ZH
Liestal	BL
Genève-Aéroport	GE
...	...

Die Where-Klausel

Mit der Where-Klausel können wir Tupel ausfiltern:

```
select      *  
from       Bahnhof  
where      Kanton = "ZH";
```

Didok	Name	Kanton
3000	Zürich HB	ZH
3006	Zürich Oerlikon	ZH
...

Relationen verbinden

Informationen können über mehrere Relationen verteilt sein.
Dann müssen wir die Informationen zusammensetzen:

```
select      B.Name, P.Jahr, P.DTV
from        Bahnhof B, Passagierfrequenz P
where       B.Didok = P.Didok;
```

B.Name	P.Jahr	P.DTV
Zürich HB	2018	423600
Liestal	2017	17500
...

Kombination mit anderen Filtern

```
select      B.Name, P.Jahr, P.DTV
from        Bahnhof B, Passagierfrequenz P
where       B.Didok = P.Didok
and         P.Jahr = 2018
and         B.Kanton = "ZH";
```

B.Name	P.Jahr	P.DTV
Zürich HB	2018	423600
Zürich Oerlikon	2018	80000
...

Sortieren

```
select      B.Name, P.Jahr, P.DTV
from        Bahnhof B, Passagierfrequenz P
where       B.Didok = P.Didok
and         P.Jahr = 2018
order by    P.DTV desc;
```

B.Name	P.Jahr	P.DTV
Zürich HB	2018	423600
Bern	2018	184000
...

Was geht noch?

Gruppieren und einfache Statistiken, z.B.

- Summe
- Anzahl
- Durchschnitt
- Max/Min

Daten ändern (siehe nächste Folien)

- Relationen anlegen
- Daten einfügen
- Daten löschen
- Daten aktualisieren

Relation anlegen

Dafür müssen wir ihr Schema definieren:

```
create table Bahnhof (
    Didok      integer,
    Name       varchar(80),
    Kanton     char(2),
    primary key (Didok)
);
```

Daten einfügen

Jetzt können wir Daten einfügen:

```
insert into Bahnhof
values (3000, 'Zürich', 'ZH');
```

Daten löschen

Wir können Daten auch wieder loswerden:

```
delete from Bahnhof  
where Didok = 3000;
```

Daten löschen

Wir können Daten auch wieder loswerden:

```
delete from Bahnhof  
where Didok = 3000;
```

Vorsicht: bei leerer Where-Klausel wird alles gelöscht!

Daten aktualisieren

Daten können auch korrigiert/aktualisiert werden:

```
update Bahnhof  
set     Name = 'Zürich HB'  
where   Didok = 3000;
```

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich^{UZH}

Digital Society Initiative

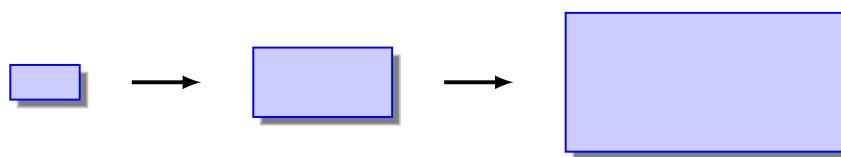


Studium Digitale

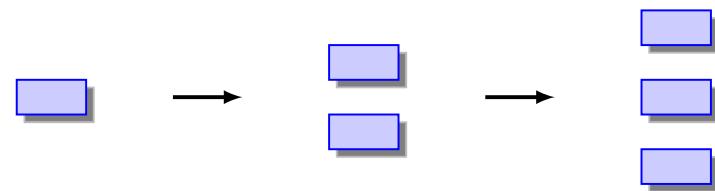
Kursbaustein 03: Datenmanagement

Lektion 7: Big Data und die Cloud

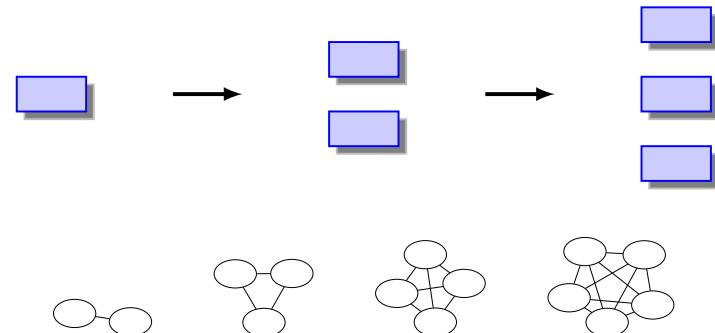
Scale-Up



Scale-Out



Scale-Out



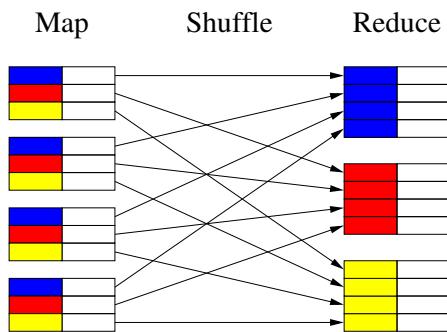
Ende 1990er

- Firmen wie Amazon, Ebay, Google, Yahoo produzieren riesige Datenmengen
- Diese Daten passen nicht mehr auf einen einzelnen Computer

Fehlertolerante Systeme

- Google entwickelt das Google File System und Map-Reduce
- Yahoo: Hadoop Open-Source Framework (samt Dateisystem)
- Apache: Spark
- Parallel zu allem: NoSQL-Datenbanksysteme

Map-Reduce



Beispiel

Map	Shuffle	Reduce
Dokument 1: Heile, heile Segen	(Heile, 1) (heile, 1) (Segen, 1)	a - i: (alles, 1) (heile, 3)
Dokument 2: sieben Tage Regen	(sieben, 1) (Tage, 1) (Regen, 1)	j - s: (Regen, 1) (Segen, 1) (sein, 1)
Dokument 3: sieben Tage Sonnenschein	(sieben, 1) (Tage, 1) (Sonnenschein, 1)	(sieben, 2) (Sonnenschein, 1)
Dokument 4: wird alles wieder heile sein	(wird, 1) (alles, 1) (wieder, 1) (heile, 1) (sein, 1)	t - z: (Tage, 2) (wieder, 1) (wird, 1)

Big-Data Analytics

- Die Verarbeitung von grossen Datenmengen ist heute für fast jeden/jede möglich
- Ressourcen können gemietet werden:
 - Amazon (AWS, EC2)
 - Google Cloud DataProc
 - Microsoft Azure HDInsight
 - (Azure) Databricks

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Universität
Zürich^{UZH}

Digital Society Initiative



Studium Digitale

Kursbaustein 03: Datenmanagement

Lektion 8: Rückblick und Ausblick

Kurze Geschichte des Datenmanagements

3400 - 3100 v.Chr.(?):	Erfindung der Schrift
2000 v.Chr.:	sumerische Tontafellisten
300 v.Chr.:	Bibliothek von Alexandria
1440:	Erfindung des Buchdrucks
16. Jhd.:	erste Buchindizes

Buchindex

Antilope	Adler	Ader
Ameise	Ader	Adler
Ader	Ameise	Ameise
Ansicht	Ansicht	Ansage
Adler	Antilope	Ansicht
Ansage	Ansage	Antilope

B...

Referent

Dr. Sven Helmer
Institut für Informatik

© Universität Zürich
Digital Society Initiative

6 Digitalisierung

PROF. ABRAHAM BERNSTEIN

6.1 Digitale Daten



The logo of the University of Zurich (UZH) is located on the left side of the slide. It consists of a circular seal with a building and text around it, followed by the text "Universität Zürich UZH". To the right of the logo is a graphic illustration of a network or social structure, showing a group of stylized human figures connected by a web of lines and dots.

Studium Digitale

Kursbaustein 2: Digitalisierung von Information



Universität
Zürich^{UZH}

Digital Society Initiative



Studium Digitale

Kursbaustein 2: Digitalisierung von Information

Lektion 1: Digitale Daten

Prof. Abraham Bernstein
UZH, Institut für Informatik und Digital Society Initiative

Diese Lektion:

- Einleitung
- Digitalisierung von Text
- Digitalisierung von Ton
- Digitalisierung von Bild

Diese Lektion:

- **Einleitung**
- Digitalisierung von Text
- Digitalisierung von Ton
- Digitalisierung von Bild

Information kann in verschiedenen Ausprägungen vorliegen:



Ton



Bild, Film



Text

Der Mensch nimmt Informationen mit Sinnesorganen auf. Computer benötigen diese in einem Binärformat.

By Jack Cai, CC BY 3.0, softicons.com

Was ist Binär? Von Dezimal- zu Binärzahlen

Computer kennen nur zwei Grundzustände:

Strom fliesst oder kein Strom fliesst. Dies wird als 1 und 0 abstrahiert.

Wir sind gewöhnt, im Dezimalsystem zu rechnen:

...	1000	100	10	1
	10^3	10^2	10^1	10^0

Beispiel

$$\begin{array}{cccc} 2 & 0 & 1 & 3 \\ 2 \times 10^3 & 0 \times 10^2 & 1 \times 10^1 & 3 \times 10^0 \end{array}$$

Von Dezimal- zu Binärzahlen

Computer benutzen das Binärsystem:

...	128	64	32	16	8	4	2	1
	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Beispiel

$$\begin{array}{cccccccccccc} 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 \times 2^{10} & 1 \times 2^9 & 1 \times 2^8 & 1 \times 2^7 & 1 \times 2^6 & 0 \times 2^5 & 1 \times 2^4 & 1 \times 2^3 & 1 \times 2^2 & 0 \times 2^1 & 1 \times 2^0 \end{array}$$

Von Dezimal- zu Binärzahlen

Umwandeln von Dezimalzahl 333 in Binärzahl:

Von Dezimal- zu Binärzahlen

Rechenbeispiel: Umwandeln von Dezimalzahl 333 in Binärzahl:

Dezimal	333 : 2 =	166	Rest 1	
	166 : 2 =	83	Rest 0	
	83 : 2 =	41	Rest 1	
	41 : 2 =	20	Rest 1	
	20 : 2 =	10	Rest 0	
	10 : 2 =	5	Rest 0	
	5 : 2 =	2	Rest 1	
	2 : 2 =	1	Rest 0	

Dual ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓

1 0 1 0 0 1 1 0 1

Von Dezimal- zu Binärzahlen

Umwandlung der Binärzahl 1101101_2 in eine Dezimalzahl:

Von Dezimal- zu Binärzahlen

Umwandlung der Binärzahl 1101101, in eine Dezimalzahl:

1	1	0	1	1	0	1
2^6	2^5	2^4	2^3	2^2	2^1	2^0
$1 * 2^6$	$1 * 2^5$	$0 * 2^4$	$1 * 2^3$	$1 * 2^2$	$0 * 2^1$	$1 * 2^0$
64	32	0	8	4	0	1

$$64 + 32 + 8 + 4 + 1 = 109$$

$$2^0 = 1$$

Kleines 1x1 der Bits und Bytes

1 Byte	= 8 Bits			
1 KB	= 1 Kilobyte	= 10^3 Bytes	=	1'000 Bytes
1 MB	= 1 Megabyte	= 10^6 Bytes	=	1'000'000 Bytes
1 GB	= 1 Gigabyte	= 10^9 Bytes	=	1'000'000'000 Bytes
1 TB	= 1 Terabyte	= 10^{12} Bytes	=	1'000'000'000'000 Bytes

Referenzen

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Mit Inhalten von Matthias Baumgartner, Katharina Reinecke und Tom Schreiber

Credits

- ?
- Icon Super Mono by Double-J
- Image On-Off-Switch by All Vectors
- Image Hörfläche und Farbmodelle: wikipedia
- Image Soundwave-vector by vectorstock
- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Hidden:
 - Image Smoke signals by sundve (www.flickr.com/photos/sundve/3766277273)
 - Font AlphaSmoke by Beeline



Referent

Prof. Abraham Bernstein
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Sie möchten dieses Video weiterverwenden?
Hier geht es zu den Nutzungsbedingungen...

6.2 Digitalisierung von Text



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 2: Digitalisierung von Information

Lektion 2: Digitalisierung von Text



Prof. Abraham Bernstein

UZH, Institut für Informatik und Digital Society Initiative

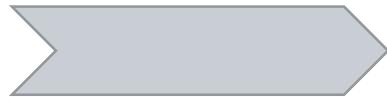


Diese Lektion:

- Einleitung
- **Digitalisierung von Text**
- Digitalisierung von Ton
- Digitalisierung von Bild

Digitalisierung von Text: Nicht alle Verfahren sind geeignet

A

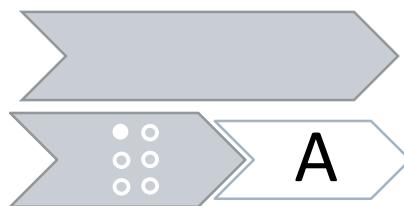


Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

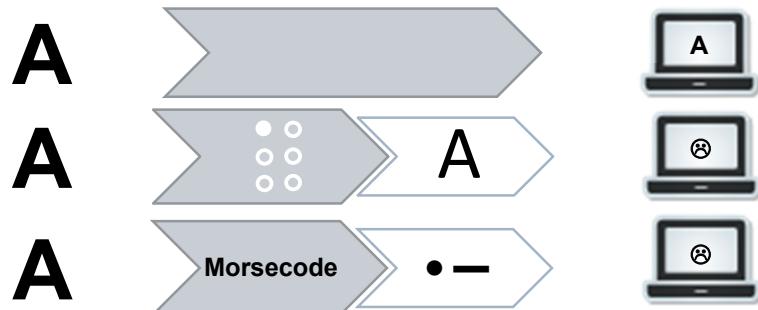
Digitalisierung von Text: Nicht alle Verfahren sind geeignet

A

A



Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Digitalisierung von Text: Nicht alle Verfahren sind geeignet

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Digitalisierung von Text: ASCII-Code

American Standard Code for Information Interchange

ASCII-Codetabelle für Zahlen und Zeichen

0-9: 48-57 A-Z: 65-90 a-z: 97-122

7 Bit: Reicht für 128 Zeichen (2^7)

→ OK für alle Zeichen der englischen Sprache

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Digitalisierung von Text: ASCII-Code



American Standard Code for Information Interchange

ASCII-Codetabelle für Zahlen und Zeichen

0-9: 48-57 A-Z: 65-90 a-z: 97-122

7 Bit: Reicht für 128 Zeichen (2^7)

→ OK für alle Zeichen der englischen Sprache

→ Ungenügend u.a. für europäische Sprachen

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Digitalisierung von Text: ASCII-Code



American Standard Code for Information Interchange

ASCII-Codetabelle für Zahlen und Zeichen

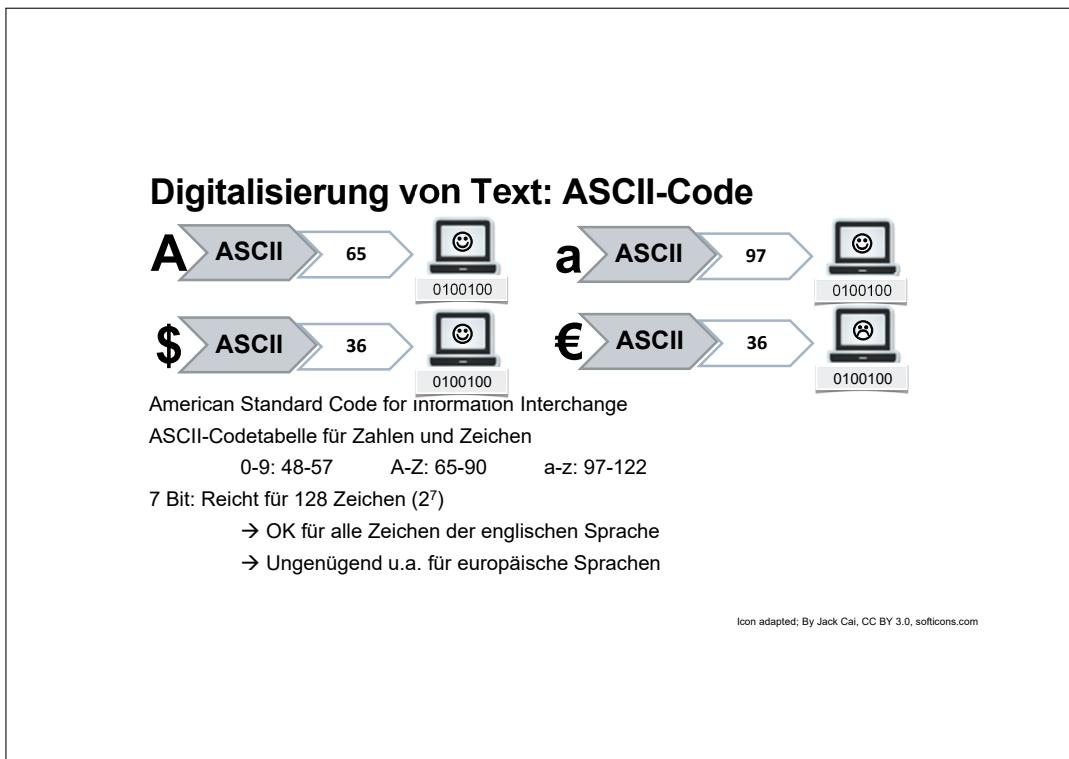
0-9: 48-57 A-Z: 65-90 a-z: 97-122

7 Bit: Reicht für 128 Zeichen (2^7)

→ OK für alle Zeichen der englischen Sprache

→ Ungenügend u.a. für europäische Sprachen

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com



Digitalisierung von Text: ASCII-Code Tabelle

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
0	0	00000000	0		48	30	11000000	60		96	60	11000000	140	
1	1	00000001	1		49	31	11000001	61		97	61	11000001	141	
2	2	00000010	2		50	32	11000010	62		98	62	11000010	142	b
3	3	00000011	3		51	33	11000011	63		99	63	11000011	143	c
4	4	00000100	4		52	34	11000100	64		100	64	11000100	144	d
5	5	00000101	5		53	35	11000101	65		101	65	11000101	145	e
6	6	00000110	6		54	36	11000110	66		102	66	11000110	146	f
7	7	00000111	7		55	37	11000111	67		103	67	11000111	147	g
8	8	00001000	10		56	38	11000000	70		104	68	11000000	150	h
9	9	00001001	11		57	39	11000001	71		105	69	11000001	151	i
10	A	00001010	12		58	3A	11000110	72		106	6A	11000110	152	j
11	B	00001011	13		59	3B	11000111	73		107	6B	11000111	153	k
12	C	00001100	14		60	3C	11000110	74		108	6C	11000110	154	l
13	D	00001101	15		61	3D	11000111	75		109	6D	11000111	155	m
14	E	00001110	16		62	3E	11000110	76		110	6E	11000110	156	n
15	F	00001111	17		63	3F	11000111	77		111	6F	11000111	157	o
16		00000000	20		64	40	10000000	100		112	70	11000000	160	p
17		00000001	21		65	41	10000001	101		113	71	11000001	161	q
18		00000010	22		66	42	10000010	102	B	114	72	11000010	162	r
19		00000011	23		67	43	10000011	103	C	115	73	11000011	163	s
20		00000100	24		68	44	10000100	104	D	116	74	110000100	164	t
21		00000101	25		69	45	10000101	105	E	117	75	110000101	165	u
22		00000110	26		70	46	10000110	106	F	118	76	110000110	166	v
23		00000111	27		71	47	10000111	107	G	119	77	110000111	167	w
24		00001000	28		72	48	10000000	108	H	120	78	11000000	168	x
25		00001001	29		73	49	10000001	109	I	121	79	11000001	169	y
26		00001010	30		74	4A	10000010	110	J	122	7A	11000010	170	z
27		00001011	31		75	4B	10000011	111	K	123	7B	11000011	171	f
28		00001100	32		76	4C	10000100	114	L	124	7C	110000100	174	l
29		00001101	33		77	4D	10000101	115	M	125	7D	110000101	175	j
30		00001110	34		78	4E	10000110	116	N	126	7E	110000110	176	i
31		00001111	35		79	4F	10000111	117	O	127	7F	110000111	177	{DEL}
32		00000000	37		80	50	10000000	118						
33	21	00000001	41		81	51	10000001	121	Q					
34	22	00000010	42		82	52	10000010	122	R					
35	23	00000011	43	#	83	53	10000011	123	S					
36	24	00000100	44	\$	84	54	10000100	124	T					
37	25	00000101	45	%	85	55	10000101	125	U					
38	26	00000110	46	&	86	56	10000110	126	V					
39	27	00000111	47	*	87	57	10000111	127	W					
40	28	00001000	50	_	88	58	10000000	130	X					
41	29	00001001	51		89	59	10000001	131	Y					
42	30	00001010	52		90	5A	10000010	132	Z					
43	28	00001011	53	+	91	5B	10000011	133	!					
44	29	00001100	54	-	92	5C	10000100	134	,					
45	2D	00011010	55	*	93	5D	10011010	135						
46	2E	00011010	56	/	94	5E	10011010	136	~					
47	2F	00011111	57	:	95	5F	10011111	137	-					

Source: Wikimedia Commons <https://commons.wikimedia.org/wiki/File:ASCII-Table.svg>

Digitalisierung von Text: ISO 8859-15



8 Bit: ISO 8859 definiert 256 Zeichen (2⁸)

Die ersten 128 Zeichen stimmen mit ASCII überein

Nicht geeignet für arabische, chinesische, griechische, hebräische, japanische, etc.

Schriften

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Digitalisierung von Text: Unicode



32 Bit: Über 4 Mrd. Zeichen (2³²) können codiert werden

Grundsätzlich können alle Alphabete der Erde codiert werden

Z.Z. werden über 100'000 Zeichen erfasst (Unicode 6.2)

Auf Klingonisch muss weiterhin verzichtet werden: Unicode-Konsortium verweigert die Aufnahme hartnäckig

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Digitalisierung von Text: Unicode

Es existieren verschiedene weitere Codierungssysteme – passend für die jeweiligen Bedürfnisse

Je universeller das System, desto mehr Bits werden für die Codierung benötigt → Speicherbedarf!

Damit die Zeichen korrekt interpretiert werden können, ist die Angabe des verwendeten Codierungssystems zwingend

Icon adapted; By Jack Cai, CC BY 3.0, softicons.com

Je mehr Text, desto höher der Speicherbedarf

1 Seite Text

80 Zeichen/Zeile

64 Zeilen/Seite

8 Bit pro Zeichen (= 1 Byte, ISO 8859-15)

$$= 80 \times 64 \times 8 = 40'960 \text{ Bits}$$

$$\rightarrow 40'960 / 8 = 5'120 \text{ Bytes} \simeq 5,1 \text{ kB}$$

Speicherbedarf: 5,1 kB

Axelsson, Bernhard und Nathalie. *Not Is This Really Science? The Semantic Web's Guide to Evaluating Research Contributions*. 2014 Technical Report.

standing on a platform and the train stops, then we will be left with three figures in the platform. Note that we have now removed words from the top (including the first one) and added three new ones. However, the last is just a pure textual construct and we have decided whether it is correct. In the rest of this paper, we will focus on the first two cases, where the question is whether it is correct and, more important, of extracting sentences and others that are truly such.

Now what about stamps? Stamp-collecting is a great tradition in science. Even though some interpret Birkhoff's statement as something derogatory towards stamp-collecting, it is still a good example of how we can gather observations about "The World" and we usually organize our structures around them. For example, we can collect all the stamps that people communicate about things. Just consider the International Classification of Diseases (ICD) or the International Standard Book Number (ISBN) which organizes books and the Standard Industrial Classification (SIC) Code which organizes companies. These are just examples of how we can collect things in the world, but how do we know that we selected our collection "correctly"? How do we demonstrate that our organization is not only "correct", but also helps others? This is the main idea behind the Semantic Web. We can collect information in projects that, at least on the surface, appear to be stamp collecting. There cannot be any guarantee that the collected information is correct. This applies not only for the goal of collecting them (i.e., we are observing the world not of texts), but also, like there are many potential sources contradicting one another.

With being other "stamps" or "stamp collectors", our overarching goal is obviously to collect as much information as possible. This is why, in general, we want others to be able to use our theories and maybe to build upon them. The idea is that we can collect as much as possible and then share it with others, making it available for reuse and making intercouver compatibility. Many have relied on this theory since Stigler's work on the history of science.

In both approaches, there is a need to establish a notion of correctness of our results. In stamp collecting, we can just compare our collection with other collections, we typically gather evidence supporting our case and often compare our collection with other collections. In the Semantic Web, we need to prove that our collection is correct. This means that we have to prove that we adhered following this procedure were correct. In other words, the results of our collection must be correct. This is a very difficult task and it must be done carefully. These are questions of reliability and validity. Let's do this by looking at the following example.

http://www.w3.org/2001/sw/wiki/contributing/evaluating_an_order.html
https://www.w3.org/2001/sw/wiki/contributing/evaluating_an_order.html
http://www.w3.org/2001/sw/wiki/contributing/evaluating_an_order.html

Referenzen

Icons: Jack Cai, Double-J Design, CC-BY 3.0

Bilder:

- <https://commons.wikimedia.org/wiki/File:ASCII-Table.svg>
- <https://commons.wikimedia.org/wiki/File:Hoerflaeche.svg>
- Mehrere basierend auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>
- <https://commons.wikimedia.org/wiki/File:AdditiveColorMixingII.png>

Inhalte

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I

Das sind die von
mir nun erfassten

Credits

- ?
- Icon Super Mono by Double-J
- Image On-Off-Switch by All Vectors
- Image Hörfläche und Farbmodelle: wikipedia
- Image Soundwave-vector by vectorstock
- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Hidden:
 - Image Smoke signals by sundve (www.flickr.com/photos/sundve/3766277273)
 - Font AlphaSmoke by Beeline

Weiss nicht, welche
davon noch aktuell sind

Referenzen

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Mit Inhalten von Matthias Baumgartner, Katharina Reinecke und Tom Schreiber

Referentin

Prof. Abraham Bernstein
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Sie möchten dieses Video weiterverwenden?
Hier geht es zu den Nutzungsbedingungen...

6.3 Digitalisierung von Ton



Universität
Zürich UZH

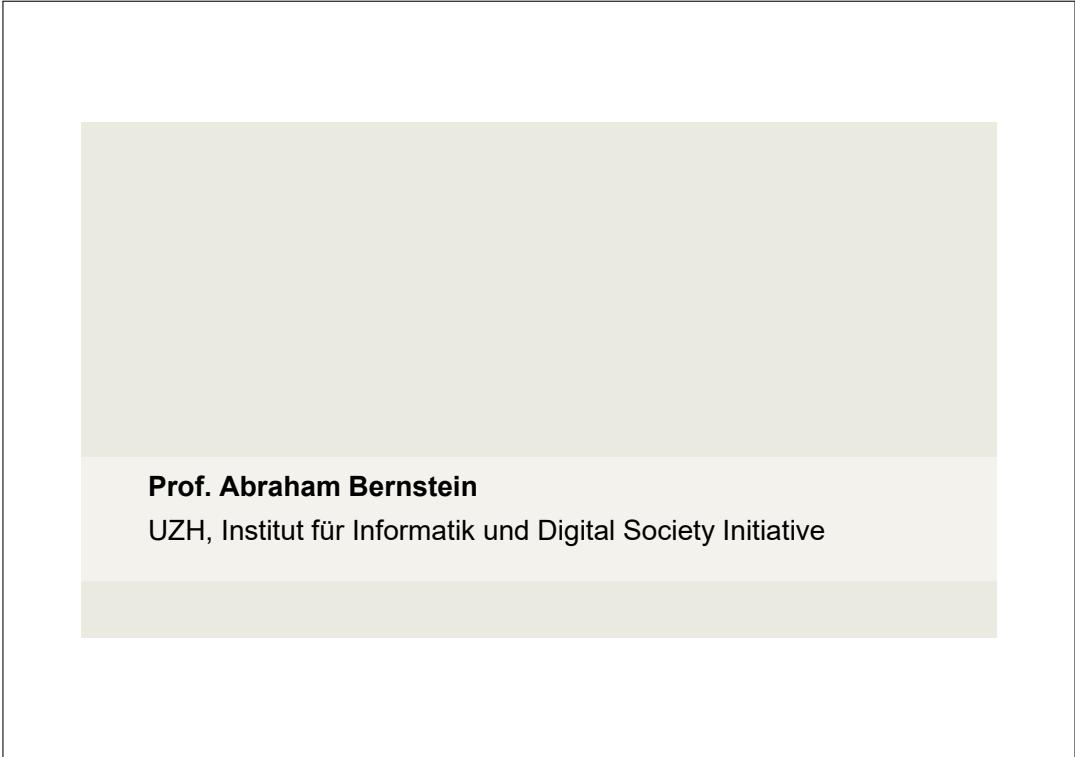
Digital Society Initiative



Studium Digitale

Kursbaustein 2: Digitalisierung von Information

Lektion 3: Digitalisierung von Ton



Prof. Abraham Bernstein

UZH, Institut für Informatik und Digital Society Initiative



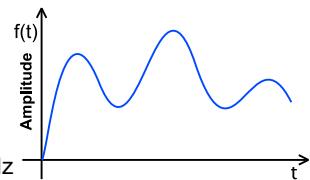
Diese Lektion:

- Einleitung
- Digitalisierung von Text
- **Digitalisierung von
Ton**
- Digitalisierung von Bild

Exkurs: Schalleigenschaften

Frequenz (Tonhöhe)

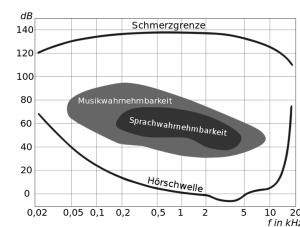
- 1 Hz = eine Schwingung pro Sekunde
- Je höher die Frequenz, desto höher der Ton
- Der Mensch hört zwischen 20 Hz und 22'000 Hz



<https://commons.wikimedia.org/w/index.php?title=Hörfläche.svg&oldid=16208011>
basiert auf <https://commons.wikimedia.org/w/index.php?title=Digital.signal.svg&oldid=16208011>

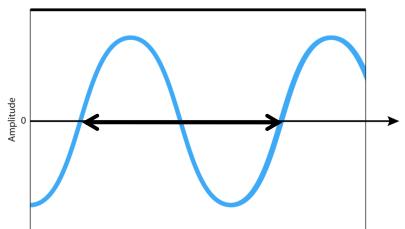
Lautstärke

- 60 dB: Unterhaltung
- 90 dB: Laute Musik
- 120 dB: Flugzeug



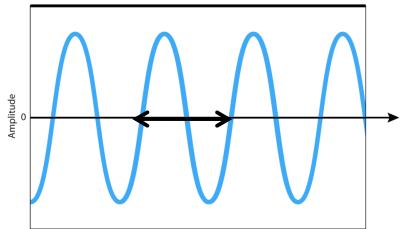
Exkurs: Schalleigenschaften

- Frequenz (Tonhöhe)
 - 1 Hz = eine Schwingung pro Sekunde

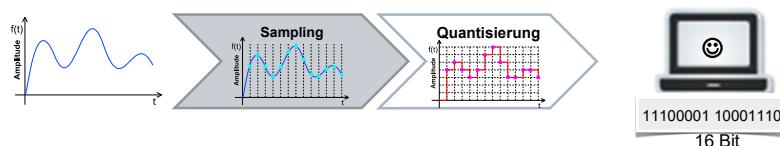


Exkurs: Schalleigenschaften

- Frequenz (Tonhöhe)
 - 1 Hz = eine Schwingung pro Sekunde
 - Je höher die Frequenz, desto höher der Ton



Tondigitalisierung: Sampling und Quantisierung



Abtasten (Sampling)

→ Periodisches Abtasten des analogen Signals

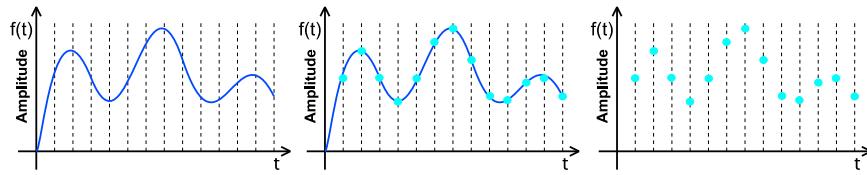
Quantisierung

→ Zuordnen der ermittelten Werte an vorgegebene Werte

Basierend auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>
Icon adapted by Jack Cai, CC BY 3.0, sonicons.com

Sampling

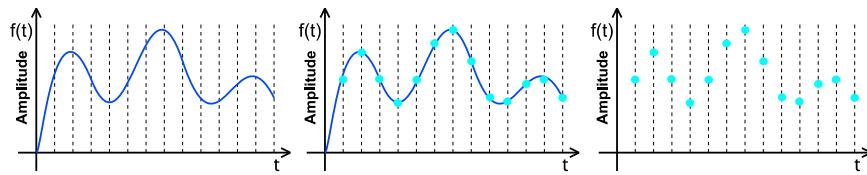
- Analoges Signal wird in regelmässigen Zeitabständen gemessen
- Gemessene Werte ergeben Abbild des Ursprungssignals



basiert auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>

Sampling

Je höher die Zahl der Samples pro Sekunde (= Abtastrate), desto besser entspricht das digitale Abbild dem Original

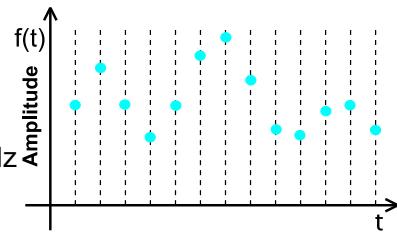


basiert auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>

Sampling

Für gute Ergebnisse muss die Abtastrate *mindestens das Doppelte* der erwünschten maximalen Frequenz betragen

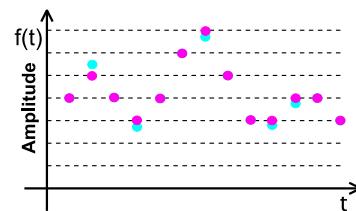
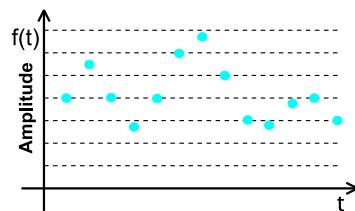
Da der Mensch maximal 22'000 Hz hört, sollte die Abtastrate mindestens 44'000 Hz betragen
(CD-Qualität: 44'100 Hz)



basiert auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>

Quantisierung

Erfasste Messwerte werden gerundet und einem vordefinierten Wert zugewiesen

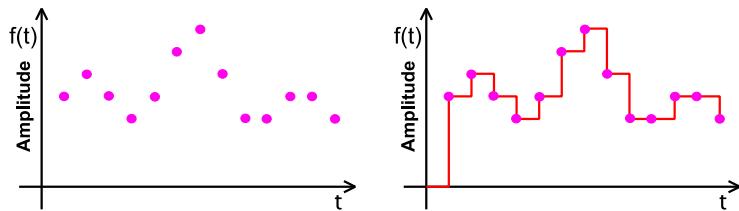


basiert auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>

Quantisierung

Durch Hinzufügen eines Bits wird das Ergebnis doppelt so genau

- 8 Bits: 256 unterscheidbare Tonwerte
- 16 Bits: 65'536 unterscheidbare Tonwerte



basiert auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>

**Je besser die Tonqualität,
desto höher der
Speicherbedarf**

1 Minute Musik (CD-Qualität)

- 60 Sekunden
- 44'100 Samples pro Sekunde
- 16 Bit pro Sample (= 2 Bytes)
- Stereo = 2 Kanäle
- = $60 \times 44'100 \times 2 \times 2$
- = 10'584'000 Bytes
- ➔ Speicherbedarf: 10,6 MB

Referenzen

Icons: Jack Cai, Double-J Design, CC-BY 3.0

Bilder:

- <https://commons.wikimedia.org/wiki/File:ASCII-Table.svg>
- <https://commons.wikimedia.org/wiki/File:Hoerflaeche.svg>
- Mehrere basierend auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>
- <https://commons.wikimedia.org/wiki/File:AdditiveColorMixingII.png>

Inhalte

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I

Das sind die von
mir nun erfassten

Credits

- ?
- Icon Super Mono by Double-J
- Image On-Off-Switch by All Vectors
- Image Hörfläche und Farbmodelle: wikipedia
- Image Soundwave-vector by vectorstock
- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Hidden:
 - Image Smoke signals by sundve (www.flickr.com/photos/sundve/3766277273)
 - Font AlphaSmoke by Beeline

Weiss nicht, welche
davon noch aktuell sind

Referenzen

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Mit Inhalten von Matthias Baumgartner, Katharina Reinecke und Tom Schreiber

Referentin

Prof. Abraham Bernstein
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Sie möchten dieses Video weiterverwenden?
Hier geht es zu den Nutzungsbedingungen...

6.4 Digitalisierung von Bild



Universität
Zürich UZH

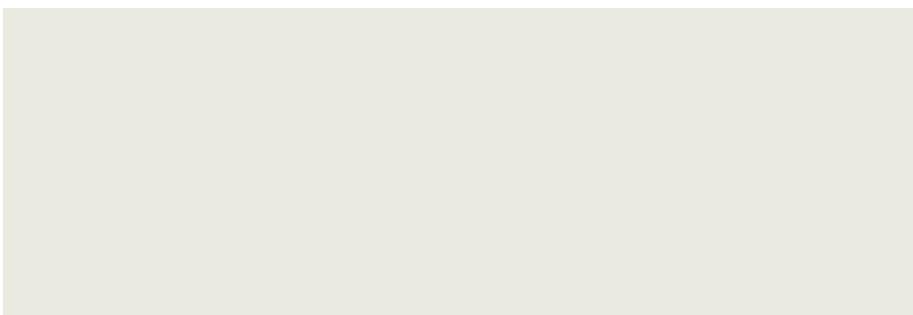
Digital Society Initiative



Studium Digitale

Kursbaustein 2: Digitalisierung von Information

Lektion 4: Digitalisierung von Bild



Prof. Abraham Bernstein

UZH, Institut für Informatik und Digital Society Initiative



Diese Lektion:

- Einleitung
- Digitalisierung von Text
- Digitalisierung von Ton
- **Digitalisierung von Bild**



The diagram shows a large light grey area on the left and a smaller dark grey vertical bar on the right containing text and a color model diagram.

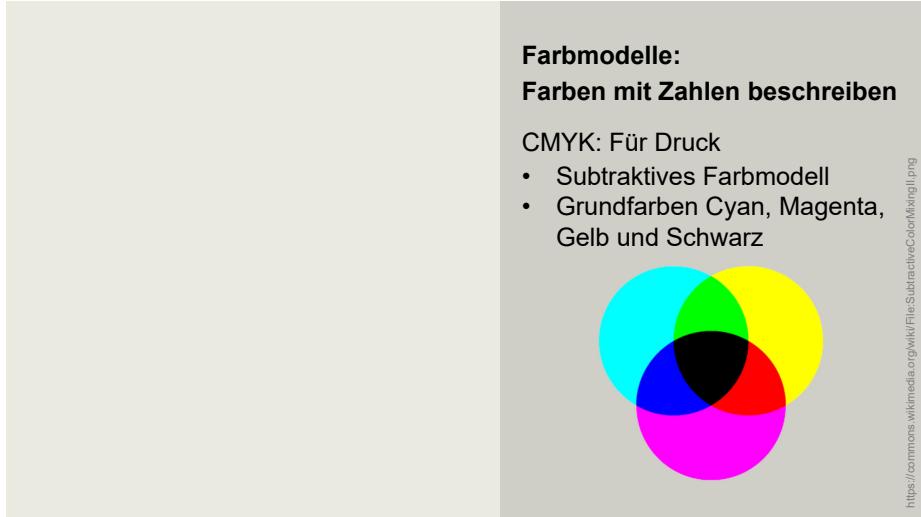
Farbmodelle:
Farben mit Zahlen beschreiben

RGB: Für Bildschirme

- Additives Farbmodell
- Drei Grundfarben:
Rot, Grün, Blau

[https://commons.wikimedia.org/wiki/File/AdditiveColorMixing.png](https://commons.wikimedia.org/wiki/File>AdditiveColorMixing.png)

A circular diagram illustrating additive color mixing. Three overlapping circles represent primary colors: red, green, and blue. The overlapping areas show secondary colors: cyan (green and blue), magenta (red and blue), and yellow (red and green). The central area where all three circles overlap is white.



The diagram shows a large light gray rectangle on the left and a smaller dark gray rectangle on the right. Inside the dark gray rectangle, there is a circular diagram of four overlapping colored circles: cyan, magenta, yellow, and black. The overlapping areas create various colors, demonstrating the subtractive color mixing process used in printing.

Farbmodelle:
Farben mit Zahlen beschreiben

CMYK: Für Druck

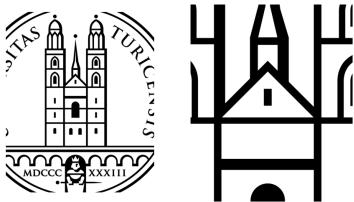
- Subtraktives Farbmodell
- Grundfarben Cyan, Magenta, Gelb und Schwarz

<https://commons.wikimedia.org/wiki/File/SubtractiveColorMixing.png>



Vektorgrafik

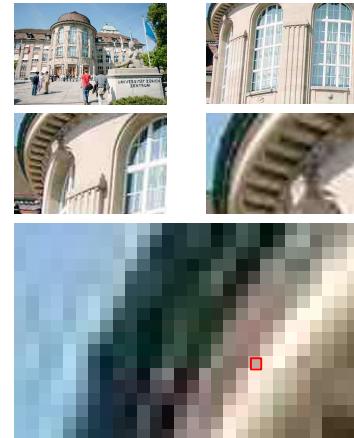
- Besteht aus mathematisch definierten Objekten
- Jedes Objekt hat Attribute. Es werden nur Attributwerte gespeichert
- Jedes Objekt kann einzeln bearbeitet werden
- Vergrössern ohne Qualitätseinbusse



The diagram illustrates the difference between vector and raster graphics. On the left is the official seal of the University of Zurich, which is a complex vector graphic. On the right is a simplified, blocky representation of the same seal, which is a raster graphic. This comparison demonstrates that vector graphics can be scaled infinitely without losing quality, unlike raster graphics which become pixelated when enlarged.

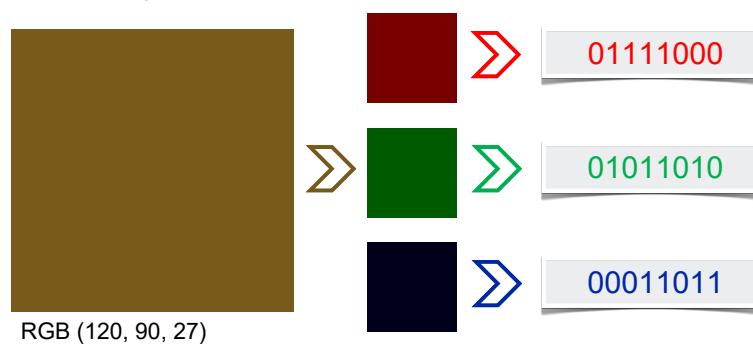
Pixelgrafik

- Besteht aus rasterförmig angeordneten Punkten (Pixel)
- Jeder Punkt hat Farbwert gemäss Farbmodell
- Jeder Punkt kann einzeln bearbeitet werden



Pixelgrafik: Farben

Beispiel RGB, 24 Bit Farbtiefe: Für jedes Pixel 8 Bit pro Grundfarbe → 256 Farben pro Kanal → 16.7 Mio. Farben



Pixelgrafik: Speicherbedarf

Beispiel RGB, 24 Bit Farbtiefe: Für jedes Pixel 8 Bit pro Grundfarbe →
256 Farben pro Kanal → 16.7 Mio. Farben



RGB (120, 90, 27)



Pixelgrafik: Speicherbedarf

Postkarte



1748×1240 Pixel = 2'167'520 Pixel

×

011110000101101000011011

=

24 Bit pro Pixel

52'020'480 Bit = 6'502'560 Bytes

Speicherbedarf: 6,5 MB

Icon adapted: By Jack Cai, CC BY 3.0, softicons.com

Fazit

- Information kann in verschiedenen Formen vorkommen (z.B. Text, Bild, Ton)
- Für die informatikgestützte Verarbeitung müssen die Informationen aufbereitet und in Binärdaten umgewandelt werden
- Detailtreue Erfassung vergrössert den Speicherbedarf

Referenzen

Icons: Jack Cai, Double-J Design, CC-BY 3.0

Bilder:

- <https://commons.wikimedia.org/wiki/File:ASCII-Table.svg>
- <https://commons.wikimedia.org/wiki/File:Hoerflaeche.svg>
- Mehrere basierend auf <https://commons.wikimedia.org/wiki/File:Digital.signal.svg>
- <https://commons.wikimedia.org/wiki/File:AdditiveColorMixingII.png>

Inhalte

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I



Das sind die von mir nun erfassten

Credits

- ?
- Icon Super Mono by Double-J
- Image On-Off-Switch by All Vectors
- Image Hörfläche und Farbmodelle: wikipedia
- Image Soundwave-vector by vectorstock
- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Hidden:
 - Image Smoke signals by sundve (www.flickr.com/photos/sundve/3766277273)
 - Font AlphaSmoke by Beeline



Weiss nicht, welche davon noch aktuell sind

Referenzen

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Mit Inhalten von Matthias Baumgartner, Katharina Reinecke und Tom Schreiber

Referentin

Prof. Abraham Bernstein
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Sie möchten dieses Video weiterverwenden?
Hier geht es zu den Nutzungsbedingungen...

6.5 **Datenspeicherung**



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 2: Digitalisierung von Information

Lektion 5: Datenspeicherung

Prof. Abraham Bernstein

UZH, Institut für Informatik und Digital Society Initiative

Diese Lektion:

Datenspeicherung

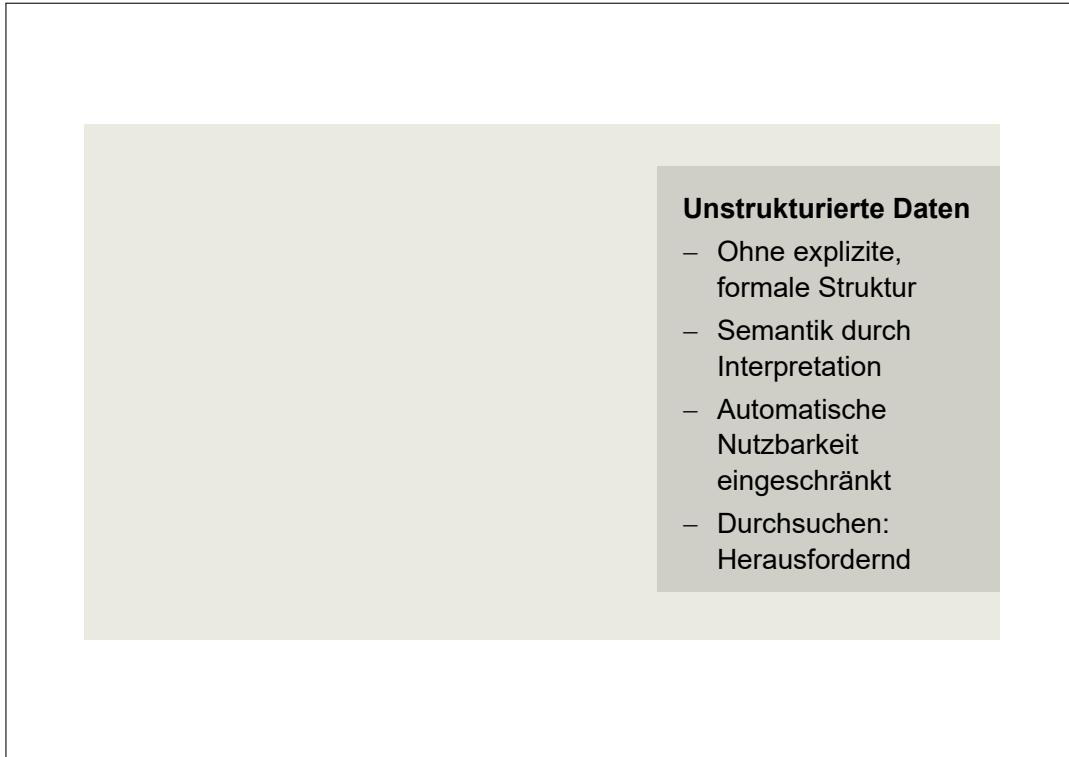
- Unstrukturierte Daten
- Strukturierte Daten

Diese Lektion:

- Datenspeicherung
- **Unstrukturierte Daten**
 - Strukturierte Daten

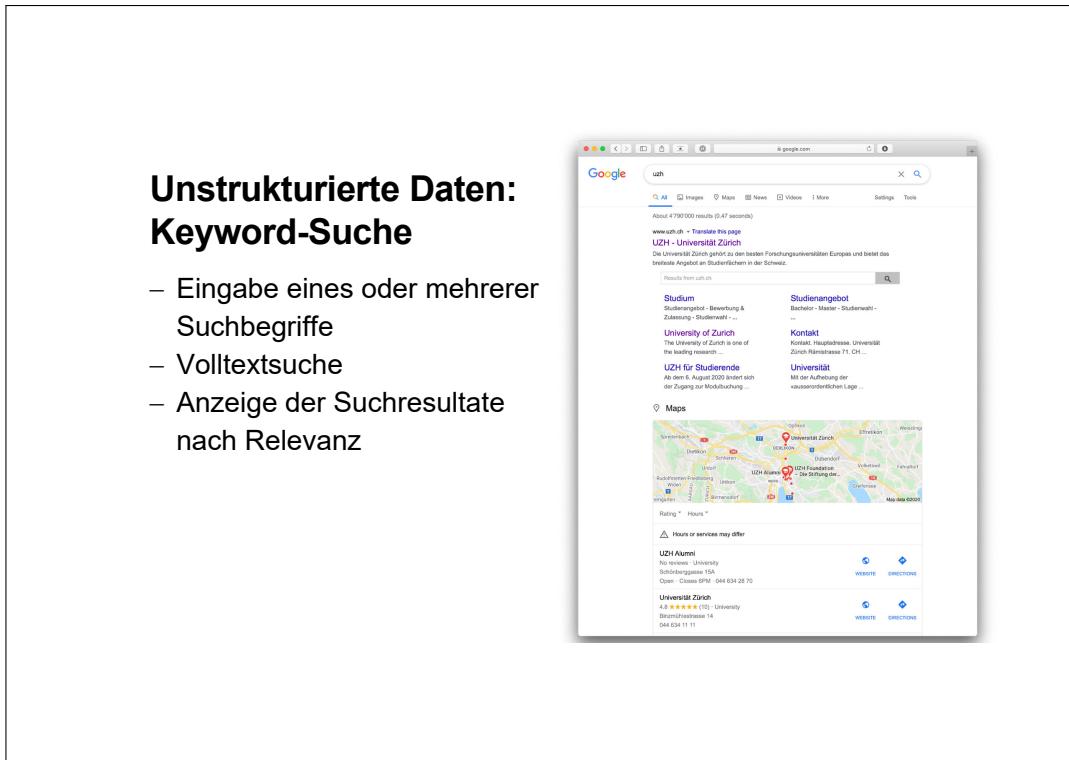
Getting information off the Internet is like taking a drink from a fire hydrant.

Mitch Kapor



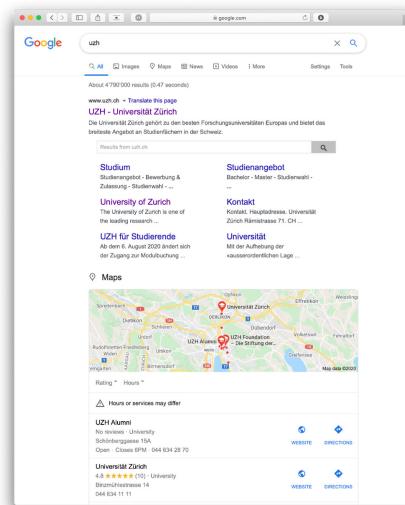
Unstrukturierte Daten

- Ohne explizite, formale Struktur
- Semantik durch Interpretation
- Automatische Nutzbarkeit eingeschränkt
- Durchsuchen: Herausfordernd



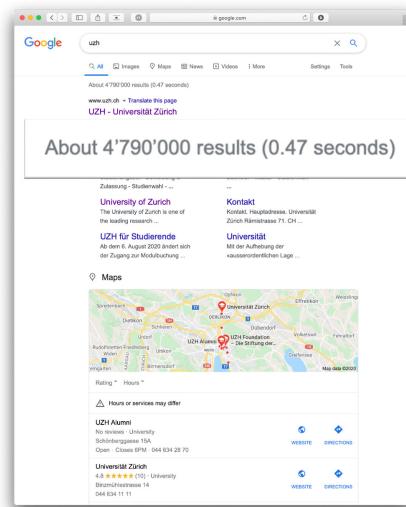
Unstrukturierte Daten: Keyword-Suche

- Eingabe eines oder mehrerer Suchbegriffe
- Volltextsuche
- Anzeige der Suchresultate nach Relevanz

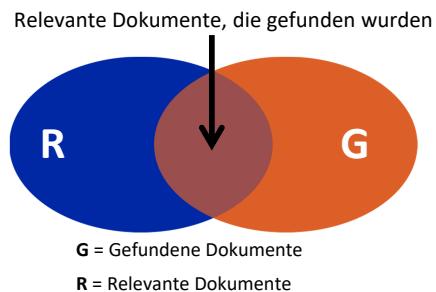


Unstrukturierte Daten: Keyword-Suche

- Eingabe eines oder mehrerer Suchbegriffe
- Volltextsuche
- Anzeige der Suchresultate nach Relevanz



Probleme der Keyword-Suche

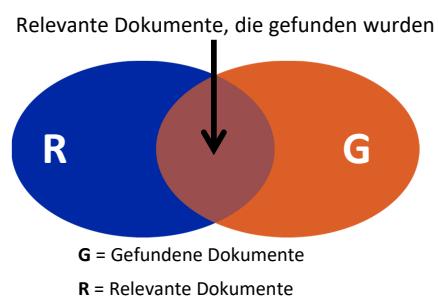


Probleme der Keyword-Suche

- Keyword-Suche findet viele irrelevante Dokumente

$$\text{Precision} = \frac{|R \cap G|}{|G|}$$

P = 100%
Alle gefundenen Dokumente sind relevant



Probleme der Keyword-Suche

- Keyword-Suche findet viele irrelevante Dokumente

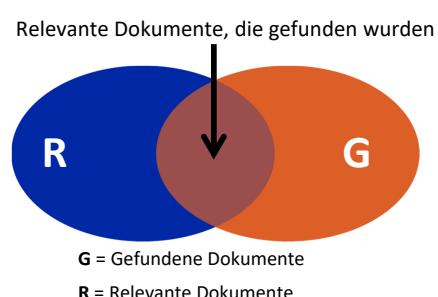
$$\text{Precision} = \frac{|R \cap G|}{|G|}$$

P = 100%
Alle gefundenen Dokumente sind relevant

- Keyword-Suche findet nicht alle relevanten Dokumente

$$\text{Recall} = \frac{|R \cap G|}{|R|}$$

R = 100%
Alle relevanten Dokumente wurden gefunden



Wie kann man diese Problematik angehen?

- Semantik
- Personalisierung
- Page Rank

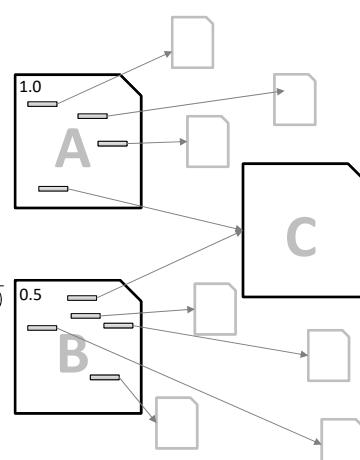
Page Rank

Versucht anhand der Linkstruktur die Wichtigkeit von Dokumenten zu erschliessen

$$\text{PageRank}_A = (1 - d)$$

$$+ d * \sum_{n \in N} \frac{\text{PageRank}_n}{\text{Anzahl(AusgehendeLinks}_n)}$$

d = Dämpfungsfaktor (z.B. 0.85)
 N = Dokumente mit Link nach A



Page Rank von C berechnen

$$d = 0.85$$

$$N = \{A, B\}$$

A: PageRank = 1.0, 4 ausgehende Links

B: PageRank = 0.5, 5 ausgehende Links

$$\text{PageRank}_C = (1 - d)$$

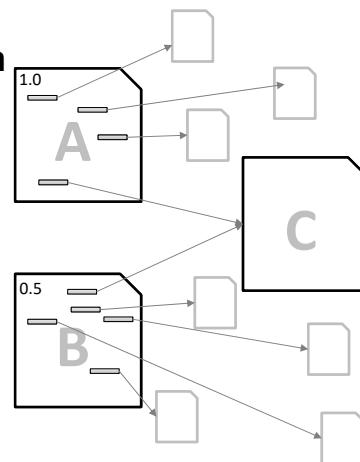
$$+ d * \sum_{n \in N} \frac{\text{PageRank}_n}{\text{Anzahl(AusgehendeLinks}_n)}$$

$$= (1 - 0.85)$$

$$+ 0.85 * (\frac{1.0}{4} + \frac{0.5}{5})$$

$$= 0.15 + 0.85 * (0.25 + 0.1)$$

$$= 0.4475$$



Diese Lektion:

Datenspeicherung

- Unstrukturierte Daten
- **Strukturierte Daten**

Strukturierte Daten

- Haben explizite, formaler Struktur
- Semantik vorgegeben
- Automatische Nutzbarkeit möglich
- Durchsuchen: Problemlos

Strukturierte Daten

Daten werden strukturiert in einer Datenbank gespeichert

- Beziehungen durch mathematische Relationen modelliert

Verwaltung, z.B., mittels Datenbankmanagementsystemen

Suche mit logikbasierter Sprache

- Relevanz gemäss Suchkriterien
- Zusätzliche Ergebnisse dank Nutzung von Relationen

Von Unstrukturierten zu Strukturierte Daten

Petra Graf, Tulpenweg 14 in Zürich, hat am 24. August die Bücher Prozessstudien zur Golfstromablösung in windgetriebenen numerischen Modellen von Joachim Dengg sowie die Geschichte des Golfstroms und seiner Erforschung von Johann Georg Kohl ausgeliehen. Andreas Zenk aus Bern (Brunnenstrasse 27) hat sich am 28. August das Buch *Golf: Zehn Lektionen in der Kunst des Scheiterns* des Autors Gregor Eisenhauer sowie die DVD *Golf für Dummies* ausgeborgt. Matthias Schwab, der am Idaplatz 7 in Bremgarten wohnt, sollte das *Golf Schrauberhandbuch - Reparieren und optimieren leicht gemacht* (von Lindsay Porter) schon vor einer Woche zurückbringen (23 Juni ausgeliehen).

Von Unstrukturierten zu Strukturierte Daten

Petra Graf wohnt am Tulpenweg 14 in Zürich. Sie hat das Buch „Prozessstudien zur Golfstromablösung in windgetriebenen numerischen Modellen“ des Autors Joachim Dengg am 24. August ausgeliehen. Sie hat das Buch „Die Geschichte des Golfstroms und seiner Erforschung“ des Autors Johann Georg Kohl am 24. August ausgeliehen.

Andreas Zenk wohnt an der Brunnenstrasse 27 in Bern. Er hat das Buch „Golf: Zehn Lektionen in der Kunst des Scheiterns“ des Autors Gregor Eisenhauer am 28. August ausgeliehen. Er hat die DVD „Golf für Dummies“, am 28. August ausgeliehen.

Matthias Schwab wohnt am Idaplatz 7 in Bremgarten. Er hat das Buch „Das Golf Schrauberhandbuch - Reparieren und optimieren leicht gemacht“ der Autorin Lindsay Porter am 23 Juni ausgeliehen.

Von Unstrukturierten zu Strukturierte Daten

Petra Graf wohnt am Tulpenweg 14 in Zürich. Sie hat das Buch „Prozessstudien zur Golfstromablösung in windgetriebenen numerischen Modellen“ des Autors Joachim Dengg am 24. August ausgeliehen. Sie hat das Buch „Die Geschichte des Golfstroms und seiner

des Autors Joh **Person** **Strasse** **Ort** **Medienart** **Titel**
Andreas Zenk wohnt an der Brunnenstrasse 27 in Bern. Er hat das Buch „Golf: Zehn Lektionen in der Kunst des Scheiterns“ des Autors Gregor Eisenhauer am 28. August ausgeliehen. Er hat die DVD „Golf für Dumm **Autor** **Datum**

Matthias Schwab wohnt am Idaplatz 7 in Bremgarten. Er hat das Buch „Das Golf Schrauberhandbuch - Reparieren und optimieren leicht gemacht“ der Autorin Lindsay Porter am 23 Juni ausgeliehen.

Von Unstrukturierten zu Strukturierte Daten

Person Andreas Zenk wohnt an der Strasse Brunnenstrasse 27 in Ort Bern. Er hat das Medienart Buch Titel Golf: Zehn Lektionen in der Kunst des Scheiterns des Autors Autor Gregor Eisenhauer am Datum 28. August ausgeliehen. Er hat die Medienart DVD Titel „Golf für Dummies,“ am Datum 28. August ausgeliehen.

Von Unstrukturierten zu Strukturierte Daten

Person_Start

Name_Start Andreas Zenk Name_Ende wohnt an der Strasse_Start Brunnenstrasse 27

Strasse_Ende in Ort_Start Bern Ort_Ende.

Er hat das Medienart_Start Buch Medienart_Ende Titel_Start Golf: Zehn Lektionen in der Kunst des Scheiterns Titel_Ende des Autors Autor_Start Gregor Eisenhauer Autor_Ende am Datum_Start 28. August Datum_Ende ausgeliehen.

Er hat die Medienart_Start DVD Medienart_Ende Titel_Start „Golf für Dummies, Titel_Ende am Datum_Start 28. August Datum_Ende ausgeliehen.

Person_Ende

Von Unstrukturierten zu Strukturierte Daten

```
<Person>
  <Name>Andreas Zenk </Name>
  <Strasse>Brunnenstrasse 27 </Strasse>
  <Ort>Bern </Ort>
  <Ausleihe>
    <Datum>28. August </Datum>
    <Medienart>Buch </Medienart>
    <Titel>Golf: Zehn Lektionen in der Kunst des Scheiterns </Titel>
    <Autor>Gregor Eisenhauer </Autor>
  </Ausleihe>
  <Ausleihe>
    <Datum>28. August </Datum>
    <Medienart>DVD </Medienart>
    <Titel>Golf für Dummies </Titel>
  </Ausleihe>
</Person>
```

Von Unstrukturierten zu Strukturierte Daten

```
<Person>
  <Name>Andreas Zenk </Name>
  <Strasse>Brunnenstrasse 27 </Strasse>
  <Ort> Bern </Ort>
  <Ausleihe>
    <Datum> 28. August </Datum>
    <Medienart> Buch </Medienart>
    <Titel> Golf: Zehn Lektionen in der Kunst des Scheiterns </Titel>
    <Autor> Gregor Eisenhauer </Autor>
  </Ausleihe>
  <Ausleihe>
    <Datum> 28. August </Datum>
    <Medienart> DVD </Medienart>
    <Titel> Golf für Dummies </Titel>
  </Ausleihe>
</Person>
```

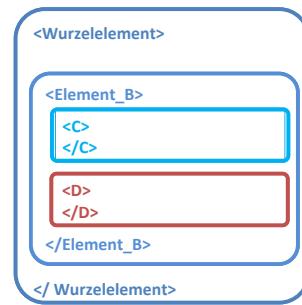
- XML (Extensible Markup Language) ist ein (menschenlesbares) Datenformat
- Häufig zum Datenaustausch verwendet
- Trägt einen Teil der Strukturinformation mit sich

Struktur eines XML Dokuments

- Ein Wurzelement
- Wurzelement kann weitere Elemente beinhalten
- Richtig verschachtelt:
Jedes Kindelement muss geschlossen sein, bevor das Elternelement geschlossen wird

Struktur eines XML Dokuments

- Ein Wurzelement
- Wurzelement kann weitere Elemente beinhalten
- Richtig verschachtelt:
Jedes Kinderelement muss geschlossen sein, bevor das Elternelement geschlossen wird



Fazit

- Viele unstrukturierte Daten → Probleme
- Struktur vereinfach Verarbeitung
- XML ist eine Strukturierungs-möglichkeit

Fazit

Viele Daten liegen unstrukturiert vor

Eine Volltextsuche liefert viele irrelevante Ergebnisse

Verfahren wie PageRank, Personalisierung etc. versuchen, die Relevanz der Suchresultate zu erhöhen

Strukturierte Daten werden basierend auf einem Datenmodell in Datenbanken gespeichert

Ein Datenbankmanagementsystem ermöglicht die effiziente Datenbankbewirtschaftung

SQL-Abfragen (Datenbankabfragen) durchsuchen auch riesige Datenbanken schnell und liefern relevante Ergebnisse

XML ermöglicht es, Daten mit Strukturinformationen zu erfassen

Referenzen

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Mit Inhalten von Matthias Baumgartner, Katharina Reinecke und Tom Schreiber

Referent

Prof. Abraham Bernstein
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Sie möchten dieses Video weiterverwenden?
Hier geht es zu den Nutzungsbedingungen...

6.6 Datenformate



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 2: Digitalisierung von Information

Lektion 6: Datenformate

Prof. Abraham Bernstein
UZH, Institut für Informatik und Digital Society Initiative

Diese Lektion:

- Datenformate
- Digital Rights Management
- Datenkompression

Diese Lektion:

- **Datenformate**
- Digital Rights Management (DRM)
- Datenkompression

Dateien haben
verschiedene Formate:

DOC doc, html, odt, rtf, txt,
xml, ...

MUSIC aac, dts, mp3, ogg, wav,
wma, ...

PIC Pixel: bmp, gif, jpg, png,
raw, tiff, ...

VECTOR: eps, svg, ...

VIDEO: avi, H.264, mov,
mp4, ...

By Jack Cai, CC BY 3.0, softicons.com

Formate:

- Interpretation der Inhalte
- Werden über oft Endungen identifiziert
- Werden durch Hersteller oder Gremien definiert.

Formate unterscheiden sich bezüglich:

- Struktur
(Syntax und Semantik)
- Möglichkeit für Digital Rights Management
- Datenkompression

Diese Lektion:

- Datenformate
- **Digital Rights Management (DRM)**
- Datenkompression



The content of this slide could not be displayed because the associated license is not valid.

Please contact your assistant to purchase a valid license key.

Digital Rights Management (DRM)



- Definition
 - Digitales Management von Rechten, um die Verbreitung und Nutzung digitaler Güter durch technische Massnahmen zu kontrollieren
- Ziel
- Durchsetzen von Urheberrechten
- Übertragen bestehender Geschäftsmodelle auf digitale Welt

Digital Rights Management (DRM)

- Definition
 - Digitales Management von Rechten, um die Verbreitung und Nutzung digitaler Güter durch technische Massnahmen zu kontrollieren
- Ziel
- Durchsetzen von Urheberrechten
- Übertragen bestehender Geschäftsmodelle auf digitale Welt

Vier Rechte stehen typischerweise im Fokus



- **Transportrechte**
Kopieren, Übertragen, Verleihen
- **Vorführrechte**
Anzeigen, Ausdrucken, Abspielen
- **Sicherungsrechte**
Schutz vor Beschädigung oder Verlust
- **Editierrechte**
Verändern, Extrahieren

Nicht mehr das digitale Gut, sondern die Nutzung kostet



- Rechteinhaber vergibt Nutzungslicenz
- Definiert, *wer, was, wann, wie, und wie oft innerhalb welcher Zeitperiode zu welchem Preis* mit den Daten machen darf:
 - Access Control
 - Usage Control
 - Content Tracking
 - Payment Management
- Lizenznehmer hat das Recht, das Gut gemäss den vereinbarten Konditionen zu nutzen

Kritik

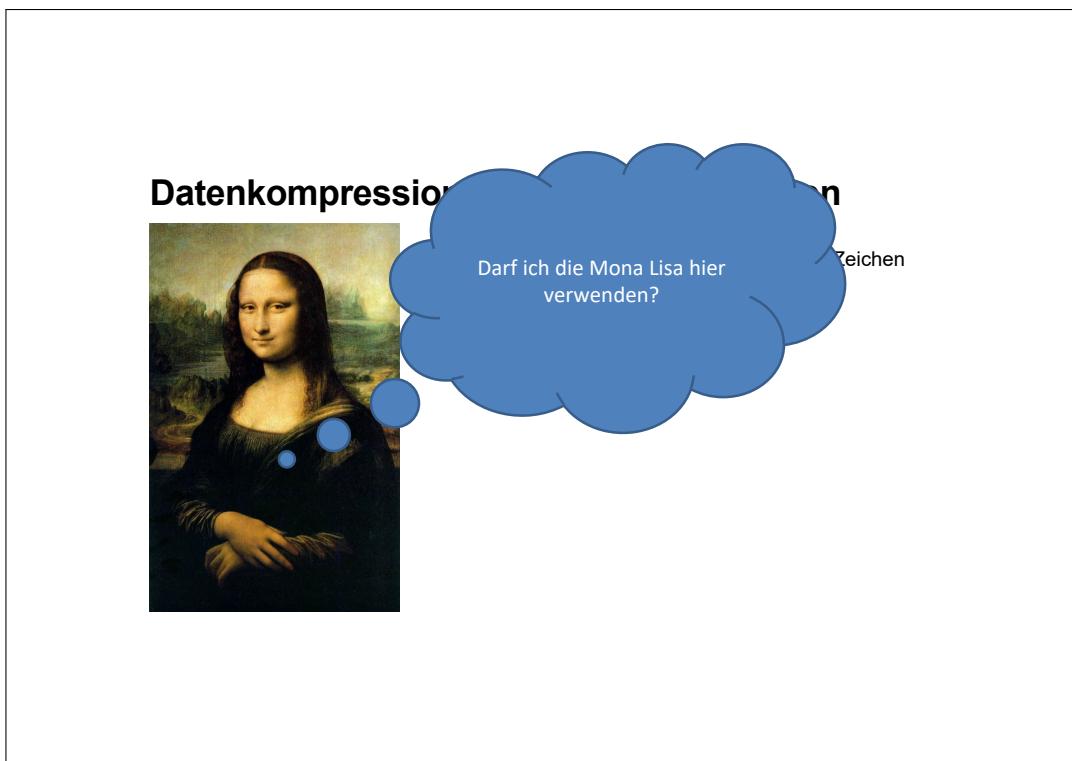
drm is
drm is **good**
drm is **bad**
drm is **evil**
drm is **stupid**



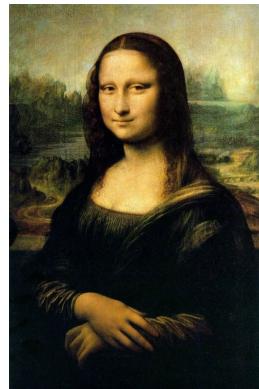
- Kein einheitlicher Standard
→ Kompatibilitätsprobleme
- Systeme nicht sicher
→ DRM kann umgangen werden
- Wettbewerb wird geschwächt
→ Marktzutrittsschranken
- Datenschutz
- Kosten vs. Nutzen

Diese Lektion:

- Datenformate
- Digital Rights Management (DRM)
- **Datenkompression**



Datenkompression: verlustfreie Verfahren



- Redundante Daten werden durch kürzere Zeichen ersetzt.

– Musterwiederholung (z.B. Lauflängenkodierung)

AAAAAAABBBBAAAAAA

6A4B5A



Datenkompression: verlustfreie Verfahren



- Redundante Daten werden durch kürzere Zeichen ersetzt.

– Musterwiederholung (z.B. Lauflängenkodierung)

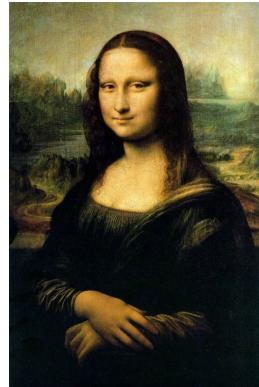
AAAAAAABBBBAAAAAA

6A4B5A



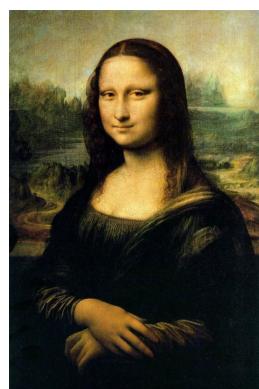
- Reversibel: Daten lassen sich wieder Bit-genau herstellen

Datenkompression: verlustbehaftete Verfahren



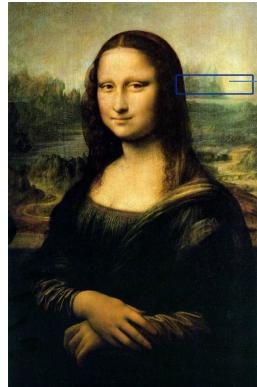
- Nutzen die *Begrenztheit menschlicher Sinne*
 - Helligkeit wird stärker registriert als Farbe oder Farbsättigung
 - Bestimmte Ton-Frequenzen werden nicht wahrgenommen

Datenkompression: verlustbehaftete Verfahren



- Nutzen die *Begrenztheit menschlicher Sinne*
 - Helligkeit wird stärker registriert als Farbe oder Farbsättigung
 - Bestimmte Ton-Frequenzen werden nicht wahrgenommen
- Kompression entfernt Teile des Originals

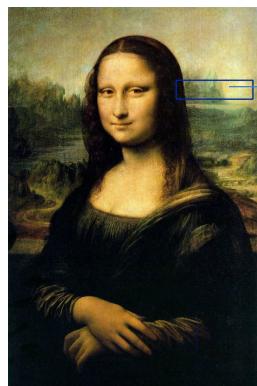
Datenkompression: verlustbehaftete Verfahren



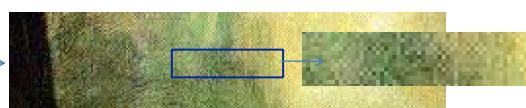
- Nutzen die *Begrenztheit menschlicher Sinne*
 - Helligkeit wird stärker registriert als Farbe oder Farbsättigung
 - Bestimmte Ton-Frequenzen werden nicht wahrgenommen
- Kompression *entfernt* Teile des Originals
- *Nicht reversibel*: Rekonstruktion entspricht nur näherungsweise den Originaldaten



Datenkompression: verlustbehaftete Verfahren



- Nutzen die *Begrenztheit menschlicher Sinne*
 - Helligkeit wird stärker registriert als Farbe oder Farbsättigung
 - Bestimmte Ton-Frequenzen werden nicht wahrgenommen
- Kompression *entfernt* Teile des Originals
- *Nicht reversibel*: Rekonstruktion entspricht nur näherungsweise den Originaldaten



Fazit

- Zahlreiche Formate
- Einige ermöglichen eine Kontrolle der Nutzung
- Dateigröße kann durch Kompression verringert werden

Referenzen

- Hansen, H.R., Neumann, G. (2001): Wirtschaftsinformatik I
- Mit Inhalten von Matthias Baumgartner, Katharina Reinecke und Tom Schreiber

Referent

Prof. Abraham Bernstein
Institut für Informatik

© Universität Zürich
Digital Society Initiative



Sie möchten dieses Video weiterverwenden?
Hier geht es zu den Nutzungsbedingungen...

7 Arten von IS

PROF. ABRAHAM BERNSTEIN

7.1 Einsatzgebiete



Institut für Informatik

Arten von Informationssystemen I – Einsatzgebiete

Prof. Abraham Bernstein, Ph.D.

Arten von Informationssystemen I – Einsatzgebiete

- Das Einsatzgebiet von IS

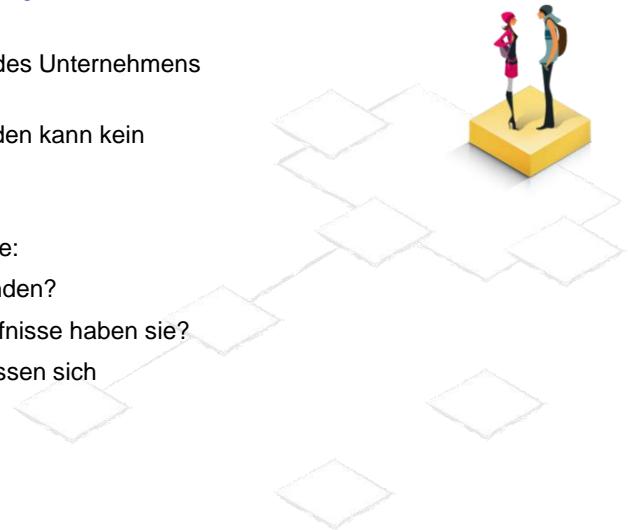
Die meisten Daten entstehen intern

- Das Einsatzgebiet von IS ist gleichzeitig die wichtigste Informationsquelle



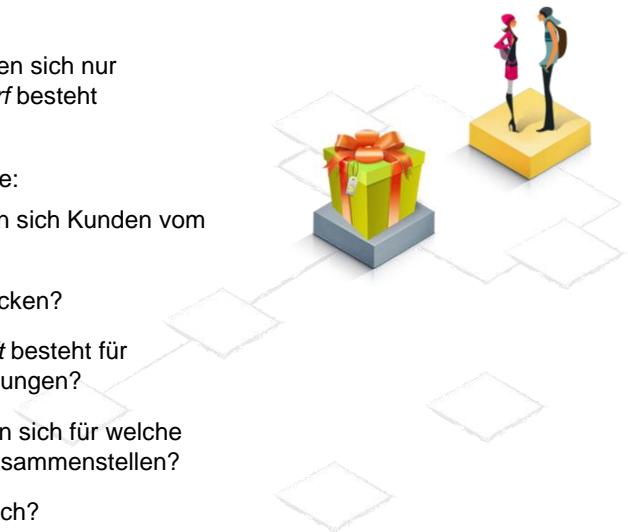
Kunden bilden die Basis für jedes Geschäftsmodell

- Zentrale Orientierungsgröße jedes Unternehmens sollte der *Kunde* sein
 - Ohne gewinnbringende Kunden kann kein Unternehmen überleben
- IS liefern Hinweise zu Fragen wie:
 - Wer sind die wichtigsten Kunden?
 - Welche Wünsche und Bedürfnisse haben sie?
 - Welche Kundensegmente lassen sich unterscheiden?



Kundenbedürfnisse erfassen

- Güter und Dienstleistungen lassen sich nur verkaufen, wenn dafür ein *Bedarf* besteht
- IS liefern Hinweise zu Fragen wie:
 - Welchen *Nutzen* versprechen sich Kunden vom Angebot?
 - Wie lässt sich der *Bedarf* wecken?
 - Welche *Zahlungsbereitschaft* besteht für welche Güter und Dienstleistungen?
 - Welche *Produktbündel* lassen sich für welche Kundensegmente sinnvoll zusammenstellen?
 - Wie viel *Qualität* ist erforderlich?



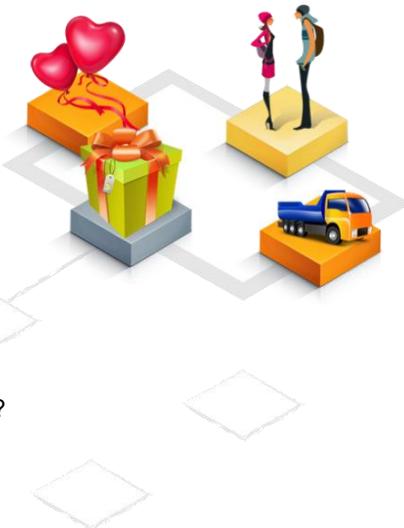
Kunden müssen erreicht werden

- Um mit Kunden in Kontakt zu treten, müssen die richtigen *Verkaufs-, Distributions- und Kommunikationskanäle* etabliert werden
- IS liefern Hinweise zu Fragen wie:
 - Über welche *Kanäle* wird der Markt am besten erreicht?
 - Wie erfahren Kunden von Vorteilen der Angebote?
 - Wie kommen Kunden an Güter & Dienstleistungen?
 - Welcher Kanal ist am *wirtschaftlichsten*?
 - Welcher wird von den Kunden *bevorzugt*?



Kunden wollen betreut werden

- Um neue Kunden zu gewinnen und bestehende zu behalten, muss ein *Vertrauensverhältnis* geschaffen werden
- IS liefern Hinweise zu Fragen wie:
 - Welche *Massnahmen* sind zur Gewinnung von Neukunden am effektivsten?
 - Welche *Massnahmen* können die Verkäufe steigern?
 - Kann/soll der *Kundenservice* automatisiert werden?
 - Kann/soll der Kunde bei der *Ausgestaltung* von Gütern & Dienstleistungen miteinbezogen werden?



Einnahmen sichern Unabhängigkeit und Anpassungsfähigkeit

- *Gewinn, Rentabilität und Liquidität* sind u.a. Bedingungen für das langfristige Überleben des Unternehmens
- IS liefern Hinweise zu Fragen wie:
 - Welche *Preise und Zahlungsbedingungen* können verlangt, welche Rabatte gewährt werden?
 - Wie hoch sind die *laufenden Einnahmen*?
 - Wofür geben welche Kunden wie viel aus?
 - Wie *entwickeln sich einzelne Segmente* über die Zeit?
 - Welche Massnahmen bewirken welches Einnahmewachstum?



Schlüsselressourcen sind ein entscheidender Faktor im Wettbewerb

- Der Zugang zu den *Schlüsselressourcen* ist für das Bestehen im Wettbewerb zwingend
 - Personal, Patente, Rohstoffe
- IS liefern Hinweise zu Fragen wie:
 - Welche *Ressourcen sind notwendig* für Produktion, Vertrieb, Kundenbetreuung?
 - Wie können Ressourcen *langfristig gesichert* werden?



Partner vergrössern die Erfolgschancen

- Gute Beziehungen zu *Lieferanten und Partnern* öffnen zusätzliche Möglichkeiten
- IS liefern Hinweise zu Fragen wie:
 - Welches sind unsere wichtigsten *Lieferanten*?
 - Wer sind potentielle *Partner*?
 - Welche *Ressourcen/ Kernkompetenzen* einkaufen? Welche können durch Partnerschaften gesichert werden?



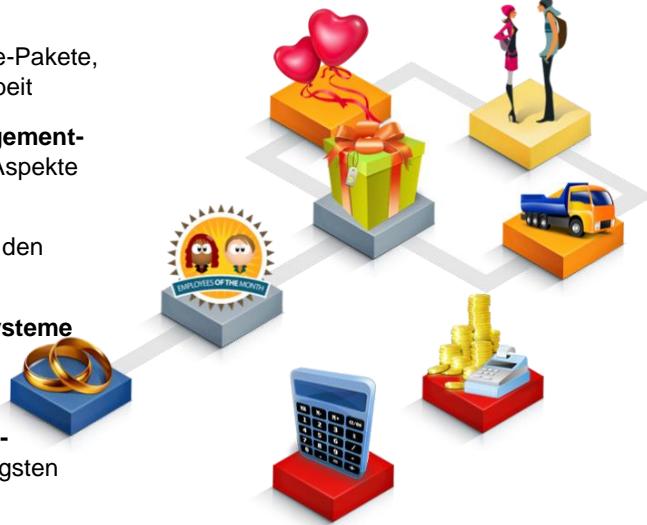
Kostenkalkulation zeigt Schwachstellen auf

- Beschaffung, Produktion, Vertrieb und Verwaltung bestimmen die *Kostenstruktur*
- IS liefern Hinweise zu Fragen wie:
 - Welches sind die *Kostentreiber*?
 - Welche *Schlüsselressourcen / Kernkompetenzen* sind am teuersten?
 - Wie unterscheiden sich die *Produktionskosten* der Güter & Dienstleistungen?
 - Wie sieht die *Kostenstruktur* aus?
 - Wo treten *Skalen- oder Verbundeffekte* auf?



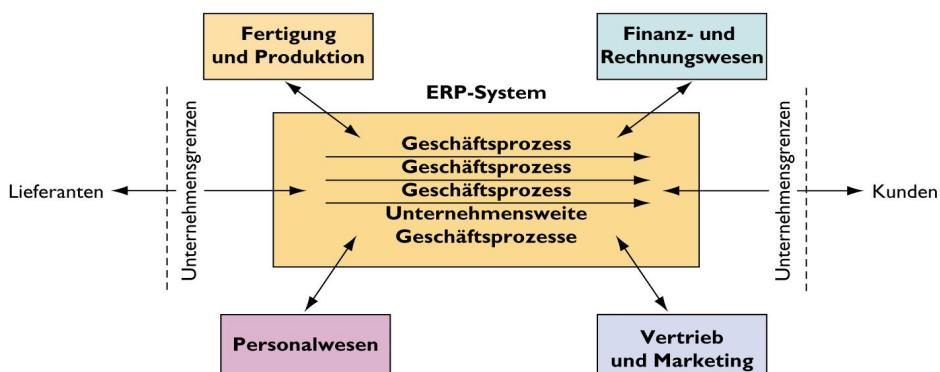
Zahlreiche IS unterstützen Unternehmen im täglichen Betrieb

- **Individualanwendungen** (Office-Pakete, Projektplaner) erleichtern die Arbeit
- **Customer-Relationship-Management-Systeme** betrachten sämtliche Aspekte der Kundenbeziehung
- **Buchhaltungssysteme** sichern den Durchblick bei den Finanzen
- **Supply-Chain-Management-Systeme** optimieren komplett Wertschöpfungsketten
- **Enterprise-Resource-Planning-Systeme** koordinieren die wichtigsten Geschäftsprozesse



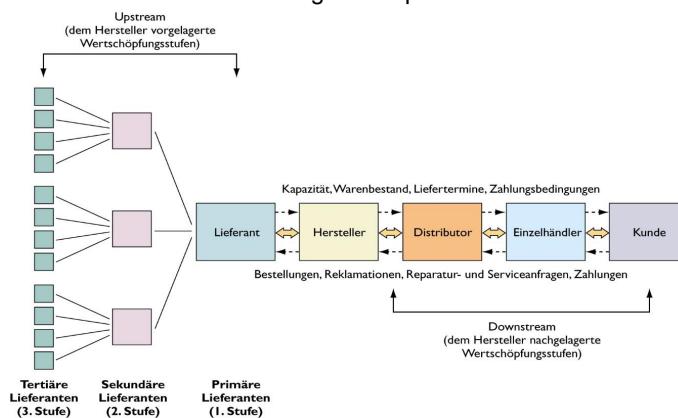
Enterprise-Resource-Planning (ERP)

Integrierte unternehmensweite Anwendungssysteme, die zur Koordination wichtiger interner Prozesse eines Unternehmens dienen.



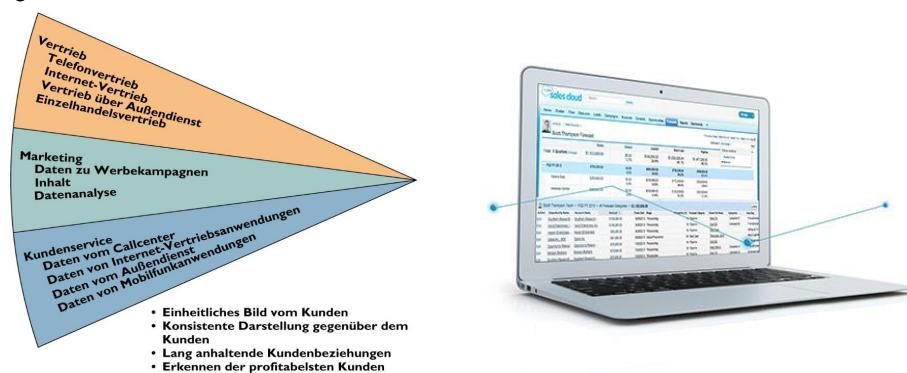
Supply-Chain-Management (SCM)

Anwendungssysteme, die den Informationsaustausch zwischen einem Unternehmen und seinen Lieferanten und Kunden automatisieren, um Planung, Beschaffung, Fertigung und Vertrieb von Produkten und Dienstleistungen zu optimieren.



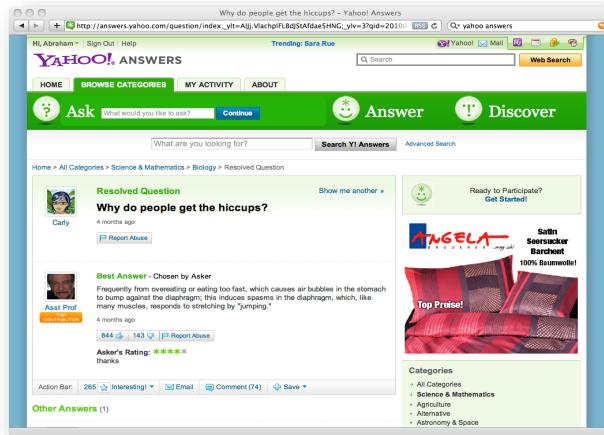
Customer-Relationship-Management (CRM)

Analyse von Kunden aus unterschiedlichen Perspektiven. Sämtliche Aspekte der Kundenbeziehung werden berücksichtigt, wie z.B. Kundenservice, Vertrieb und Marketing



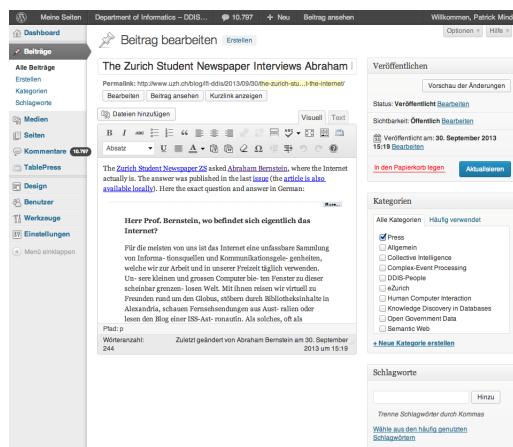
Wissensmanagement

Umfasst Systeme, die den Erwerb, die Erfassung, Speicherung und Weitergabe von Wissen und Fachkenntnissen unterstützen.



Content-Management-Systeme (CMS)

CMS ist Software, die bei der *kollaborativen* Erarbeitung und Organisation von Inhalten unterstützt.



Fazit

- IS finden in allen Funktionen der betrieblichen Wertschöpfung Verwendung
- Ihr Einsatz in der Beschaffung, Produktion, Absatz und Verwaltung liefert gleichzeitig die Daten für analytische Informationssysteme
- Durch Automatisierung können Kosten reduziert resp. Gewinne gesteigert werden

Credits

- Osterwalder, A. und Pigneur, Y. (2011): Business Model Generation, Campus, Frankfurt
- Images & Icons
 - Balloons by IconDrawer
 - 3D calculator by BrightKnight
 - GlamourGirls by Artshare
 - Glossy Business Icons by (xooplate)
 - Gifts icon by Miniartx
 - Gold Coins by Antialiasfactory
 - Magnifyingglass by Ataei
 - Truck Icon by cemagraphics

7.2 Wertschöpfungskette



Institut für Informatik

Arten von Informationssystemen II – Wertschöpfungskette

Prof. Abraham Bernstein, Ph.D.

Arten von Informationssystemen II – Wertschöpfungskette

- #### ▪ IS Wertschöpfungskette

Die Entwicklung von Informationssystemen ist auf verschiedene Interessen des Unternehmens ausgerichtet

„Wir erhalten so viele Nachrichten, ungefragt und aus so vielen Quellen, in unterschiedlichen Formen und Konzentrationen, dass Information zu einer Art Müll wurde.“

Neil Postman in: Hoechst „Future Magazin“ 1/199



Kein System kann allein sämtliche Informationen bereitstellen, die benötigt werden

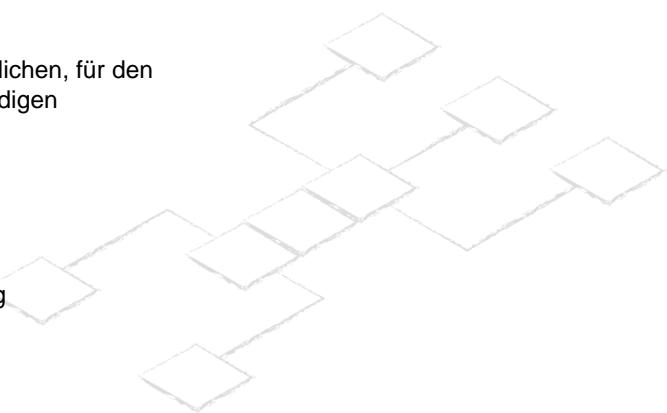
- Informationssysteme helfen, dass
 - die relevanten *Informationen*
 - dem berechtigten *Empfänger*
 - im richtigen *Augenblick*
 - für den geeigneten *Zweck*
 - auf gewünschte *Art und Weise*



zur Verfügung stehen

IS helfen bei der Abwicklung und Planung der Geschäftsprozesse

- Informationssysteme unterstützen:
 - operative Abläufe mit täglichen, für den Geschäftsbetrieb notwendigen Routinetransaktionen
 - Unternehmensführung
 - Planung
 - Entscheidungsfindung
 - Kontrolle



IS-Wertschöpfungskette

- Funktionsweise von IS lässt sich wie folgt generalisieren:

- Daten
 - erfassen
 - verarbeiten
 - speichern
 - analysieren
 - bereitstellen



Die meisten Daten entstehen intern

- Einsatzgebiet von IS ist gleichzeitig die wichtigste Informationsquelle
 - Operative Daten aus Transaktionen
 - Daten stammen i.d.R. aus unterschiedlichen Systemen



Externe Quellen liefern ergänzende Daten

- WWW
- Medien
- Externe Datenbanken und Informationsdienste (Bloomberg, Reuters, Telekurs etc.)
- Marktanalysen, Umfragen
- Institutionen, Staat



Daten werden aufbereitet und längerfristig gespeichert

- *Extraktion* der Daten aus internen und externen Quellen
- *Transformation*: Auswahl der relevanten Daten, Fehlerbereinigung, Konsolidierung, Aggregation
- *Dauerhaftes Speichern* in einer Datenbank
- *Archivierung / gesetzliche Aufbewahrungspflichten*
- Problem: *Kompatibilität* alter Daten mit neuen Systemen



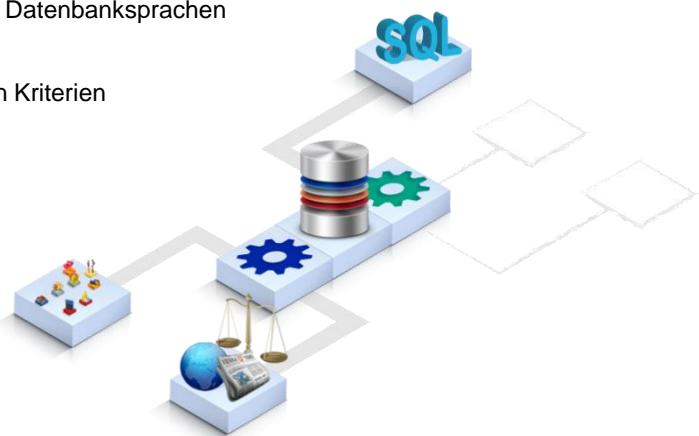
Datenbanken ermöglichen die Verarbeitung grosser Datenmengen

- Daten stehen aufbereitet und in einheitlichem Format zur Verfügung
- Datenbanken dienen als Datenbasis für alle Arten von Auswertungen und Analysen
 - **Reporting**
Was ist in der Vergangenheit passiert?
 - **Analyse**
Warum ist etwas passiert?
 - **Überwachung**
Was passiert aktuell?
 - **Prognose**
Was kann in der Zukunft passieren?



Bedarfsgerechter Datenzugriff: Freie Datenbankrecherche

- Datenabfrage mithilfe von Datenbanksprachen
- Analyse nach individuellen Kriterien



Bedarfsgerechter Datenzugriff: Analysesysteme (I)

- Modellgestützte Analysesysteme

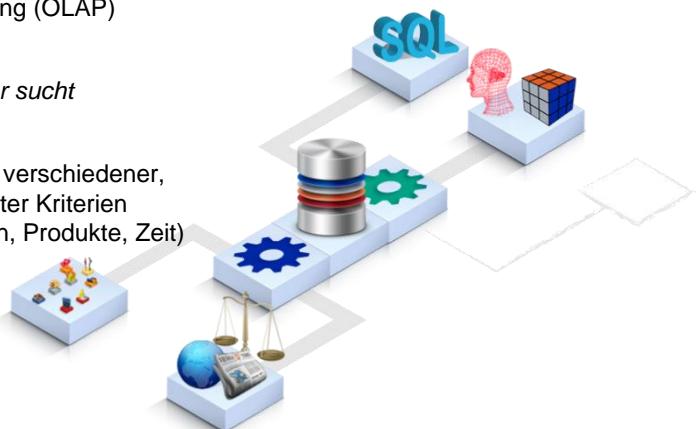
- *Beruhen auf mathematischen Modellen, Statistik, künstlicher Intelligenz*
 - Beispiel Data Mining:
 - *Benutzer weiss nicht, was er sucht*
 - Daten werden nach *bisher unbekannten Mustern und verborgenen Zusammenhängen durchsucht*



Bedarfsgerechter Datenzugriff: Analysesysteme (II)

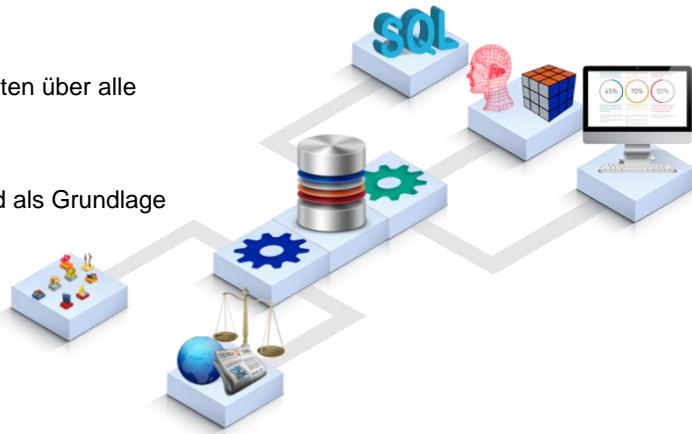
- Online Analytical Processing (OLAP)

- *Benutzer weiss, was er sucht*
 - Daten werden anhand verschiedener, gleichzeitig ausgewählter Kriterien analysiert (z.B. Kunden, Produkte, Zeit)



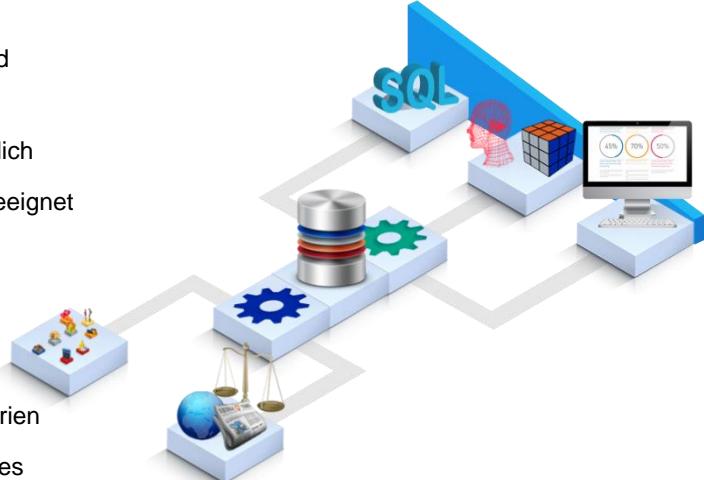
Bedarfsgerechter Datenzugriff durch passendes IS

- Berichtssysteme
 - Erstellen von Übersichten über alle relevanten Daten
 - Dient der Kontrolle und als Grundlage für Entscheidungen



Zunehmende Flexibilität setzt höhere IT-Kompetenz voraus

- Datenbankrecherchen sind
 - Sehr flexibel
 - Wenig benutzerfreundlich
 - Für versierte Nutzer geeignet
- Berichtssysteme
 - Leicht erlernbar
 - informieren anhand vorselektionierter Kriterien
 - Benötigen kein vertieftes Informatikverständnis



Fazit

- Die IS Wertschöpfungskette besteht aus Daten erfassen, verarbeiten und bereitstellen
- Daten können von internen wie auch externen Quellen stammen
- Aufbereitete Daten können für Reporting, Analyse, Überwachung sowie Prognose genutzt werden
- Dazu gibt es verschiedene Systeme, die unterschiedliche Anwendungen ermöglichen und entsprechend auch unterschiedliche Kompetenzen erfordern

7.3 Klassifizierung



Institut für Informatik

Arten von Informationssystemen III – Klassifizierung

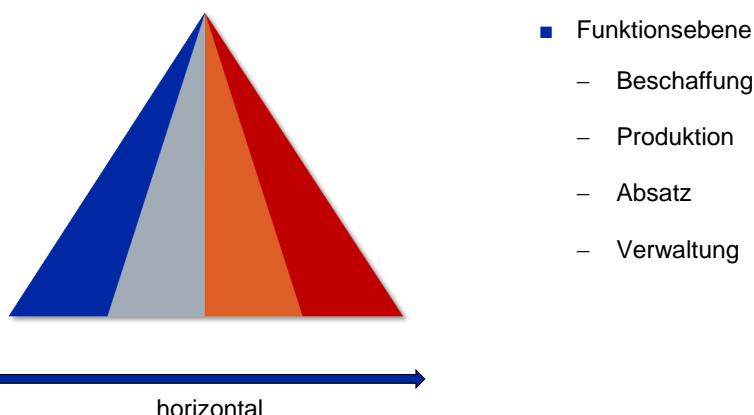
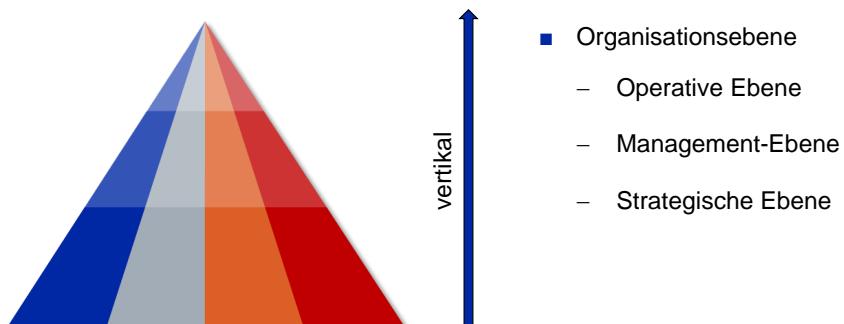
Prof. Abraham Bernstein, Ph.D.

Arten von Informationssystemen III – Klassifizierung

- Klassifizierung von IS

Klassifizierungskriterien im Überblick

Funktion (betriebliche Wertschöpfungskette)	Beschaffung	Produktion	Absatz	Verwaltung
Adressatenhierarchie	Operative Ebene	Mittleres Management	Oberes Management	Verwaltungsrat
Adressatenzahl	Einzelperson		Gruppe	
Informationsherkunft	Interne Quellen		Externe Quellen	
Informationsart	Quantitativ	Qualitativ	Strukturiert	Unstrukturiert
Informationsauslöser	Ereignis / Signal		Termin	Benutzerwunsch
Zweck	Reporting	Analyse	Kontrolle	Prognose
Abfragemodus	Freie Abfrage	Abfrage mit Parametervariation		Vordefinierte Abfrage
Aufbereitung	Tabelle		Grafik	Bericht
Informationsverteilung	Pull		Push	

Klassifizierung kann gemäss horizontaler Integration erfolgen**Klassifizierung kann gemäss vertikaler Integration erfolgen**

Operative Systeme

Beispiel: Modulbuchungssystem der Universität

 EMPLOYEES OF THE MONTH	Ausführen und Aufzeichnen der täglichen, für den Geschäftsbetrieb notwendigen Routinetransaktionen			
	Operative Ebene	Mittleres Management	Oberes Management	Verwaltungsrat
	Einzelperson		Gruppe	
	Interne Quellen			Externe Quellen
	Quantitativ	Qualitativ	Strukturiert	Unstrukturiert
	Ereignis / Signal		Termin	Benutzerwunsch
	Reporting	Analyse	Kontrolle	Prognose
	Freie Abfrage	Abfrage mit Parametervariation	Vordefinierte Abfrage	
	Tabelle	Grafik	Bericht	
	Pull		Push	

Managementinformationssysteme (MIS)

Beispiel: System eines Supermarktes

 MANAGER OF THE YEAR	Stellen Standardübersichtsberichte bereit, Helfen bei der Planung, Kontrolle und Entscheidungsfindung			
	Operative Ebene	Mittleres Management	Oberes Management	Verwaltungsrat
	Einzelperson		Gruppe	
	Interne Quellen			Externe Quellen
	Quantitativ	Qualitativ	Strukturiert	Unstrukturiert
	Ereignis / Signal		Termin	Benutzerwunsch
	Reporting	Analyse	Kontrolle	Prognose
	Freie Abfrage	Abfrage mit Parametervariation	Vordefinierte Abfrage	
	Tabelle	Grafik	Bericht	
	Pull		Push	

Entscheidungsunterstützungssysteme (EUS)

Beispiel: System über dem MIS eines Supermarktes

	Kombinieren Daten mit ausgeklügelten analytischen Modellen oder Werkzeugen Unterstützung schwach strukturierter oder unstrukturierter Entscheidungsfindungsprozesse			
	Operative Ebene	Mittleres Management	Oberes Management	Verwaltungsrat
	Einzelperson		Gruppe	
Informationsherkunft	Interne Quellen		Externe Quellen	
Informationsart	Quantitativ	Qualitativ	Strukturiert	Unstrukturiert
Informationsauslöser	Ereignis / Signal		Termin	Benutzerwunsch
Zweck	Reporting	Analyse	Kontrolle	Prognose
Abfragemodus	Freie Abfrage	Abfrage mit Parametervariation		Vordefinierte Abfrage
Aufbereitung	Tabelle	Grafik		Bericht
Informationsverteilung	Pull		Push	

Führungsunterstützungssysteme (FUS)

Beispiel: Unterstützungssystem für die Führungsebene einer Organisation

	Systeme auf der strategischen Ebene Unterstützung der unstrukturierten Entscheidungsfindung, insbesondere durch erweiterte Grafik- und Kommunikationsfunktionen			
	Operative Ebene	Mittleres Management	Oberes Management	Verwaltungsrat
	Einzelperson		Gruppe	
Informationsherkunft	Interne Quellen		Externe Quellen	
Informationsart	Quantitativ	Qualitativ	Strukturiert	Unstrukturiert
Informationsauslöser	Ereignis / Signal		Termin	Benutzerwunsch
Zweck	Reporting	Analyse	Kontrolle	Prognose
Abfragemodus	Freie Abfrage	Abfrage mit Parametervariation		Vordefinierte Abfrage
Aufbereitung	Tabelle	Grafik		Bericht
Informationsverteilung	Pull		Push	

Probleme bei der Klassifizierung

- Entscheidungsunterstützung kann auf *jeder Ebene* einer Firma passieren
- Andere Systeme sind meist entweder für Operative-, Management- oder Führungsebene gebaut
- Keine klare Trennung möglich.
Je nach *Zielgruppe, Art der Verarbeitung* von Informationen und anderen Klassifizierungskriterien können sich Systeme unterscheiden und auch vermischen

Fazit

- Informationssysteme helfen, Daten adressatengerecht aufzubereiten
- Es existieren vielfältige Lösungen entsprechend den Anforderungen der Anwender
 - Operative Systeme unterstützen die täglichen, für den Geschäftsbetrieb notwendigen **Routinetransaktionen**
 - Management-Informationssysteme (MIS) helfen bei der **Planung, Kontrolle und Entscheidungsfindung**
 - Nutzen viele Grunddaten der operativen Systeme
 - Entscheidungsunterstützungssysteme (EUS) unterstützen in **Entscheidungssituationen**
 - Führungsunterstützungssysteme (FUS) sind **konsequent auf die oberen Führungsebenen** ausgerichtet und bieten leistungsstarke Visualisierungs- und Analysetools

Credits

- Eckerson, W. (2007): Predictive Analytics. Extending the Value of Your Data Warehousing Investment. In: TDWI Best Practices Report
- Lehner, F. und Fteimi, N.: Wissensmanagementsysteme, in: WISU 11/2012, S. 1474-1480
- Mertens P. et al. (2012): Grundzüge der Wirtschaftsinformatik, Springer-Verlag, Berlin
- Postman, N. (1999): Wir amüsieren uns zu Tode, Interview in Hoechst „Future Magazin“1/99
- Images & Icons
 - Database by barrymieny
 - Glossy Business Icons by vectorarts.net
 - Magnetic poetry by surrealmouse
 - Wire Head by Nick2001
 - Magnifyingglass by Ataei

8 Wert von IS und Information

PROF. ABRAHAM BERNSTEIN

8.1 Teil I



Institut für Informatik

Wert von Informationssystemen I

Prof. Abraham Bernstein, Ph.D.

Wert von Informationssystemen I

Wie werden Investitionen bewertet?

- **Bewertungen von Investitionen**
- Kapitalwertmethode
- Entscheidungsbaumverfahren
- Realoptionsmethode

Investitionen wollen gut überlegt sein

“Wrong decisions are an inevitable part of life. But bad decisions are unforced errors. They’re eminently avoidable - and there are proven techniques to avoid the most predictable pitfalls.”

(“Decisions, Decisions”, Fortune-Magazin, 27.06.05)

“I think there is a world market for maybe five computers”

T.J. Watson, chairman and
CEO of IBM, 1943
(unbelegt)

In der Praxis existieren zahlreiche Bewertungsverfahren

- Klassische Kosten-Nutzen-Analysen
(u.a. Kapitalwertmethode)
- Optionstheoretische Modelle
- Entscheidungsbaumverfahren
- ... (zahlreiche weitere)



In der Praxis steht die Finanzperspektive im Vordergrund

- **Finanzperspektive**
 - Zukunftspotential, tiefere Stückkosten, Umsatz, Cashflow
- **Innovationsperspektive**
 - Besseres Image, Basis für Investitionen
- **Kundenperspektive**
 - Marktpotentiale, Kundenzufriedenheit und –treue

In der Praxis steht die Finanzperspektive im Vordergrund

- **Prozessperspektive**
 - Effizienz, Fehlerreduktion, Produktivitätssteigerung
- **Mitarbeiterperspektive**
 - Ergonomie, IT Kompetenz, Mitarbeiterzufriedenheit
- **Sicherheitsperspektive**
 - Datensicherheit und –schutz

Bei der Bewertung spielen die Investitionsziele eine wichtige Rolle

	Transaktionsorientierte IS-Investitionen	Informationsorientierte IS-Investitionen	Strategieorientierte IS-Investitionen
Ziel	Vereinfachung bestehender Abläufe durch Automatisierung	Informationen für Unternehmenssteuerung	Wettbewerbsvorteile
Aspekt	Mehrwert	Beseitigung von Unsicherheit	Handlungsspielraum
Wert IS	Zahlungsströme	Informationswert	Flexibilitätswert

Wert von Informationssystemen I

Wie werden Investitionen bewertet?

- Bewertungen von Investitionen
- **Kapitalwertmethode**
- Entscheidungsbaumverfahren
- Realoptionsmethode

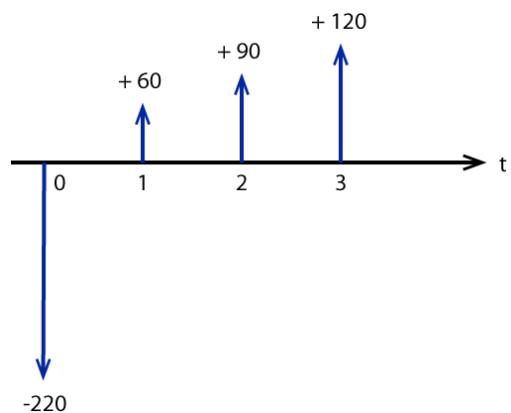
Kapitalwertmethode

Wert des IS = Nettoegenwartswert der Einnahmeüberschüsse

- Berücksichtigt Zeitwert zukünftiger Zahlungsströme
- Durch Investition bedingte Einnahmeüberschüsse werden diskontiert:

$$\sum_{t=1}^T \frac{(\text{Einzahlungen}_t - \text{Auszahlungen}_t)}{(1+i)^t}$$

- Investition, falls (erwarteter) Wert zukünftiger Einnahmeüberschüsse > Investitionskosten
- Geeignet bei gut abschätzbarer Erträgen und Risiken



Kapitalwertmethode

Wert des IS = Netto gegenwartswert der Einnahmeüberschüsse

Beispiel: Ein neues CRM-System soll angeschafft werden

- Heutige Investition: 220
- Einnahmeüberschüsse in den Folgejahren: 60, 90, 120
- Der risikogerechte Diskontierungssatz beträgt 10%

	Ende Jahr 0	Ende Jahr 1	Ende Jahr 2	Ende Jahr 3
Investition	-220			
Mehrumsetz		100	120	150
Betriebsausgaben		-40	-30	-30
Saldo		+ 60	+ 90	+ 120
Barwert	-220	54.5	74.4	90.2
NPV	-0.9			

Kapitalwertmethode

Wert des IS = Netto gegenwartswert der Einnahmeüberschüsse

$$\text{Netto gegenwartswert} = \sum_{t=1}^T \frac{(\text{Einzahlungen}_t - \text{Auszahlungen}_t)}{(1+i)^t}$$

Varianten:

- Reingewinn
- Free Cash Flow
- ... (weitere)

Varianten:

- Risikofreier Basiszins
- Rendite bestmöglicher Alternative
- Weighted Average Cost of Capital
- ... (weitere)

Kapitalwertmethode**Wert des IS = Nettogegenwartswert der Einnahmeüberschüsse**

Kritik:

- „Jetzt-oder-nie“-Entscheidung
- Wert neuer Information wird nicht berücksichtigt
- Handlungsspielraum des Managements (z.B. Umentscheidung) wird vernachlässigt

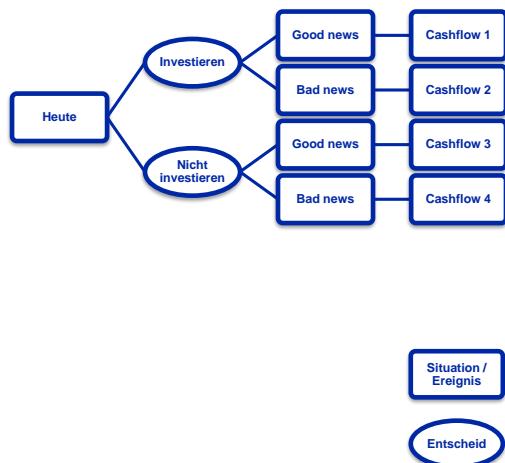
Wert von Informationssystemen I

Wie werden Investitionen bewertet?

- Bewertungen von Investitionen
- Kapitalwertmethode
- **Entscheidungsbaumverfahren**
- Realoptionsmethode

Entscheidungsbaumverfahren

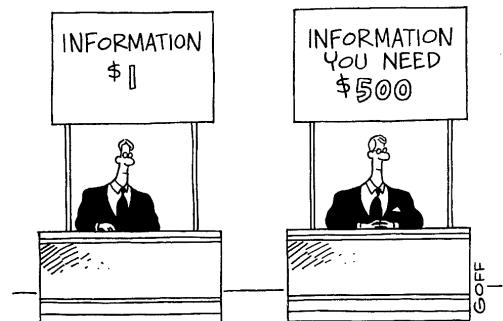
- Betrachtet Investitionsentscheidungen und ihre Folgen im Zeitablauf
 - Mehrstufiger Entscheidungsprozess wird als *Entscheidungsbaum* abgebildet
 - Mögliche *Ereignisse und ihre Eintrittswahrscheinlichkeiten* werden festgehalten
 - *Entscheidungsalternativen* werden mit erwarteten Folgen notiert
- Mögliche Ergebnisse einzelner Handlungsfolgen werden mit Hilfe von *Entscheidungsregeln* bewertet



Entscheidungsbaumverfahren

Wert des IS = Informationswert

- Berücksichtigt gewonnene Entscheidungssicherheit dank Information
- Wert IS = Informationswert = Mehrwert durch verbesserte Entscheidungsfindung
- Investition erfolgt, falls $Informationswert > Kosten IS$
- Geeignet bei abschätzbaren Risiken und hohem Handlungsspielraum

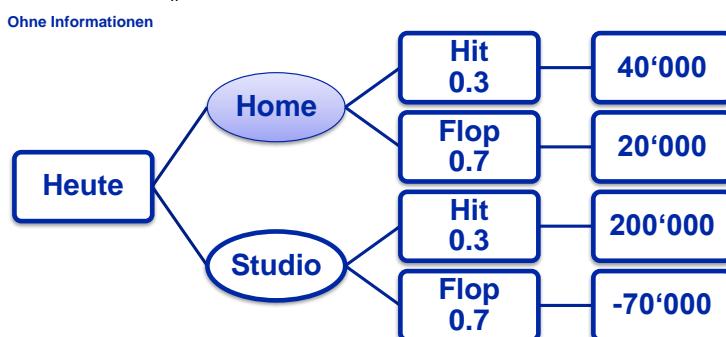


Entscheidungsbaumverfahren

Wert des IS = Informationswert

$$E_H = 0.3 \times 40'000 + 0.7 \times 20'000 = 26'000$$

Ohne Informationen



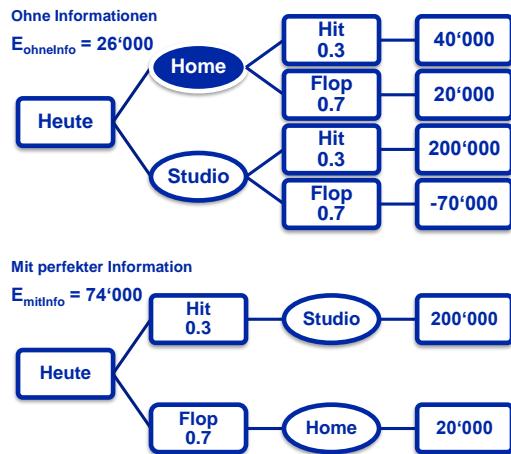
$$E_S = 0.3 \times 200'000 + 0.7 \times -70'000 = 11'000$$



Entscheidungsbaumverfahren

Wert des IS = Informationswert

- Berücksichtigt gewonnene Entscheidungssicherheit dank Information
 - Wert IS = Informationswert = Mehrwert durch verbesserte Entscheidungsfindung
 - Investition erfolgt, falls $Informationswert > Kosten IS$
 - Geeignet bei abschätzbaren Risiken und hohem Handlungsspielraum
- Wert IS = $E_{mitInfo} - E_{ohneInfo} = 48'000$**
- Für genauere Erläuterungen siehe Selbstlernmodul



Entscheidungsbaumverfahren

Wert des IS = Informationswert

Kritik:

- Wahrscheinlichkeiten der Ereignisse (bspw. Hit/Flop) müssen bekannt sein
- Pay-offs für jeden möglichen Fall müssen ebenfalls bekannt sein

Wert von Informationssystemen I	<p>Wie werden Investitionen bewertet?</p> <ul style="list-style-type: none"> ▪ Bewertungen von Investitionen ▪ Kapitalwertmethode ▪ Entscheidungsbaumverfahren ▪ Realoptionsmethode
--	--

Realoptionsmethode																	
<ul style="list-style-type: none"> ■ Realoptionsmethode berücksichtigt Handlungsspielräume des Managements <ul style="list-style-type: none"> – Zeitpunkt und Ausmass der Investition i.d.R. frei wählbar – Manager können Projekt aufschieben, pausieren, modifizieren, abbrechen ■ Flexibilität hat einen Wert, der in Analogie zu Finanzoptionen modelliert und berechnet werden kann ■ Geeignet bei hohen Risiken und Handlungsspielraum 																	
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Eine</th><th style="text-align: center;">Aktienoption</th><th style="text-align: center;">Realoption</th></tr> </thead> <tbody> <tr> <td style="text-align: center;">bezeichnet das Recht, aber nicht die Pflicht</td><td></td><td></td></tr> <tr> <td style="text-align: center;">einen Vermögenswert</td><td style="text-align: center;">Aktie</td><td style="text-align: center;">Nutzen aus der im Projekt erarbeiteten Lösung</td></tr> <tr> <td style="text-align: center;">zu kaufen</td><td style="text-align: center;">Call</td><td style="text-align: center;">zB. Warteoption</td></tr> <tr> <td style="text-align: center;">oder zu verkaufen</td><td style="text-align: center;">Put</td><td style="text-align: center;">zB. Abbruchoption</td></tr> </tbody> </table>			Eine	Aktienoption	Realoption	bezeichnet das Recht, aber nicht die Pflicht			einen Vermögenswert	Aktie	Nutzen aus der im Projekt erarbeiteten Lösung	zu kaufen	Call	zB. Warteoption	oder zu verkaufen	Put	zB. Abbruchoption
Eine	Aktienoption	Realoption															
bezeichnet das Recht, aber nicht die Pflicht																	
einen Vermögenswert	Aktie	Nutzen aus der im Projekt erarbeiteten Lösung															
zu kaufen	Call	zB. Warteoption															
oder zu verkaufen	Put	zB. Abbruchoption															
Quelle: Hilpisch, Options Based Management																	

Vergleichende Betrachtung der Bewertungsmodelle

	Kapitalwert-methode	Entscheidungsbaumverfahren	Optionstheore-tische Modelle
Klarheit der Aussage	Hoch	Hoch	Mittel
Ganzheitlichkeit	Gering	Mittel	Mittel
Nachvollziehbarkeit	Hoch	Hoch	Gering
Aufwand	Gering	Mittel	Hoch
Praxisrelevanz	Hoch	Hoch	Gering

Fazit

- Für die Beurteilung einer IS-Investition spielen die damit verbundenen Ziele und die Beurteilungsperspektive eine Rolle
- In der Praxis haben sich zahlreiche Methoden etabliert, welche sich auf die finanziellen Auswirkungen fokussieren
- Die Verfahren unterscheiden sich hinsichtlich ihrer zugrundeliegenden Annahmen, der Berücksichtigung von Unsicherheit und ihrer Komplexität
- In der Praxis werden i.d.R. einfach anwendbare Methoden bevorzugt

Credits

- Hilpisch, Y. (2006): Options Based Management, GWV Fachverlage GmbH, Wiesbaden
- Jonen, A. / Lingnau, V. / Müller, J. / Müller, P. (2004): Balanced IT-Decision Card - Ein Instrument für das Investitionscontrolling von IT-Projekten. In: Wirtschaftsinformatik, 46 (2004) Schwerpunkttheft Nr. 3, 196 - 203.
- Stübner, M. (2012): Ein ressourcentheoretischer Ansatz zur strategischen Bewertung von Informationssystemen, Fakultät für Maschinenbau, Elektrotechnik und Wirtschaftsingenieurwesen der Brandenburgischen Technischen Universität Cottbus

8.2 Teil II



Universität
Zürich UZH

Institut für Informatik

Wert von Informationssystemen II

Prof. Abraham Bernstein, Ph.D.

Wert von Informationssystemen II

Wann lohnen sich Investitionen in IS?

- Kosten
- Nutzen
- Unsicherheit

Wann lohnen sich Investitionen in IS?

- Investitionen lohnen sich, wenn *Nutzen > Kosten*
- Herausforderung: Beziffern von Kosten und Nutzen *unter Unsicherheit*
 - Kosten: Ermittlung i.d.R. unproblematisch

Wann lohnen sich Investitionen in IS?

Kosten

- Direkt
 - Einmalig
 - Laufend

Indirekt

Total Cost of Ownership

- Anschaffung Hard-/Software
- Projektmanagement
- Migrationskosten (Schulung, Daten)
- Schulung
- Technischer Support, Wartung
- Arbeitszeitverluste (Schulung, technische Probleme)
- Produktivitätsverluste (Einarbeitungszeit)
- Für den Investitionsentscheid relevant:
 - Gesamtkosten, die direkt oder indirekt über die Lebensdauer der Investition anfallen

Wert von Informationssystemen II

Wann lohnen sich Investitionen in IS?

- Kosten
- Nutzen
- Unsicherheit

Wann lohnen sich Investitionen in IS?

- Investitionen lohnen sich, wenn *Nutzen > Kosten*
- Herausforderung: Beziffern von Kosten und Nutzen *unter Unsicherheit*
 - Kosten: Ermittlung i.d.R. unproblematisch
 - Nutzen: Erfassen und Quantifizieren anspruchsvoll

Nutzen

- Grundsätzlich: Dank IS werden
 - ▶ Informationen schneller bereitgestellt
 - ▶ Mehr und bessere Informationen geliefert
 - ▶ Neue Handlungsalternativen erkannt
 - ▶ Handlungskonsequenzen besser eingeschätzt
 - ▶ Unsicherheiten eliminiert
- IS ermöglichen bessere Entscheidungen

Wann lohnen sich Investitionen in IS?

Nutzen

- Direkt
 - ▶ Quantifizierbar
 - Bewertbar

Das reine Halten von Information hat einen eher geringen Nutzen

- Aber: Wenn dank Informationen bessere Entscheide gefällt werden hat dies messbare und monetär bewertbare Folgen
- Besserer Ressourceneinsatz → tiefere Kosten
 - ▶ Reduzierte Lagerkosten
 - ▶ Geringerer Personalbedarf
 - ▶ Materialeinsparungen
 - Höhere Produktivität

Exkurs: Das Produktivitätsparadoxon „Mehr IT führt nicht zu höherer Produktivität“

- Frühere Studien: Kein Zusammenhang zwischen IT-Investitionen und Produktivitätsgewinnen
- Neuere Studien widerlegen das Produktivitätsparadoxon mehrheitlich
- Mögliche Gründe für widersprüchliche Aussagen:
 - Methodische Mängel
 - Quantifizierungsprobleme
 - Ignorieren von Verzögerungseffekten

Untersuchte Ebene	Welche Auswirkungen haben IS auf die Untersuchungsergebnisse (Produktivität, Gewinn, oder Geschäftswert)			
	Positive	Negative	Keine Effekte	Unklar
Volkswirtschaft	1	0	1	3
Branche	6	0	1	5
Unternehmen	47	2	1	23

Quelle: Wan, A Ten-Year Odyssey of the 'IS Productivity Paradox'

Wann lohnen sich Investitionen in IS?

Nutzen

- Direkt
- Indirekt
 - Quantifizierbar
 - Bewertbar
 - Nicht bewertbar
 - Nicht quantifizierbar

- Effekte ergeben sich verzögert

- Nutzen fallen in anderen Bereichen, Funktionen oder Prozessen an als vorgesehen
 - Paketverfolgung
 - Geldautomaten: Betriebskosten > erzielte Einnahmen
- Indirekte Nutzeneffekte sind oft nicht quantifizierbar oder nicht bewertbar
 - Kundenzufriedenheit
 - Image
 - Strategische Effekte

Wert von Informationssystemen II

Wann lohnen sich Investitionen in IS?

- Kosten
- Nutzen
- **Unsicherheit**

Wann lohnen sich Investitionen in IS?

- Investitionen lohnen sich, wenn *Nutzen > Kosten*
- Herausforderung: Beziffern von Kosten und Nutzen *unter Unsicherheit*
 - Kosten: Ermittlung i.d.R. unproblematisch
 - Nutzen: Erfassen und Quantifizieren anspruchsvoll
 - Unsicherheiten erschweren Bewertung
 - Kapital- und Ressourcenbedarf
 - Komplexität
 - Projektbeteiligte
 - Realisierungsdauer
 - Zukünftige Zahlungsströme
 - Rechtlich-politische Rahmenbedingungen

Fazit

- Ob sich eine IS-Investition lohnt, hängt von den damit verbundenen Zielen und von der Beurteilungsperspektive ab
- Im Allgemeinen lohnen sich Investitionen, wenn der damit verbundene Nutzen die Kosten übersteigt
- In der Praxis sind die Kosten einfacher, der Nutzen schwieriger zu ermitteln
- Unsicherheiten erschweren die Bewertung
- In der Praxis haben sich zahlreiche Methoden etabliert, um Investitionen unter Unsicherheit zu bewerten

Credits

- Brynjolfsson, E. (1993): The productivity paradox of information technology. In: Communications of the ACM, Jg.36, H.12: S.67-77
- Carr, Nicholas G. (2003): IT Doesn't Matter. In: Harvard Business Review, 1. Mai, Nr. 2, S. 41–49
- Wan, Zeying; Fang, Yulin; and Wade, Michael, "A Ten-Year Odyssey of the 'IS Productivity Paradox' - A Citation Analysis (1996-2006)" (2007). AMCIS 2007 Proceedings. Paper 437. online unter: <http://aisel.aisnet.org/amcis2007/437>, zuletzt geprüft am 15.05.2013

8.3 Skript: Wert von Informationen

ABRAHAM BERNSTEIN, ESTHER KAUFMANN, CHRISTOPH KIEFER,
DANIEL SPICAR

8.3.1 Einführung

Entscheidungen machen einen immer grösseren Teil unserer täglichen Arbeit aus. Um diese Entscheidungen zu treffen, benötigen wir Informationen. Aus diesem Grund kommt dem Sammeln von Informationen eine zentrale Bedeutung zu, da es informierte Entscheidungen ermöglicht. Demzufolge kann man den Wert von Information unter anderem dadurch bestimmen, dass man die Gewinne (Ersparnisse) misst, welche durch Entscheidungen mit einer zusätzlichen Information erzielt werden.

Was hat das Ganze nun mit Informatik zu tun? Informationssysteme werden im Wesentlichen zur Informationsbeschaffung, Informationsverarbeitung und Informationsverwaltung eingesetzt. Es liegt deshalb auf der Hand, den Wert von Informationssystemen (also Informatikanwendungen) zu messen, indem man den durch das System entstehenden Mehrwert mit dem durch die verbesserte Entscheidungsfindung entstehenden Mehrwert gleichsetzt.

Diese kleine Einführung stellt nun ein Verfahren vor, mit welchem der Wert von Information (und damit von Informationssystemen) durch deren Einfluss auf Entscheidungen gemessen werden kann. Es stellt zunächst ein kleines Entscheidungsproblem vor, welches anschliessend beispielhaft gelöst wird. Es schliesst mit einer theoretischen Zusammenfassung, die bei der weiteren Anwendung helfen kann.

8.3.2 Musikproduktion: To Studio or not to Studio?

Ein guter Freund von Ihnen, der als Musiker am Anfang seiner Karriere steht, bittet Sie um Ihren Rat. Er hat verschiedene Songs geschrieben und möchte nun jährlich ein neues Album produzieren (Songs aufnehmen, CDs erstellen usw.). Dabei hat er für jedes Album zwei Varianten:

Variante 1: Er könnte einerseits die Songs in seiner Garage aufnehmen und das Zusammenstellen sowie Kopieren der CDs zuhause am eigenen PC machen. Die Produktionskosten würden sich auf Fr. 10.- pro CD belaufen. So könnte er maximal 2'000 Stück produzieren.

Variante 2: Ein Tonstudio hat ihm eine Offerte gemacht, die alle Arbeitsschritte, welche zum Produzieren der CDs nötig sind, beinhaltet. Es entstehen Kosten von total Fr. 100'000.-, unabhängig von der abgesetzten Menge. Mit dieser Variante könnten maximal 10'000 CDs produziert werden.

Grundsätzliche Rahmenbedingungen: Die Absatzmenge ist natürlich abhängig vom Erfolg des Musikers. Landet er einen Top-Ten-Hit, so wird er 10'000 CDs verkaufen. Ist dies nicht der Fall, so kann er an Freunde, Bekannte und Verwandte höchstens 1'000 CDs verkaufen. Die Wahrscheinlichkeit für einen Top-Ten-Hit beträgt 30%. Der Verkaufspreis einer CD beträgt in beiden Fällen Fr. 30.-.

Frage: Soll er nun im Studio oder in seiner Garage produzieren?

8.3.3 Berechnungen: Eine Erwartungswertrechnung

Um diese Frage zu beantworten, braucht es Statistik und die darin enthaltene Wahrscheinlichkeitsrechnung. Genauer gesagt, wird der Erwartungswert einer diskreten Zufallsvariablen benötigt.

”Im diskreten Fall errechnet sich der Erwartungswert als die Summe der Produkte aus den Wahrscheinlichkeiten jedes möglichen Ergebnisses des Experiments und den ”Werten” dieser Ergebnisse. Ist X eine diskrete Zufallsvariable, die die Werte x_1, x_2, \dots mit den jeweiligen Wahrscheinlichkeiten p_1, p_2, \dots annimmt, errechnet sich der Erwartungswert $E(X)$ “² zu:

$$E(X) = \sum_i x_i p_i = \sum_i x_i P(X = x_i)$$

Das Problem Ihres Freundes lässt sich nun ebenfalls als Erwartungswertrechnung formulieren:

$$E(Vorkommnis) = \sum_{x_i \in \{Hit, Flop\}} x_i \times P(Vorkommnis = x_i) \quad (1)$$

Dabei bezeichnen x_i den Gewinn (oder Verlust) des Vorkommnisses “Hit” (bzw. “Flop”) und $P(Vorkommnis = x_i)$ deren Wahrscheinlichkeiten.

- Variante 1:

²Wikipedia: <http://de.wikipedia.org/wiki/Erwartungswert>

- Kosten der CD-Produktion in der **Garage** (pro Stück):

$$Kosten_G^{CD} = 10.-$$

- Maximal mögliche Produktion in der **Garage**:

$$Absatz_G^{Hit} = 2'000 \text{ Stück}$$

- Variante 2:

- Kosten der CD-Produktion im **Tonstudio** (total):

$$Kosten_{TS} = 100'000.-$$

- Maximaler Absatz bei einem Hit: $Absatz_{TS}^{Hit} = 10'000 \text{ Stück}$

- Allgemeine Bedingungen:

- Wahrscheinlichkeit Hit: $p(Hit) = 0.3$

- Wahrscheinlichkeit Flop: $p(Flop) = 1 - p(Hit) = 0.7$

- Maximaler Absatz bei Flop (egal ob Garage oder Tonstudio):

$$Absatz^{Flop} = 1'000 \text{ Stück}$$

- Verkaufspreis CD: $Preis^{CD} = 30.-$

Demnach ist der Erwartungswert bei der Produktion **in der Garage**:

$$\begin{aligned} E_G &= p(Hit) \times [Absatz_G^{Hit} \times (Preis^{CD} - Kosten_G^{CD})] \\ &\quad + p(Flop) \times [Absatz^{Flop} \times (Preis^{CD} - Kosten_G^{CD})] \\ &= 0.3 \times [2'000 \times (30 - 10)] + 0.7 \times [1'000 \times (30 - 10)] \\ &= \underline{\underline{26'000}} \end{aligned}$$

Der Erwartungswert bei der Produktion **im Tonstudio** ist:

$$\begin{aligned} E_{TS} &= p(Hit) \times [(Absatz_{TS}^{Hit} \times Preis^{CD}) - Kosten_{TS}] \\ &\quad + p(Flop) \times [(Absatz^{Flop} \times Preis^{CD}) - Kosten_{TS}] \\ &= 0.3 \times [(10'000 \times 30) - 100'000] + 0.7 \times [(1'000 \times 30) - 100'000] \\ &= \underline{\underline{11'000}} \end{aligned}$$

Dadurch ergibt sich, dass Ihr Freund **in der Garage** produzieren sollte. Das heisst, der Erwartungswert **OHNE** weitere Information ist:

$$E_{\text{ohne_Information}} = E_G = \underline{\underline{26'000}}$$

8.3.4 Wert von perfekter Information

Nun hat Ihr Freund eine gute Freundin, die ein Programm geschrieben hat, welches voraussagen kann, ob ein Musikstück ein Hit oder ein Flop wird. Wie viel sollte Ihr Freund bereit sein, für dieses Programm zu bezahlen?

Der naive Ansatz wäre, zu behaupten, dass der Wert des Programmes (oder der Information, ob ein Stück ein Hit oder ein Flop wird) die Differenz zwischen dem Erwartungswert in der Garage und demjenigen im Tonstudio ist (also $E_G - E_{TS}$). Dieser Ansatz wäre jedoch falsch! Denn das Programm sagt nur, **ob** ein Stück ein Hit oder ein Flop wird; es kann einen Flop nicht in einen Hit umwandeln!

Schauen wir uns mal den Entscheidungsprozess an: Wenn das Programm vorhersagt, dass das Musikstück ein Hit wird, dann geht Ihr Freund ins Studio. Wenn das Programm einen Flop vorhersagt, dann produziert er in der Garage. Wenn das Programm den Hit perfekt vorhersagen kann, dann tut es dies in $p(Hit) = 30\%$ Fällen. Graphisch kann die Entscheidung also wie in Abbildung 1 dargestellt werden.

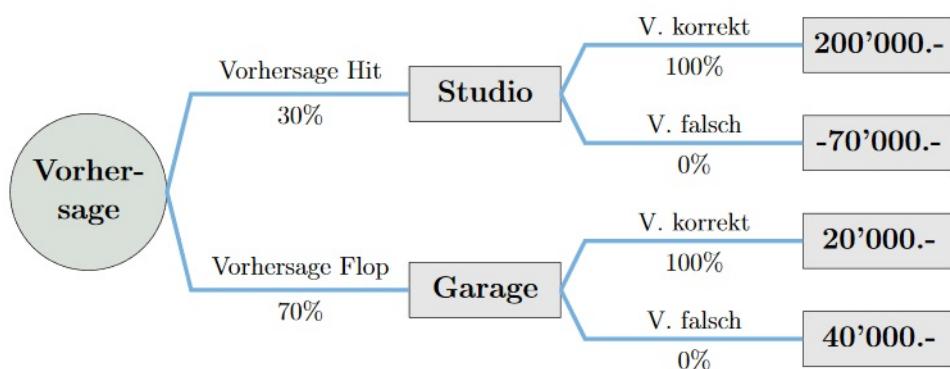


Abbildung 1: Entscheidungsbaum perfekte Information: Gewinne und Verluste. Weil das Programm perfekt arbeitet, entsprechen die Vorhersagewahrscheinlichkeiten den Wahrscheinlichkeiten für $p(Hit)$ und $p(Flop)$ und nach der Vorhersage ist jeweils nur ein Ast relevant.

Konsequenterweise ist der Erwartungswert mit perfekter Information:

$$\begin{aligned}
 E_{\text{perfekte_Information}} &= p(\text{Hit}) \times [(Absatz_{TS}^{\text{Hit}} \times Preis^{CD}) - Kosten_{TS}] \\
 &\quad + p(\text{Flop}) \times [Absatz^{\text{Flop}} \times (Preis^{CD} - Kosten_G^{CD})] \\
 &= 0.3 \times [(10'000 \times 30) - 100'000] \\
 &\quad + 0.7 \times [1'000 \times (30 - 10)] \\
 &= \underline{\underline{74'000}}
 \end{aligned}$$

Und damit der Wert der perfekten Information:

$$\begin{aligned}
 Wert_{\text{perfekte_Information}} &= E_{\text{perfekte_Information}} - E_{\text{ohne_Information}} \\
 &= 74'000 - 26'000 = \underline{\underline{48'000}}
 \end{aligned}$$

Demzufolge sollte Ihr Freund bereit sein, $\leq 48'000$ für das Programm zu bezahlen.

9 Digitale Güter in der Volkswirtschaft

PROF. ABRAHAM BERNSTEIN

9.1 Eigenschaften und Besonderheiten digitaler Güter



Institut für Informatik

Eigenschaften und Besonderheiten digitaler Güter

Prof. Abraham Bernstein, Ph.D.

Eigenschaften und Besonderheiten digitaler Güter

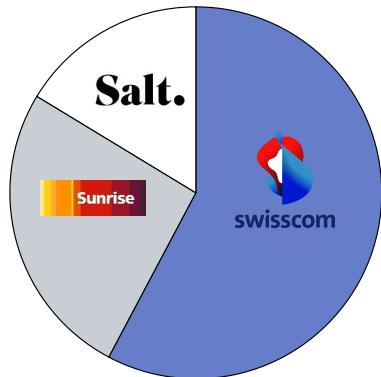
- **Einführung**
- Tendenz zu öffentlichen Gütern
- Tendenz zu natürlichen Monopolen
- Asymmetrische Informationen
- Netzwerkeffekte

Wieso muss man für bestimmte Software nichts bezahlen?



Weshalb beherrscht Swisscom weiterhin den Mobilmarkt?

Swisscom ist Nr. 1 im Schweizer Mobilmarkt obwohl Mitbewerber gleiche Dienstleistungen günstiger anbieten.



Marktanteile Schweizer Mobilfunkmarkt 2016

Warum verlangt iTunes für denselben Song unterschiedliche Preise?

Land	Preis	USD	Markup
Mexiko	15 pesos	1.24	-4%
USA	\$ 1.29	1.29	
UK	£ 0.99	1.56	21%
Eurozone	€ 1.29	1.72	33%
Japan	¥ 150	1.81	41%
Schweiz	SFr. 2.20	2.30	78%



Quelle: <http://www.macstories.net/stories/the-great-disparity-in-global-itunes-prices/> (Januar 2011)

Der Grund: Bei digitalen Gütern können die Marktkräfte ihre Wirkung oft nicht voll entfalten

- Verschiedene Eigenschaften digitaler Güter erschweren optimale Allokation der Ressourcen.
- Diese Eigenschaften können folgenden ökonomischen Erklärungsmodellen zugeordnet werden:
 - Öffentliche Güter
 - Natürliches Monopol
 - Asymmetrische Informationen
 - Externe Effekte

Eigenschaften und Besonderheiten digitaler Güter

- Einführung
- **Tendenz zu öffentlichen Gütern**
- Tendenz zu natürlichen Monopolen
- Asymmetrische Informationen
- Netzwerkeffekte

Konkurrenten profitieren von Forschung & Entwicklung und fabrizieren Nachahmerprodukte

"I'm going to destroy Android, because it's a stolen product. I'm willing to go thermonuclear war on this."

(Steve Jobs)



Aber:



Braun vs. Apple

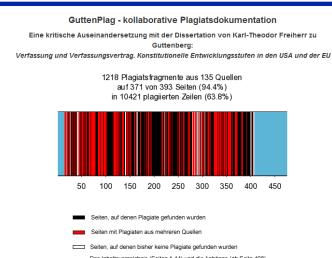
Braun vs. Apple

SBB vs. Apple

Konkurrenten profitieren von Forschung & Entwicklung und fabrizieren Nachahmerprodukte

"If you copy from one author, it's plagiarism.
If you copy from two, it's research."

(Wilson Mizner)



Siehe auch: Annalee Saxenian, Regional Advantage: Culture and Competition in Silicon Valley and Route 128, Harvard University Press, 1994

Digitale Güter weisen Eigenschaften von öffentlichen Gütern auf

Zwei Eigenschaften machen gemäss ökonomischer Theorie ein öffentliches Gut aus:

- Versagen des Ausschlussprinzips
- Nichtrivalität in der Verwendung (→ Konsumation zerstört das Gut nicht)

Folgen:

- Trittbrettfahrer profitieren gratis von Leistungen
- Eigentumsrechte kaum durchsetzbar
- Kein Anreiz zur Herstellung von öffentlichen Gütern
- Es kommt kein Markt zustande

Eigenschaften und Besonderheiten digitaler Güter

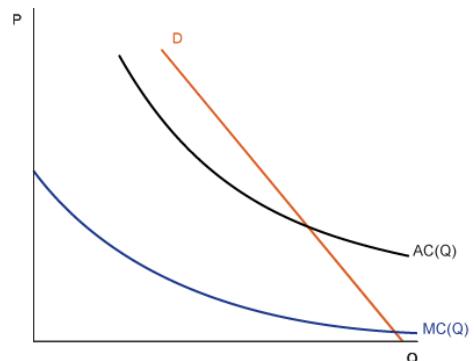
- Einführung
- Tendenz zu öffentlichen Gütern
- **Tendenz zu natürlichen Monopolen**
- Asymmetrische Informationen
- Netzwerkeffekte

Kostenstruktur von digitalen Gütern fördert natürliche Monopole

- Digitale Güter haben i.d.R. hohe Entwicklungskosten
 - Hohe Kosten für die Generierung des ersten Exemplars
- Reproduktionskosten vernachlässigbar tief
 - Einfach kopierbar
 - Keine Kapazitätsbeschränkungen bei der Produktion
- Distribution einfach und kostengünstig

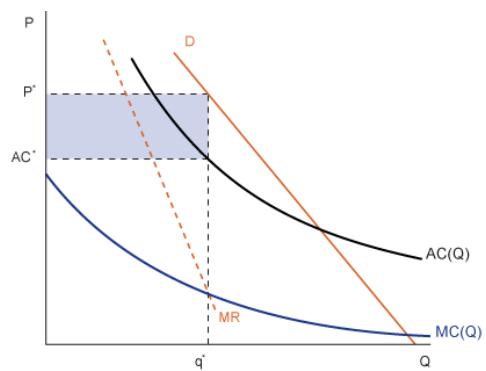
Hohe Fixkosten zusammen mit vernachlässigbaren Grenzkosten führen zu steigenden Skalenerträgen

- **Skalenerträge:** Verdopplung des Inputs führt zu überproportionaler Vergrößerung des Outputs.
→ Durchschnittskosten sinken laufend.
- Dank sinkender Durchschnittskosten kann die betroffene Firma jede beliebige Gütermenge günstiger produzieren als mehrere Unternehmen gemeinsam.



Dauerhaft sinkende Durchschnittskosten führen zu natürlichen Monopolen

- Das effizienteste Unternehmen übernimmt oder verdrängt Konkurrenten.
- Überlebendes Unternehmen wird zum Monopolisten.



Eigenschaften und Besonderheiten digitaler Güter

- Einführung
- Tendenz zu öffentlichen Gütern
- Tendenz zu natürlichen Monopolen
- **Asymmetrische Informationen**
- Netzwerkeffekte

Der Nutzen vieler Produkte kann erst nach dem Konsum beurteilt werden

“Whether we’re buying a pair of jeans, ordering a cup of coffee or selecting a long-distance carrier - everyday decisions have become increasingly complex .“

(Barry Schwartz - *The Paradox of Choice*)



Das Problem der asymmetrischen Informationen: Digitale Güter sind Erfahrungsgüter

Wissen zwischen Anbietern und Nachfragern ist bei digitalen Gütern ungleich verteilt:

- *Ex ante (vor Entscheid)*
 - **Defizit beim Nachfrager:** Potentieller Käufer kann Nutzen von digitalen Gütern erst nach dem Kauf beurteilen
 - **Defizit beim Anbieter:** Potentieller Investor kann Qualität bahnbrechender digitaler Güter schlechter beurteilen

Das Problem der asymmetrischen Informationen: Digitale Güter sind Erfahrungsgüter

Wissen zwischen Anbietern und Nachfragern ist bei digitalen Gütern ungleich verteilt:

- *Ex post (nach Entscheid)*
 - **Defizit beim Nachfrager:** Kunde weiss nicht, ob der Anbieter das Produkt weiter entwickelt/unterstützt.
 - **Defizit beim Anbieter:** Bank kann das von ihr finanzierte Startup im laufenden Geschäftsbetrieb nicht beobachten.

Die Informationssuche kostet Mühe, Zeit und Geld

- Informationsbeschaffung ist dann rational, wenn der daraus gewonnene Nutzen den Aufwand übersteigt.
- Suche lohnt sich solange, bis der Grenznutzen der Informationsbeschaffung gleich dem Grenznutzen der Information ist.



Eigenschaften und Besonderheiten digitaler Güter

- Einführung
- Tendenz zu öffentlichen Gütern
- Tendenz zu natürlichen Monopolen
- Asymmetrische Informationen
- Netzwerkeffekte

„The Winner takes it all“-Prinzip

“When two or more firms compete for a market where there is strong positive feedback, only one may emerge as a winner. It's unlikely that all will survive”

(Carl Shapiro,
Professor für Unternehmensstrategie, UC Berkeley)



Digitale Güter profitieren von Netzwerkeffekten

Positive Externalitäten:

- Je mehr Personen das digitale Gut verwenden, desto mehr Nutzen habe ich und jeder andere.
 - Austausch von Daten und Anwendungs-Know-how
 - Schnellere Koordination
 - Raschere Produktverbesserungen
 - Mehr Supportdienstleistungen
 - Komplementäre Güter und Dienstleistungen

Netzwerkeffekte bergen auch Risiken

Negative Externalitäten:

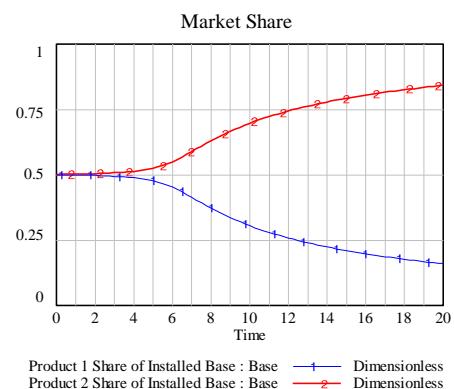
- Attraktives Ziel für Hacker, Viren, Trojaner
- Netzwerkauslastung
- Abhängigkeiten
- Cybermobbing, „Shitstorm“

Netzwerkeffekte finden sich auf vielen Ebenen

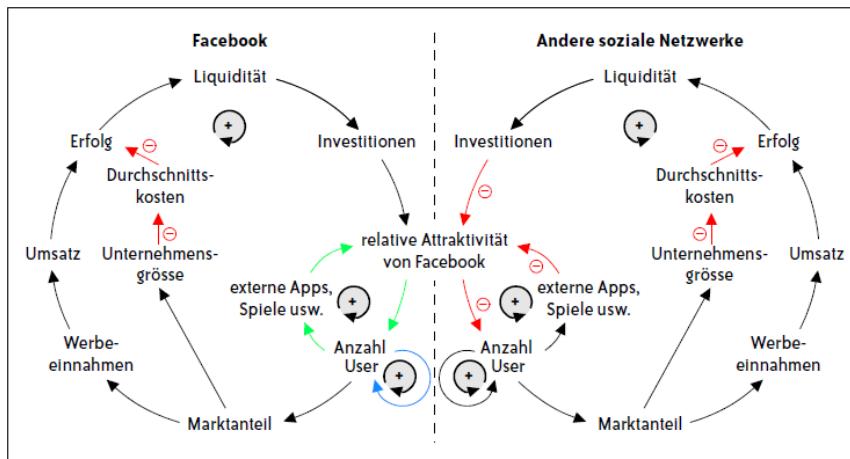
- Kommunikationstechnologie (Email, Skype, SMS)
- Anwendungssoftware (MS Office, Acrobat)
- Datenformate (MP3, ogg, jpg, mov, odt)
- Communities (Facebook, Ebay, studiVZ)
- Komplementäre Produkte und Dienstleistungen (Abspielgeräte, Entwicklerwerkzeuge, physische Netzwerke)

Märkte mit Netzwerkeffekten tendieren zu Monopolen

- First-Mover-Advantage: Frühe Gewinner dominieren den Markt
- Der Markt-Leader erzielt hohe Gewinne
- Selbst mit einem überdurchschnittlichen Produkt ist es schwierig, in einem bestehenden Markt Fuss zu fassen
- Oft gewinnt die zweitbeste Technologie



Facebook: „The winner takes it all“ dank Netzwerkeffekten



Quelle: Roland Waibel/Daniel Beyeler, Vernetztes Denken in BWL und VWL

Fazit: Die Eigenschaften digitaler Güter führen zu suboptimalen Marktergebnissen

- Intransparenz
- Suchkosten
- Verdrängungswettbewerb
- „Ungerechter“ Monopolgewinn
- Wohlfahrtsverluste

Credits

- Paul Butler for Facebook Engineering
- janosch500.deviantart.com
- www.Guttenplag.com
- www.gizmodo.com
- Frankfurter Allgemeine
- ZDNet
- Handelszeitung

9.2 Massnahmen gegen Marktverzerrungen



Institut für Informatik

Massnahmen gegen Marktverzerrungen

Prof. Abraham Bernstein, Ph.D.

Massnahmen gegen Marktverzerrungen

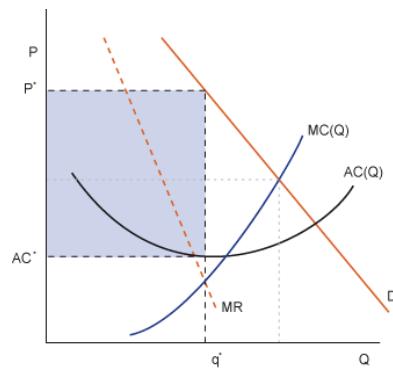
- Massnahmen gegen Marktverzerrung

Rückblick: Verschiedene Eigenschaften der digitalen Güter führen zu verzerrten Marktergebnissen

- | | |
|--|---|
| <ul style="list-style-type: none">■ Eigenschaften<ul style="list-style-type: none">– Öffentliche Güter– Natürliches Monopol– Asymmetrische Informationen– Externe Effekte | <ul style="list-style-type: none">■ Marktergebnisse<ul style="list-style-type: none">– Intransparenz– Suchkosten– Verdrängungswettbewerb– „Ungerechter“ Monopolgewinn– Wohlfahrtsverluste |
|--|---|

Anreize sollen Anbieter zu innovativem Verhalten motivieren

- Schaffung von Verfügungsrechten
 - Patente (Monopol auf Zeit)
 - Urheberrecht
- Serviceorientierte Geschäftsmodelle (Mehrwert durch Dienstleistungen und Funktionalität)



Gewinnmaximierung des Monopolisten:
Preis höher, Menge tiefer als bei vollkommenem Wettbewerb

Kosten sollen destruktives Verhalten der Nachfrager verhindern

- Glaubwürdige Strafandrohungen und drakonische Strafen
- Technologische Lösungen, um Kopieren zu erschweren
 - Kopierschutz
 - Verschlüsselung
 - Digital-Rights-Management-Systeme
 - Software-as-a-Service-Lösungen
 - Koppelung an physisches Gut
- Versionsbildung

Streit um Urheberrechte
Gema klagt gegen YouTube
28.01.2013 | Der Streit zwischen der Internet-Videoplattform YouTube und der Gema (Gesellschaft für Media and Entertainment) wegen eines Hinweises auf die Gema hat die Verurteilung erzielt. Nachdem die Richter einen Urheberrechtsverletzungshinweis von YouTube abgewiehlt, ZDNet.de IT-Business

FAZ, 28.01.2013

Oracle geht im Java-Prozess gegen Google in Berufung
D v. Google
Landgericht von Stephan Beuerlein am 14. Februar 2013, 090-01 Ute
Unterlassen
Todtberger
Oracle wird trotz der Niederlage in der ersten Instanz den Java-Rechtsstreit mit Google berichten. Am 14. Februar 2013 hat das US Court of Appeals eine Berufungserhebung abgelehnt. Grund ist, dass Googles Nutzung von Java in seinem Multimediasystem Android "vorsätzlich und fair" war. Das bestätigt der Richter.

ZDNet.de, 14.02.2013

Beitrag: Demanda und Apple-Klage: Millionen für 15 Sekunden
D v. Google
Urteil gefordert
YouTube - Dass Google die Bilder von dem Video nicht verschlüsselt hat, kann sich die Gesellschaft nicht erklären.
Der Test
Eine Schweizer Fotografin ging gegen Apple erfolgreich wegen Urheberrechts-Vorlagerungen vor. Apple zahlte Millionen, behauptete Apple (Apple) VON CHRISTIAN BURGIERER 18.03.2013

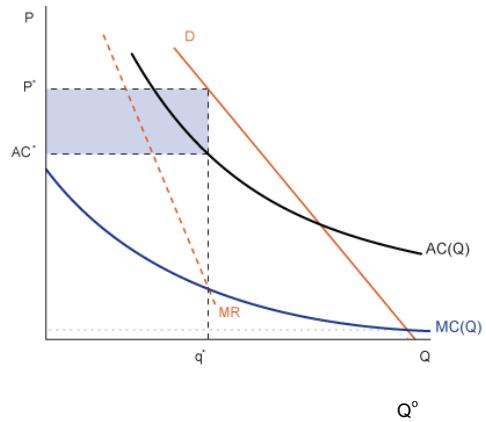
HANDELSZEITUNG, 18.03.2013

Erhöhung der Transparenz kann Suchkosten senken

- IT kann Kosten der Informationssuche und -verarbeitung senken:
 - Information kann dank *Suchmaschinen* leichter gefunden werden
 - Aktualisierung schneller möglich
 - Datenquellen können schneller nach relevanten Informationen durchsucht werden
 - Dank *Bewertungstools* kann die Qualität der Information alternativ beurteilt werden
 - *Visualisierung* macht Interpretation einfacher

Staatliche Eingriffe weisen unerwünschte Monopole in die Schranken

- Verstaatlichung
- Regulierung
- Versteigerung des Monopols
- Sicherstellen von Marktzutritt und Substitutionskonkurrenz



Unsicherheiten können auf verschiedene Arten ausgeräumt werden

- **Screening:** Die *uninformierte Seite* erweitert ihren Informationsstand, indem sie
 - aktiv nach Informationen sucht
 - dem Gegenüber Selbstselektion ermöglicht (z.B. Franchise)
 - Dritte einschaltet, um Informationen zu sammeln
- **Signaling:** Die *besser informierte Seite* verpflichtet sich zur Übernahme von Kosten, z.B.:
 - Aufbau eines Servicenetzes
 - Zertifizierung
 - Gewährleistung spezieller Garantien
 - Akzeptanz von Konventionalstrafen
- Aufbau einer Reputation

Fazit

- Marktgerechte Lösungen sowie staatliche Massnahmen können helfen, Marktineffizienzen zu beseitigen.
- Dazu gehören Anreize und Transparenz schaffen und staatliche Regulierungen und Unsicherheiten abbauen.

9.3 Wettbewerbsstrategien I - Einleitung



Institut für Informatik

Wettbewerbsstrategien I – Einleitung

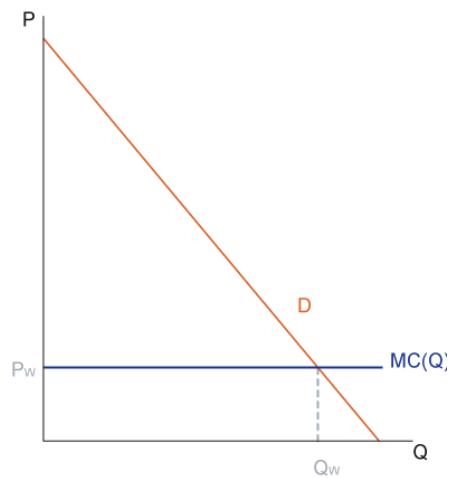
Prof. Abraham Bernstein, Ph.D.

Wettbewerbsstrategien I – Einleitung

- Einleitung

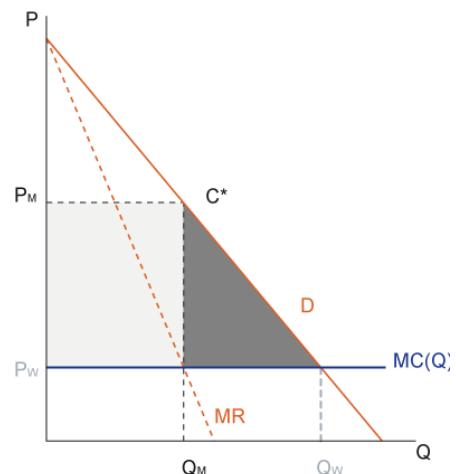
Rückblick VWL: Modellannahmen „Vollkommener Wettbewerb“

- Markteintritt bzw. –austritt jederzeit möglich
- Homogene Güter (vollkommene Substitute)
- Marktpreis exogen (von aussen gegeben)
- Vollständige Information (keine Suchkosten)



Rückblick: Merkmale der Internet-Ökonomie begünstigen das Entstehen von Monopolen

- Ein Unternehmen dominiert den gesamten Markt
- Monopolist kann Preis beliebig gestalten
- Teil der Konsumentenrente wandert zum Monopolisten
- Wohlfahrtsverluste
 - Gewinnzuwachs des Monopolisten ist geringer als Verluste der Konsumenten



Marktrealitäten weichen von den Modellvorstellungen ab

- Es existieren *Marktunvollkommenheiten*:
 - Asymmetrische Informationen
 - Transaktionskosten
 - Unterschiedliche Präferenzen (z.B. für bestimmte Marken)
- Unternehmen nutzen diese Marktunvollkommenheiten, um ihre Position zu verbessern.

Unternehmen nutzen verschiedene Strategien, um ihre Marktmacht auszubauen oder zu verteidigen

- Absprachen (z.B. Preiskartell)
- Übernahmen (z.B. durch Aktienkauf)
- Preisdifferenzierung (1)
- Produktdifferenzierung (2)
- Produktbündelung (3)
- Innovation (z.B. Alleinstellung durch innovative Produkte)



(1)



(2)



(3)

Fazit

- Der Wettbewerb ist ein Prozess, bei dem Unternehmen Marktmacht erlangen und wieder verlieren.
- Unternehmen versuchen, diesen Prozess zu ihren Gunsten zu beeinflussen.
- Dabei werden Preis- und Produktstrategien genutzt, um die eigene Position zu stärken und Konkurrenten zu behindern.

9.4 Wettbewerbsstrategien II - Preisdifferenzierung



Institut für Informatik

Wettbewerbsstrategien II – Preisdifferenzierung

Prof. Abraham Bernstein, Ph.D.

Wettbewerbsstrategien II – Preisdifferenzierung

■ Preisdifferenzierung

Preisdifferenzierung

- Definition:
 - Das gleiche Produkt wird zu unterschiedlichen Preisen an verschiedene Nachfrager verkauft.
- Ziel:
 - Gewinnmaximierung durch Abschöpfen der Konsumentenrente
 - Marktdurchdringung

MITTWOCH, 3. APRIL 2013 / WWW.20MINUTENCH

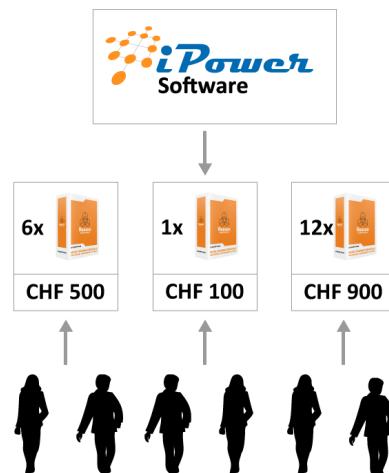


Voraussetzungen

- Produzent hat Marktmacht
- Konsumenten haben unterschiedliche Präferenzen
- Weiterverkauf zwischen Konsumenten ist (zu vertretbaren Kosten) nicht möglich

Variante 1: Selbstselektion

- Keine Kenntnisse über Zahlungsbereitschaften der Nachfrager
- Der Preis variiert in Abhängigkeit der Abnahmemenge
- Gleiche Preisstruktur für alle (keine Diskriminierung)
 - Risiko der Kannibalisierung



Beispiele Selbstselektion

The left screenshot shows a website for CYON. It features sections for Webhosting, Cloudserver, and Domains. Under Webhosting, there are three plans: Single (25 GB, 8.90 CHF/Monat), Double (50 GB, 16.90 CHF/Monat), and Triple (100 GB, 24.90 CHF/Monat). A button for 'Jetzt Bestellen!' (Buy Now) is visible. The right screenshot shows a software licensing calculator for Adobe Photoshop Extended. It lists one item with a quantity of 500 and a subtotal of 500'000. To the right, a 'Discount Visualizer' chart shows four levels: CLP 4 (yellow), CLP 3 (blue), CLP 2 (green), and CLP 1 (orange). A note indicates that 500'000 points are needed to reach CLP 4.

Variante 2: Marktsegmentierung

- Kenntnisse über Zahlungsbereitschaften begrenzt
- Einteilung aufgrund beobachtbarer Merkmale:
 - Person (Alter, Geschlecht, Ausbildung)
 - Region
 - Zeit
- Angebotspreis gilt nur bei Gruppenzugehörigkeit (Diskriminierung)

The diagram illustrates market segmentation pricing. At the top is the logo for 'iPower Software'. An arrow points downwards to three separate boxes, each containing a product box and a price: '1x CHF 49', '1x CHF 100', and '1x CHF 399'. Arrows then point from these boxes to stylized human figures below, representing different customer segments.

Beispiele Marktsegmentierung

The screenshot shows two main sections. On the left, the 'Apple Store Bildung' page offers student discounts on Macs, iPads, and iPhones. It includes a recycling program section and a 'Für Studenten' section. On the right, a promotional offer for 'Für alle unter 26' (For all under 26) is shown, comparing various mobile network plans (L, M, S, light) based on usage and features.

Variante 3: Vollständige Preisdifferenzierung

- Produzent kennt individuellen Präferenzen der Nachfrager
 - Freiwillige Angaben
 - Transaktionsgeschichte (Cumulus-Karte)
 - Beobachtung des Online-Verhaltens
- Jeder Kunde bezahlt individuellen Preis gemäss seiner Zahlungsbereitschaft

The diagram shows a central box labeled 'iPower Software' with an arrow pointing down to a box labeled '1x'. From this box, an arrow points down to six silhouettes of people, each associated with a different price: CHF 17, CHF 850, CHF 270, CHF 540, CHF 79, and CHF 125.

Beispiele Vollständige Preisdifferenzierung

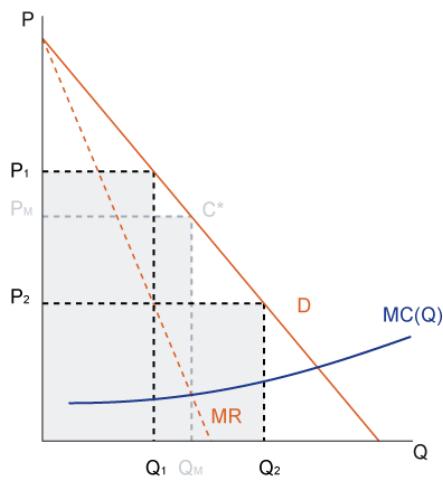
The screenshot shows the priceline.com homepage with a search bar for "Search Over 201,000 Hotels". Below it are fields for "Where are you going?", "Check-in", "Check-out", and "Rooms". Promotional banners include "Express Deals" (Save up to 45% OFF Hotels), "Book on the Go" (Download Priceline Mobile Apps), and "Last Minute Deals" (Book Now and Save up to 50%).

The Greentoe website features a "NAME YOUR PRICE" section. It shows a hand drawing a price on a screen, surrounded by various products like a refrigerator, a guitar, and a camera. Below the image are three steps: 1. You Name Your Price, 2. We Notify Retailers, and 3. You Save Green.

- Name your own price: Kunde gibt seinen Maximalpreis an und erhält Angebote

Fazit

- Preisdifferenzierung ermöglicht Gewinnmaximierung durch Abschöpfen der Konsumentenrente
- Im Vergleich zum Monopol wohlfahrtssteigernd
 - Falls Märkte erschlossen werden, die ohne Preisdifferenzierung nicht beliefert würden
- Bei perfekter Preisdiskriminierung:
 - Unternehmen schöpft gesamte Konsumentenrente ab
 - Keine Effizienzverluste



Credits

- 20min.ch

9.5 Wettbewerbsstrategien III - Produktdifferenzierung



Institut für Informatik

Wettbewerbsstrategien III – Produktdifferenzierung

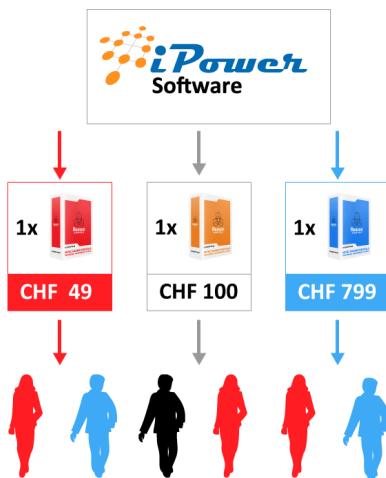
Prof. Abraham Bernstein, Ph.D.

Wettbewerbsstrategien III – Produktdifferenzierung

- **Produktdifferenzierung**

Produktdifferenzierung

- Definition:
 - Eigenschaften eines bestehenden Produktes werden *modifiziert* (\neq Innovation)
- Ziel:
 - USP (Unique Selling Proposition)
 - Erobern neuer Märkte
 - Höhere Gewinne



Voraussetzungen

- Produzent hat Marktmacht
- Konsumenten haben unterschiedliche Präferenzen
- Produzent kennt Kundenbedürfnisse

Vertikale Produktdifferenzierung: Unterscheidungsmerkmal Qualität

Bloomberg Web vs. Bloomberg Terminal

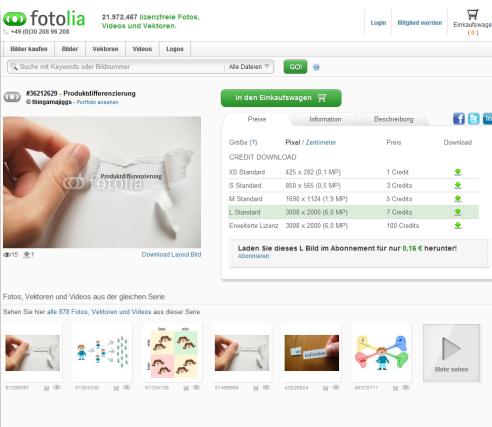


Gratis im Internet

~ CHF 1'500 pro Monat

- Mögliche Differenzierungsmerkmale:
 - Aktualität (real time/verzögert)
 - Geschwindigkeit der Datenübertragung
 - Umfang der Daten
 - Ausfallsicherheit
 - Funktionalität
 - Benutzerfreundlichkeit

Vertikale Produktdifferenzierung: Unterscheidungsmerkmal Qualität



Mögliche Differenzierungsmerkmale:

- Auflösung
- Verwendungszweck

Horizontale Produktdifferenzierung: Unterscheidungsmerkmal Aufmachung

Mögliche Differenzierungsmerkmale:

- Farbe
- Design
- Image



Produktdifferenzierung bewirkt bei Produzenten ...

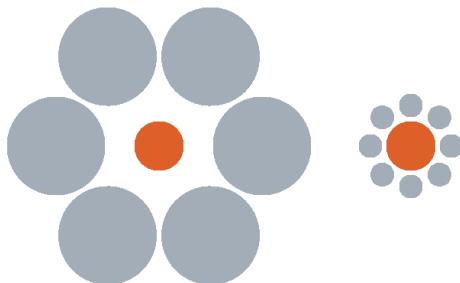
- Anpassungsmöglichkeiten an individuelle Kundenpräferenzen
- Mehr Marktmacht
 - Grössere Kundenbindung
 - Abhebung von Konkurrenzprodukten
 - Geringere Vergleichbarkeit
 - Höhere Marktdurchdringung
 - Anheben der Markteintrittsschranken für Konkurrenz
- Mehr Umsatz
 - Weiterentwicklungen lösen Ersatzkäufe aus (i.d.R. keine Abnutzung)
- Höhere Gewinne

Produktdifferenzierung verursacht bei Konsumenten ...

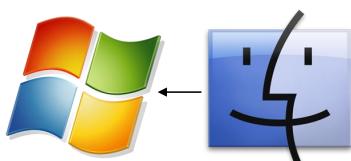
- Mehr Auswahl
- Bedarfsgerechtere Produkte
- Umfassendere Bedürfnisbefriedigung
- Kognitive Verzerrung
- Höhere Kosten
 - Suchkosten um passendes Produkt zu finden
 - Informationskosten um passende Produktvariante auszuwählen
 - Wechselkosten

Kognitive Verzerrung

- Haben Konsumenten nur die Wahl zwischen zwei Alternativen, wählen sie i.d.R. die günstigere
- Einführung eines teuren Spitzenmodells hat unerwarteten Effekt:
 - Statt Marktanteile zu verlieren, wird das mittlere Produkt vermehrt konsumiert
- Durch Einführung eines Spitzenmodells lässt sich der Absatz des mittleren Produkts gezielt steigern



Wechselkosten entstehen durch ...



Z.B. Markenwechsel von Mac zu PC:

- Neue Peripheriegeräte kaufen
- Neue Software kaufen
- Alle Mac-Files zu PC-kompatiblen Files konvertieren
- Das neue Betriebssystem kennenlernen, sich mit der neuen Benutzer-ID vertraut machen, Funktionen der Maus kennenlernen: Was passiert beim Doppelklick, beim Klick rechts...
- Neue „Support Gruppe“ finden

- Technische Inkompatibilitäten
- Produktspezifischer Lernaufwand, Umgewöhnung
- Netzwerkeffekte
- Vertragliche Verpflichtungen
- Langlebige Anschaffungen
- Loyalitätsprogramme
- Produktespezifische Datenformate

Neue Kunden können auch beim Unternehmen Kosten verursachen

Wechselkosten führen zu „Lock-in“



- Wechselkosten ermöglichen Unternehmen Preissetzungsspielraum.
- Der Preis kann über demjenigen der Konkurrenz liegen, ohne dass Kunden abwandern.
- Der Lock-in verursacht, dass Kunden trotz schlechterer Qualität beim aktuellen Anbieter bleiben.

Wechselkosten führen zu „Lock-in“: Grundidee

- Vom Kunden getragene Kosten, z.B.
 - Investition in Zubehör
 - Lernaufwand
 - Etablierung eines neuen Supportnetzwerkes
- Vorteile für Kunden, z.B.
 - Preisvorteile
 - Qualitätsvorteile
- Vom neuen Anbieter getragene Kosten, z.B.
 - Administrationsaufwand

Wechselkosten: Formeln

Totale Wechselkosten $=$

vom Kunden getragene Kosten +
vom neuen Anbieter getragene Kosten

Preissetzungsspielraum des aktuellen Anbieters $=$

Totale Wechselkosten -
Qualitäts- & Preisvorteil nach Wechsel

Wechselkosten führen zu „Lock-in“: Rechenbeispiel



Wechsel von Swisscom zur Sunrise

■ Vom Kunden getragene Kosten:

- Information
- Neue SIM-Karte
- Total: 50 CHF

■ Vorteile für Kunden (über Vertragslaufzeit):

- Günstigere Konditionen
- Besseres Netz
- Total: 70 CHF

■ Vom neuen Anbieter getragene Kosten

- Administrationsaufwand
- Total: 30 CHF

Wechselkosten führen zu „Lock-in“: Rechenbeispiel



Wechsel von Swisscom zur Sunrise

- **Totale Wechselkosten =**
vom Kunden getragene Kosten +
vom neuen Anbieter getragene Kosten
 $= 50 \text{ CHF} + 30 \text{ CHF} = 80 \text{ CHF}$

- **Preissetzungsspielraum des aktuellen Anbieters =**
Totale Wechselkosten -
Qualitäts- & Preisvorteil
 $= 80 \text{ CHF} - 70 \text{ CHF} = 10 \text{ CHF}$

Wenn Kosten des Wechsels höher sind als der daraus entstehende Vorteil, bleibt der Kunde beim alten Anbieter, selbst wenn die Konkurrenz günstiger ist.

Fazit

- Produktdifferenzierung reduziert den Wettbewerbsdruck gegenüber der Konkurrenz
- Produktdifferenzierung verstärkt die Marktmacht des Unternehmens gegenüber den Konsumenten (Lock-in)
- Neueinführung von Premium-Produkten verleiten Low-End-Käufer zum Erwerb höherwertiger Produkte
- Zu viele Varianten erzeugen Such- und Informationskosten
→ Konsumenten verzichten auf Kauf
- Bei digitalen Gütern sind Anpassungen schnell und kostengünstig – sie eignen sich besonders gut für Produktdifferenzierungsstrategien

9.6 Produktdifferenzierung IV - Produktbündelung



Institut für Informatik

Wettbewerbsstrategien IV – Produktbündelung

Prof. Abraham Bernstein, Ph.D.

Wettbewerbsstrategien IV – Produktbündelung

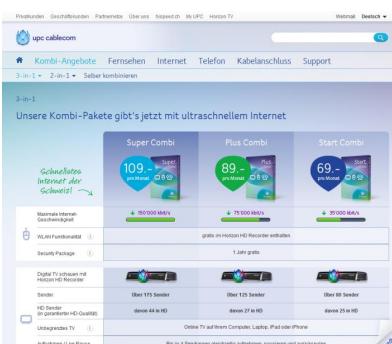
- **Einführung**
- Einzelverkauf vs. reine Bündelung
- Gemischte Bündelung
- Preisstrategie bei positiver Korrelation der Reservationspreise

Produktbündelung

- Definition
 - Zwei oder mehrere Einzelpreise (oder Dienstleistungen) werden zusammengelegt und als Set zu einem Paketpreis angeboten.
- Ziel:
 - Umsatzsteigerung („cross selling“)
 - Gewinnsteigerung durch Abschöpfung der Konsumentenrenten auf den verschiedenen Teilmärkten
 - Optimierung der Kosten (Kapazitätsauslastung)
 - Vermeidung von Preiskämpfen (weniger Preistransparenz)
 - Verhinderung von Markteintritten

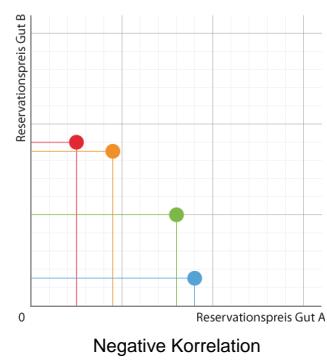
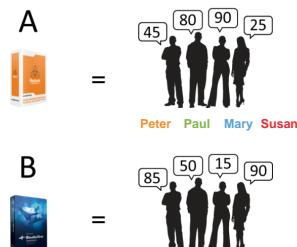
Varianten

- **Reine Bündelung:** Produkte werden ausschliesslich als Set verkauft
- **Gemischte Bündelung:** Produkte sind sowohl einzeln als auch im Bündel erhältlich



Voraussetzungen

- Produzent hat Marktmacht
- Märkte verschiedener Güter werden nicht mehr einzeln betrachtet
- Konsumenten haben unterschiedliche Präferenzen
- Reservationspreise korrelieren negativ



Wettbewerbsstrategien IV – Produktbündelung

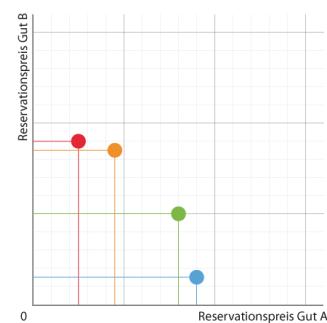
- Einführung
- **Einzelverkauf vs. reine Bündelung**
- Gemischte Bündelung
- Preisstrategie bei positiver Korrelation der Reservationspreise

Einzelverkauf

- Reservationspreis = Zahlungsbereitschaft = maximaler Preis, den ein Individuum für ein Gut oder für eine Dienstleistung zu zahlen bereit ist
- Kein Kauf, wenn Zahlungsbereitschaft (R) < Produktpreis (P)
- Kauf, wenn $R \geq P$

A B Σ

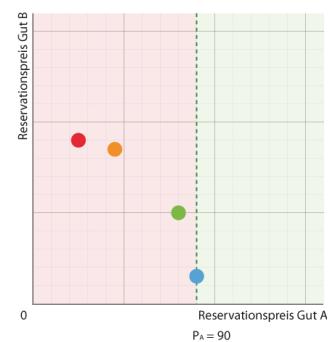
	A	B	Σ
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115



Einzelverkauf

Wenn Produktpreis von A auf 90 gesetzt wird, wird das Produkt nur von Mary gekauft und der Gewinn beträgt 90.

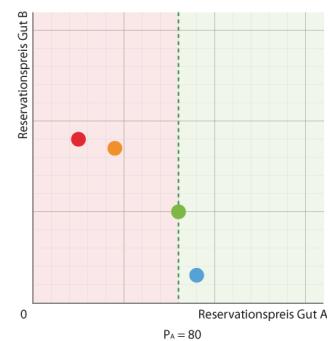
	A	B	Σ
	90		
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	90		



Einzelverkauf

Wenn Produktpreis von A auf 80 gesenkt wird, wird das Produkt zwei mal gekauft und der Gewinn beträgt $2 \times 80 = 160$.

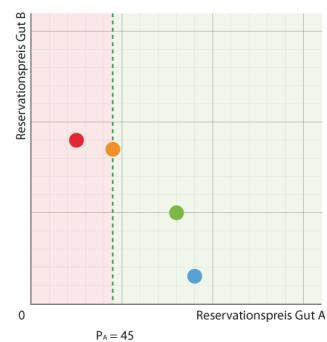
	A	B	Σ
	80		
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160		



Einzelverkauf

Wenn Produktpreis von A auf 45 gesenkt wird, wird das Produkt drei mal gekauft und der Gewinn beträgt $3 \times 45 = 135$.

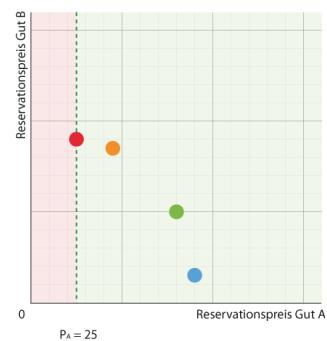
	A	B	Σ
	45		
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	135		



Einzelverkauf

Wenn Produktpreis von A auf 25 gesenkt wird, wird das Produkt von allen vier gekauft und der Gewinn beträgt $4 \times 25 = 100$.

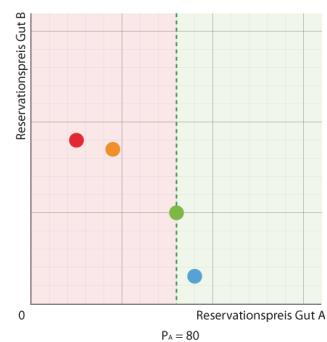
	A	B	Σ
	25		
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	100		



Einzelverkauf

Das optimale Preisniveau für Produkt A liegt also bei 80, denn dann wird der höchste Gewinn erzielt.

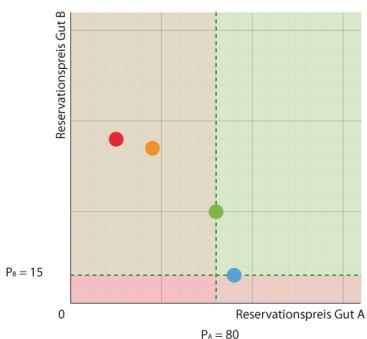
	A	B	Σ
	80		
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160		



Einzelverkauf

Dasselbe nun für Produkt B.
Wenn Produktpreis auf 15 gesetzt wird...

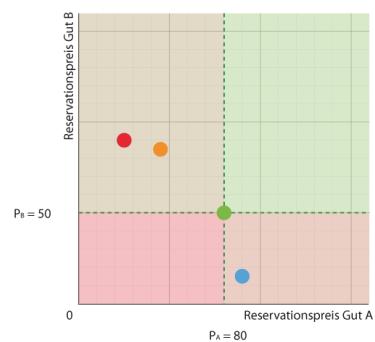
	A	B	Σ
	80	15	
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160	60	



Einzelverkauf

Wenn Produktpreis auf 50 gesetzt wird...

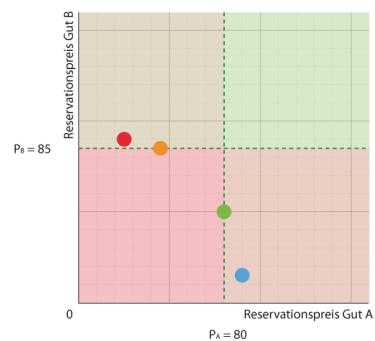
	A	B	Σ
	80	50	
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160	150	



Einzelverkauf

Wenn Produktpreis auf 85 gesetzt wird...

	A	B	Σ
	80	85	
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160	170	

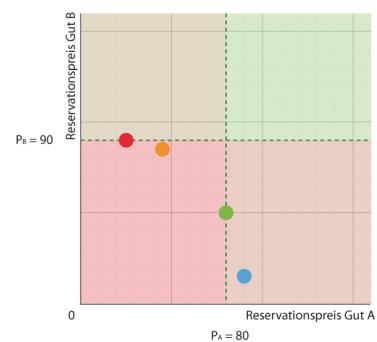


Einzelverkauf

Wenn Produktpreis auf 90 gesetzt wird...

A diagram showing a price vector from point A (80) to point B (90). The vector is labeled with a red starburst containing the number 80 and ends at a red starburst containing the number 90. To the right of the vector is the symbol Σ .

	A	B	Σ
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160	90	

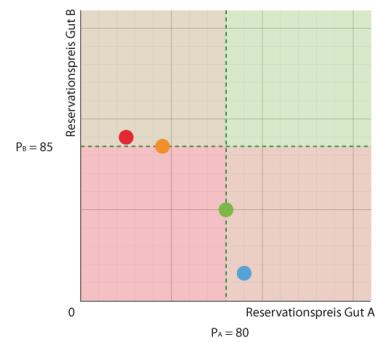


Einzelverkauf

Das optimale Preisniveau für Produkt B liegt also bei 85, denn dann wird der höchste Gewinn erzielt.

A diagram showing a price vector from point A (80) to point B (85). The vector is labeled with a red starburst containing the number 80 and ends at a red starburst containing the number 85. To the right of the vector is the symbol Σ .

	A	B	Σ
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160	170	330

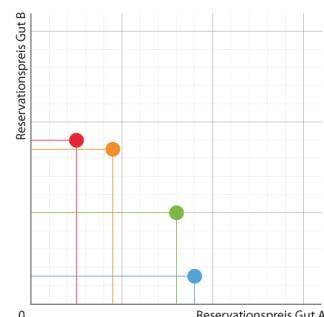


Reine Bündelung

- Güter A und B werden nur im Set (zum Bündelpreis P_{AB}) verkauft
- Konsumentenentscheid reduziert sich auf „nichts kaufen“ bzw. „Bündel kaufen“

Wenn die Produkte A und B einzeln verkauft werden, liegen ihre optimalen Preise bei 80 bzw. 85 was zu einem Profit von 160 bzw. 170 führt, was insgesamt zu einem Profit von 330 führt.

	A	B	Σ
	80	85	
Mary	90	15	105
Paul	80	50	130
Peter	45	85	130
Susan	25	90	115
Gewinn	160	170	330



Reine Bündelung

- Kein Kauf, wenn $(R_A + R_B) < P_{AB}$
- Kauf, wenn $(R_A + R_B) \geq P_{AB}$

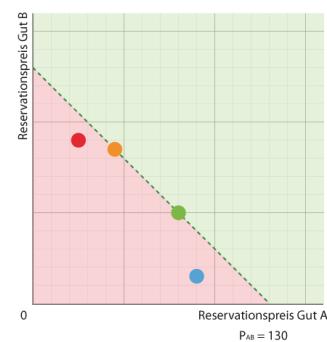
	A	B	Σ	Bündel
	80	85		
Mary	90	15	105	
Paul	80	50	130	
Peter	45	85	130	
Susan	25	90	115	
Gewinn	160	170	330	



Reine Bündelung

Wenn der Bündelpreis auf 130 gesetzt wird, wird das Bündel zwei mal gekauft und der Gewinn beträgt $2 \times 130 = 260$.

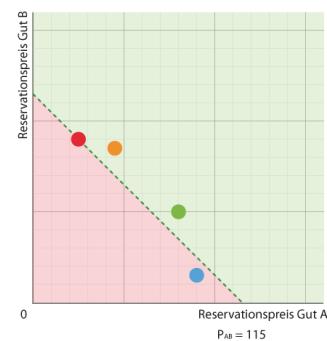
	A	B	Σ	Bündel
	80	85		
Mary	90	15	105	
Paul	80	50	130	✓
Peter	45	85	130	✓
Susan	25	90	115	
Gewinn	160	170	330	260



Reine Bündelung

Wenn der Bündelpreis auf 115 gesenkt wird, wird das Bündel drei mal gekauft und der Gewinn beträgt $3 \times 115 = 345$.

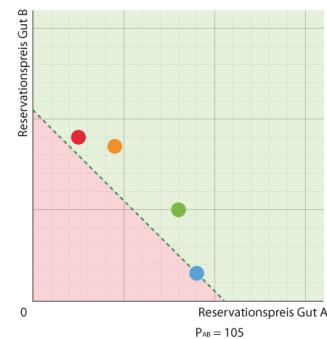
	A	B	Σ	Bündel
	80	85		
Mary	90	15	105	
Paul	80	50	130	✓
Peter	45	85	130	✓
Susan	25	90	115	✓
Gewinn	160	170	330	345



Reine Bündelung

Wenn der Bündelpreis auf 105 gesenkt wird, wird das Bündel vier mal gekauft und der Gewinn beträgt $4 \times 105 = 420$.

	A	B	Σ	Bündel
Mary	80	15	105	✓
Paul	80	50	130	✓
Peter	45	85	130	✓
Susan	25	90	115	✓
Gewinn	160	170	330	420

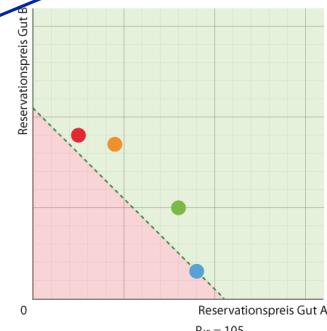


Reine Bündelung

- Durch Produktbündelungen kann das Unternehmen die Zahlungsbereitschaft auf Teilmärkten abschöpfen
- Produzent kann Umsatz und Gewinn steigern

Mit einem Bündelpreis von 105 wird der höchste Gewinn erreicht. Der Verkauf im Bündel führt zu einem höheren Gewinn als der Einzelverkauf.

	A	B	Σ	Bündel
Mary	80	15	105	✓
Paul	80	50	130	✓
Peter	45	85	130	✓
Susan	25	90	115	✓
Gewinn	160	170	330	420

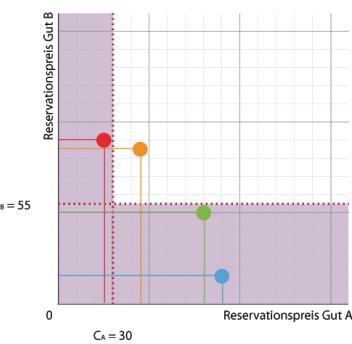


Bei hohen Grenzkosten lohnt sich reine Bündelung nicht

- Hohe Grenzkosten machen reine Bündelungs-Strategie uninteressant
- Der Einzelverkauf ist lukrativer

Wenn nun Kosten einberechnet werden, sinkt der Gewinn, und der Gewinn beim Bündel ist nun geringer als beim Einzelverkauf.

	A	B	Σ	Bündel
Kosten	80	85		105
Mary	90	15	105	✓
Paul	80	50	130	✓
Peter	45	85	130	✓
Susan	25	90	115	✓
Gewinn	100	60	160	-50% 80



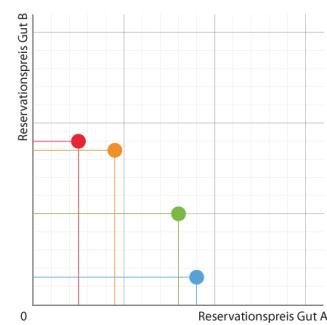
Wettbewerbsstrategien IV – Produktbündelung

- Einführung
- Einzelverkauf vs. reine Bündelung
- **Gemischte Bündelung**
- Preisstrategie bei positiver Korrelation der Reservationspreise

Gemischte Bündelung

- Verbindet die Vorteile der reinen Bündelung mit den Vorteilen des Einzelverkaufs
- Bei hohen Kosten sind bessere Ergebnisse als mit dem Einzelverkauf möglich
- Herausforderung: Wahl der optimalen Preise

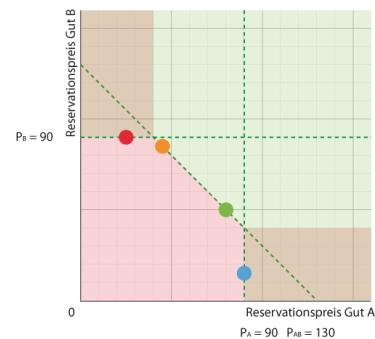
	A	B	Σ	Bündel
	?	?		?
Mary	90	15	105	
Paul	80	50	130	
Peter	45	85	130	
Susan	25	90	115	



Gemischte Bündelung

- Konsument kauft ein Produkt(bündel) nur, wenn dessen Preis kleiner oder gleich dem Reservationspreis ist und
- Konsument wählt die Alternative mit der höheren Konsumentenrente

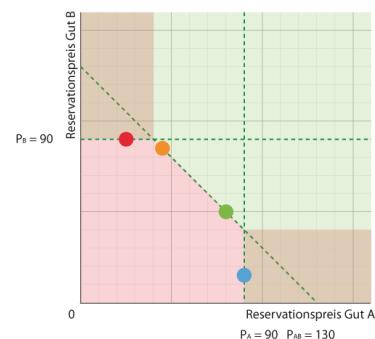
	A	B	Σ	Bündel
Mary	90	15	105	
Paul	80	50	130	
Peter	45	85	130	
Susan	25	90	115	



Gemischte Bündelung

- Kein Kauf, wenn $(R_A + R_B) < P_{AB}$ und $R_A < P_A$ und $R_B < P_B$
- Kauf Bündel, wenn $(R_A + R_B) \geq P_{AB}$ und $R_A \geq (P_{AB} - P_B)$ und $R_B \geq (P_{AB} - P_A)$ bzw. $R_A + R_B - P_{AB} \geq \max \{(R_A - P_A); (R_B - P_B)\}$
- Kauf A, wenn $R_A \geq P_A$ und $R_B < (P_{AB} - P_A)$ bzw. Kauf B, wenn $R_B \geq P_B$ und $R_A < (P_{AB} - P_B)$

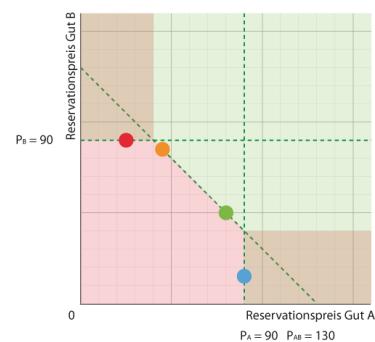
	A	B	Σ	Bündel
Mary	90	15	105	
Paul	80	50	130	
Peter	45	85	130	
Susan	25	90	115	



Gemischte Bündelung

Bei der gemischten Bündelung können die Preise von A und B jeweils auf 90 gesetzt werden, womit jeweils ein Konsument die Produkte kauft, und der Bündelpreis kann auf 130 gesetzt werden, womit zwei Konsumenten das Bündel kaufen. Damit wird der höchste Gewinn erzielt.

	A 90	B 90	Σ	Bündel 130
Mary	90	15	105	
Paul	80	50	130	✓
Peter	45	85	130	✓
Susan	25	90	115	
Gewinn	90	90	440	



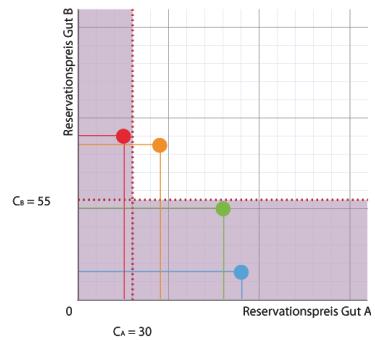
Gemischte Bündelung als Strategie zur Kostenoptimierung

- Ziel: Preise so setzen, dass Konsumenten auf den Kauf von Gütern verzichten, bei denen ihre Zahlungsbereitschaft unter den Kosten liegen

Wieder mit Einbezug der Kosten...
Sollen Güter einzeln oder im Bündel verkauft werden?

	A 30	B 55	Σ	Bündel 85
Mary	90	15	105	
Paul	80	50	130	
Peter	45	85	130	
Susan	25	90	115	

Ziel: Mary, Paul und Susan sollen das Bündel nicht kaufen, da ihr Reservationspreis für ein Produkt unter den Kosten sind.

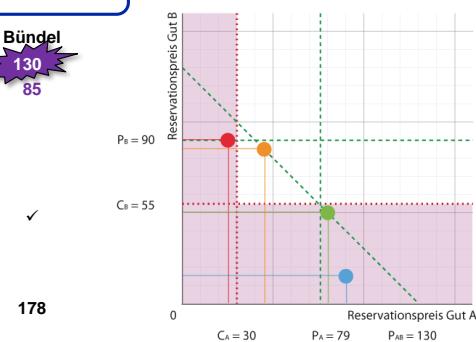


Gemischte Bündelung als Strategie zur Kostenoptimierung

- Ziel: Preise so setzen, dass Konsumenten auf den Kauf von Gütern verzichten, bei denen ihre Zahlungsbereitschaft unter den Kosten liegen
- Gemischte Bündelung führt zu höherem Gewinn als bei Einzelverkauf (178 vs. 160)

Da nur noch Peter für die Bündelung in Frage kommt, muss der Preis Von A um 1 CHF gesenkt werden, damit Paul auf das Bündel verzichtet und nur A kauft.

Kosten	A 79 30	B 90 55	Σ	Bündel 130 85
Mary	90	15	105	
Paul	80	50	130	
Peter	45	85	130	
Susan	25	90	115	
Gewinn	98	35		178



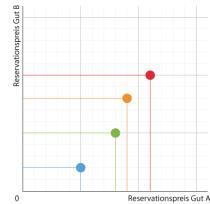
Wettbewerbsstrategien IV – Produktbündelung

- Einführung
- Einzelverkauf vs. reine Bündelung
- Gemischte Bündelung
- **Preisstrategie bei positiver Korrelation der Reservationspreise**

Hintergrund Korrelation

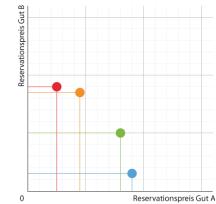
- Wie erkenne ich positive Korrelation?
→ Die Preise steigen in die gleiche Richtung („je mehr desto mehr“)

Mary	50	20	70
Paul	80	50	130
Peter	90	80	170
Susan	110	100	210



- Wie erkenne ich negative Korrelation?
→ Die Preise steigen in entgegengesetzte Richtung („je mehr desto weniger“)

Mary	50	100	150
Paul	80	80	160
Peter	90	50	140
Susan	110	20	130



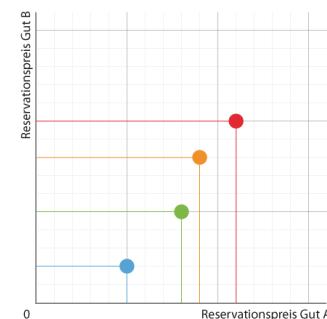
Bei positiver Korrelation der Reservationspreise

- Bei positiver Korrelation der Reservationspreise erzielt der Anbieter keinen zusätzlichen Gewinn durch Bündelung.

Bündelung kann nur dann einen Vorteil bringen, wenn keine positive Korrelation vorliegt.

	A	B	Σ	Bündel
	80	80		130
Mary	50	20	70	
Paul	80	50	130	✓
Peter	90	80	170	✓
Susan	110	100	210	✓
Gewinn	240	160	400	390

2.5%



Bündelung als Wettbewerbsstrategie

- **Verringert Preiskämpfe:** Differenziertes Angebot erschwert Markttransparenz und macht Preiskämpfe weniger wahrscheinlich
- **Erschwert Marktzutritt von Konkurrenten:** Reduziert Nachfrageelastizität dank stärkerer Kundenbindung und erhöht Marktzutrittskosten wegen kombinierter sunk costs
- **Vereinfacht Eroberung neuer Märkte:** Bestehende Marktmacht wird auf andere Märkte übertragen



Fazit

- Durch Produktbündelungen kann das Unternehmen
 - Umsatz und Gewinn steigern
 - Marktmacht sichern oder ausbauen
- Ob und welche Art der Bündelung angeboten wird, hängt massgeblich von der Höhe der Grenzkosten ab:
 - Bei hohen Grenzkosten: Keine Bündelung oder gemischte Bündelung
 - Bei tiefen Grenzkosten: Reine Bündelung
- Wenn die Reservationspreise positiv korrelieren, lohnt sich Bündelung nicht

Credits

- Clement, R.: Preis- und Erlösstrategien auf elektronischen Märkten, in: WISU 8/9/2001, S. 1176 – 1181

10 Datenrecht

DR. ALFRED FRÜH

Studium Digitale Kursbaustein Datenrecht

Begleit-Skript

Dieses Skript bereitet die Inhalte des Kursbausteins Datenrecht schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

10.1 Lektion 1: Einführung

Das Rechtssystem legt sich als normative Ebene über alle Lebensbereiche. Der Kursbaustein **Datenrecht** erläutert, wie dies in Bezug auf Daten geschieht. Datenrecht ist (noch) kein spezifisches Rechtsgebiet; es handelt sich eher um eine *Querschnittsmaterie*, die Rechtsregeln aus ganz verschiedenen Rechtsgebieten umfasst. Gegenstand dieser Querschnittsmaterie Datenrecht ist vor allem die *Zuordnung von Daten zu den Rechtsträgern* wie Personen oder Unternehmen. In diesem Kursbaustein geht es darum, die Zuordnungsinstrumente und deren Wirkungen kennenzulernen. Dies ist ausgesprochen wichtig, weil immer mehr Geschäftsmodelle auf Daten beruhen. Die Zuordnung der Daten durch das Recht wird damit zu einem wichtigen Faktor im Wettbewerb um die Datennutzung.

10.2 Lektion 2: Recht und Gesellschaft

Aber was ist das, **Recht**? In der Rechtstheorie oder Rechtsphilosophie wird Recht typischerweise mit drei (konstituierenden) Elementen beschrieben. Recht ist demnach:

- eine normative Ordnung,
- die (notfalls mit Zwang) durchgesetzt werden kann und
- über eine gewisse Legitimation verfügt.

Bei der Frage, woraus sich die **Legitimation** des Rechts ergibt, unterscheidet die Rechtstheorie zwischen *nature rechtlichen* und *rechtspositivistischen* Ansätzen. Erstere leiten die Legitimation aus übergeordneten – von den Menschen losgelösten – Ordnungsprinzipien ab, wobei oft religiöse, z.B. christliche Grundsätze herangezogen werden. Der Rechtspositivismus geht demgegenüber davon aus, dass für die Durchsetzung und Wirksamkeit des Rechts einzig dessen «positive» (vgl. lat. *ponere*: setzen, stellen, legen) Setzung notwendig ist. In westlichen Demokratien geschieht dies heute in aller Regel durch einen demokratisch legitimierten Gesetzgeber.

Die **Funktion des Rechts** als normative Ordnung liegt nach LUHMANN in der «zeitstabilen kontrafaktischen Sicherung von Erwartungshaltungen». Gewisse Erwartungen sollen also über längere Dauer geschützt sein, selbst wenn sie einmal enttäuscht werden: Das Recht erlaubt den Akteuren, auch beim nächsten Mal darauf zu vertrauen, dass der Vertragspartner die Gegenleistung erbringt, dass die Behörde sich ans Gesetz hält oder dass Sie eine Gegend nicht prinzipiell meiden müssen, obwohl sie früher dort einmal bestohlen worden sind.

10.3 Lektion 3: Rechtsgebiete

Es gibt im Recht eine grobe **Unterteilung** ins *Privatrecht*, das *öffentliche Recht* und das *Strafrecht*. In allen drei Bereichen gibt es übrigens spezifische Prozessrechte, welche die rechtlichen Verfahren regeln. Innerhalb dieser groben Dreiteilung werden wiederum weitere Rechtsgebiete unterschieden. Die Abgrenzungen zwischen den Rechtsgebieten sind nicht immer trennscharf. Das gilt besonders, wenn in privatrechtlichen Bereichen staatliche Behörden involviert sind, auf die dann öffentliches Verwaltungsrecht zur Anwendung kommt.

Das Privatrecht, um das es hier hauptsächlich geht, regelt das Verhältnis der Rechtssubjekte zueinander und zwischen Rechtssubjekten und Objekten des Rechts:

Regelungsgegenstand	Artikel	Rechtsgebiet
Personen	Art. 11 ff. ZGB	Natürliche Personen
	Art. 52 ff. ZGB	Juristische Personen
Personen zueinander	Art. 90 ff. ZGB	Ehe- und Scheidungsrecht

Regelungsgegenstand	Artikel	Rechtsgebiet
	Art. 252 ff. ZGB	Kindes- und Erwachsenenschutzrecht
	Art. 457 ff. ZGB	Erbrecht
Personen und Sachen	Art. 641 ff. ZGB	Sachenrecht
Personen und Immaterialgüter	PatG/URG/DesG	Immaterialgüterrecht (Patentrecht, Urheberrecht, Designrecht, u.a.)
Austauschverhältnisse zwischen Personen	Art. 1 ff. OR	Obligationenrecht (insbes. Vertragsrecht)
Personengesellschaften	Art. 530 ff. OR	Gesellschaftsrecht (einf. Ges., KollG- od. KommG)
Anteilsgesellschaften	Art. 620 ff. OR	Gesellschaftsrecht (Aktiengesellschaft, GmbH, Genossenschaft)
Wertpapiere	Art. 965 ff. OR	Wertpapierrecht

Tabelle 3:

Keines dieser Rechtsgebiete regelt explizit das Verhältnis zwischen Personen und Daten. Je nach Kontext können aber verschiedene Zuordnungsinstrumente einschlägig sein (s. [10.6](#))

10.4 Lektion 4: Grundbegriffe

Die Rechtssprache kennt unzählige besondere Begriffe. Zum besseren Verständnis sind hier **sechs wichtige Begriffspaare** genauer erläutert. Deren Kenntnis kann Missverständnisse vermeiden und ermöglicht eine präzisere Auseinandersetzung mit der Materie.

Begriff	Erklärung	Beispiele
Rechtssubjekte	Tatsächliche oder potenzielle Träger von Rechten	Natürliche und juristische Personen
Objekte des Rechts	Gegenstände, an denen Rechte begründet werden können	Sachen, Immaterialgüter, Forderungen
Gesetz im materiellen Sinn	Generell-abstrakte Rechtsnormen (Sollensanordnungen) unabhängig von deren Zustandekommen	Bundesverfassung, ZGB, Covid-19-Verordnungen, Kreisschreiben der Gerichte
Gesetz im formellen Sinn	Generell-abstrakte Rechtsnormen, die in einem formellen Gesetzgebungsverfahren erlassen worden sind	Bundesgesetze, kantonale Gesetze
Objektives Recht	Das für alle geltende Recht; gemeint sind bestimmte konkrete Rechtsnormen	Gesellschaftsrecht, Arbeitsrecht
Subjektive Rechte	Die sich aus den Rechtsnormen ableitenden individuellen Rechte	Aktionärsrechte, Recht auf Lohnfortzahlung
Absolute Rechte	Ein (subjektives) gegenüber Jedermann (erga omnes) wirkendes Recht	Sacheigentum
Relative Rechte	Ein (subjektives) nur gegenüber bestimmten Personen (inter partes) wirkendes Recht	Vertragliches Kündigungsrecht
Norm	Eine Bestimmung des objektiven Rechts	Art. 41 OR
Anspruch	Eine Berechtigung, welche sich für ein Rechtssubjekt aus einer Norm ergibt	Schadenersatzanspruch gemäss Art. 41 OR

Begriff	Erklärung	Beispiele
Tatbestand	Voraussetzungen, bei deren Vorliegen eine Rechtsfolge eintritt. Oft im Gesetz enthalten, teilweise kommen durch die Rechtsprechung der Gerichte weitere Voraussetzungen hinzu	Schaden, Widerrechtlichkeit, Kausalzusammenhang und Verschulden bei Art. 41 OR
Rechtsfolge	Die Wirkung einer Norm, wenn ein Tatbestand erfüllt ist	Bei Art. 41 OR: Pflicht, den Schaden zu ersetzen

Tabelle 4:

10.5 Lektion 5: Daten als Gegenstand des Rechts

Daten können verschiedenen Formen annehmen. Das lässt sich mit der Informationspyramide beschreiben.

Auf der untersten Ebene – man spricht auch von der **syntaktischen Ebene** – liegen *Daten* als eine Folge von (elementaren) Zeichen vor. Sie sind für Menschen unverständlich. Auf der darüber liegenden **semantischen Ebene** haben die Daten die Form von *Information*; auf dieser Ebene können sie vom Menschen verstanden werden. Zuoberst liegt die **pragmatische Ebene**, auf der Information sich als *Wissen* manifestiert.

Ausserhalb der Informationspyramide liegt die **strukturelle Ebene**. Sie bezeichnet die *physischen Träger*, auf denen die Daten festgelegt werden (z.B.

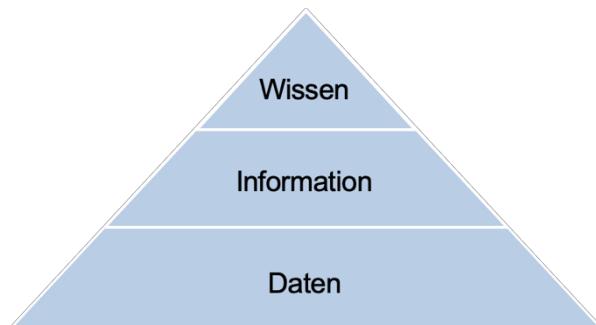


Abbildung 2: Informationspyramide

USB-Stick, Handyspeicher oder Server).

10.6 Lektion 6: Grundsätze der Zuordnung von Daten

Die Zuordnung von Daten greift in den **Interessenausgleich** zwischen dem/der Einzelnen und der Allgemeinheit ein: Mehr Datenuordnung geht zulasten der Allgemeinheit, eine geringere Zuordnung der Daten zu deren Gunsten.

Das Recht kann eine **rechtliche Zuordnung** oder den **rechtlichen Schutz der faktischen Zuordnung** vorsehen.

- *Rechtliche Zuordnung* bedeutet: Weitgehende Kontrolle über das Schutzgut. Das Schutzgut kann wiedererlangt werden oder allen Dritten kann deren Nutzung verboten werden.
- *Rechtlicher Schutz der faktischen Zuordnung* bedeutet: Verstöße gegen den faktischen Gewahrsam an den Daten werden rechtlich sanktioniert. Es besteht aber kein Anspruch, die Kontrolle über die Daten zurückzuerlangen.

Wenn man die Zuordnungsinstrumente des *Privatrechts* genauer ansieht, ergibt sich im Einzelnen:

- Das **Sachenrecht** vermittelt eine rechtliche Zuordnung – aber nur an Sachen (z.B. Datenträgern) und nicht an den Daten selbst. Die Ausdehnung des Sachbegriffs auf Daten (das sog. «Dateneigentum») wurde in Politik und Wissenschaft diskutiert, aber verworfen.
- Die **Immaterialgüterrechte** vermitteln eine rechtliche Zuordnung an bestimmten Informationen (Daten auf der semantischen Ebene), die die Schutzvoraussetzungen erfüllen (s. [10.7](#) und [10.8](#)).
- Das **Gesetz gegen den unlauteren Wettbewerb (UWG)** bietet «nur» einen rechtlichen Schutz der faktischen Zuordnung (s. [10.9](#)).
- Mit dem **Vertragsrecht** kann die Zuordnung von Daten zwischen zwei Parteien (aber nur zwischen diesen) geregelt werden (s. [10.10](#))

Auch das *Strafrecht* kann ein Zuordnungsinstrument sein:

- Das **Strafgesetzbuch** enthält verschiedene Bestimmungen für den rechtlichen Schutz der faktischen Zuordnung, namentlich die unbefugte Datenbeschaffung (Art. 143 StGB), das unbefugte Eindringen in ein Datenverarbeitungssystem (Art. 143bis StGB), die Datenbeschädigung (Art. 144bis StGB) und der betrügerische Missbrauch einer Datenverarbeitungsanlage (Art. 147 StGB).

10.7 Lektion 7: Patentrecht

Das Patentrecht soll **Innovation** fördern. Es gewährt unter gewissen *Schutzvoraussetzungen* Rechte für *Erfindungen*. Eine Erfindung ist eine Lehre zum technischen Handeln. Schutzvoraussetzungen sind die Neuheit und das Nicht-Naheliegen der Erfindung sowie deren gewerbliche Anwendbarkeit.

Daten als solche sind nicht geschützt. In Form von Information, welche die Schutzvoraussetzungen erfüllt (d.h. in Form der in der Patentschrift enthaltenen Lehre zum technischen Handeln) sind sie jedoch «mitgeschützt». **Computerprogramme** sind grundsätzlich vom Patentschutz ausgeschlossen. Eine Ausnahme gilt für sog. computerimplementierte Erfindungen. Bei diesen wird eine Software dazu verwendet, einen bestimmten technischen Effekt zu erzielen.

10.8 Lektion 8: Urheberrecht

Das Urheberrecht soll **kreative Tätigkeiten** fördern. Es gewährt unter gewissen *Schutzvoraussetzungen* Rechte an *Werken der Literatur und Kunst*. Ein Werk der Literatur und Kunst ist dann geschützt, wenn es sich um eine geistige Schöpfung mit individuellem Charakter handelt.

Daten als solche sind nicht geschützt. In Form von Information, welche die Schutzvoraussetzungen erfüllt (also etwa in Form eines urheberrechtlich geschützten Texts) sind sie jedoch «mitgeschützt». **Computerprogramme** sind ausdrücklich geschützt, gemäss Art. 2 Abs. 3 URG handelt es sich dabei um Werke der Literatur und Kunst.

10.9 Lektion 9: Wettbewerbsrecht (UWG)

Im Wettbewerb gilt grundsätzlich das Prinzip der **Nachahmungsfreiheit**. Die Wettbewerber können einander also grundsätzlich nachahmen. Wie das Immaterialgüterrecht schafft aber auch das zum Wettbewerbsrecht gehörende Gesetz gegen den unlauteren Wettbewerb (UWG) Ausnahmen hiervon. Dies geschieht beispielsweise, indem das UWG Geschäftsgeheimnisse schützt. Darüber hinaus verhindert es mit einer spezifischen Norm, nämlich Art. 5 lit. c UWG, die Übernahme fremder Leistungen unter bestimmten Voraussetzungen. Dieses «Abkopfern», d.h. die Übernahme durch eine Reproduktionstechnik und ohne angemessenen eigenen Aufwand ist unzulässig. Rechtsfolge sind privatrechtliche Beseitigungs-, Unterlassungs- und Schadenersatzansprüche und evtl. auch eine strafrechtliche Sanktion.

10.10 Lektion 10: Vertragsrecht

Das Vertragsrecht vermittelt relative Rechte, die nur zwischen den Vertragsparteien (*inter partes*) wirken; die vereinbarten Regeln binden keine Dritten.

Bei der *vertraglichen Verwertung von Immaterialgüterrechten*, d.h. bei Verträgen über Patente oder Urheberrechte ist danach zu unterscheiden, ob das Patent oder Urheberrecht **übertragen** oder **lizenziert** wird. Die Vertragsparteien können aus dem gewährten Immaterialgüterrecht gegen Dritte vorgehen, welche die Rechte verletzen.

Bei *Verträgen über rein faktisch kontrollierte Daten* können die Vertragsparteien nicht mehr gegen Dritte vorgehen, wenn diese einmal an die Daten gelangt sind. Im Vertrag muss deshalb möglichst exakt umschrieben werden, was, bzw. welche Daten genau Vertragsgegenstand sind, welche Nutzungen erlaubt sind, wie lange die Nutzung dauert, wie die Geheimhaltung sichergestellt ist und was passiert, wenn vertragliche Pflichten verletzt werden.

© Digital Society Initiative

Anhang

Abs. Absatz (in einem Gesetz)

Art. Artikel (in einem Gesetz)

DesG Designgesetz

d.h. das heisst

einf. Ges Einfache Gesellschaft (Art. 530 ff. OR)

f./ff. folgende Seite(n)

GmbH Gesellschaft mit beschränkter Haftung (Art. 772 ff. OR)

KollG Kollektivgesellschaft (Art. 552 ff. OR)

KommG Kommanditgesellschaft (Art. 594 ff. OR)

lit.	Litera, Buchstabe (in einem Gesetz)
OR	Obligationenrecht
PatG	Patentgesetz
s.	siehe
u.a.	und andere
URG	Urheberrechtsgesetz
UWG	Gesetz über den unlauteren Wettbewerb
ZGB	Zivilgesetzbuch

Tabelle 5: Glossar

10.11 Folien

The slide features the logos of the University of Zurich (UZH) and the Center for Information Technology Society and Law (CITS). The UZH logo is a circular emblem with a building and text. Next to it, the text "Universität Zürich" is written above "UZH". Below these, "Digital Society Initiative" is written. To the right, the CITS logo consists of the text "CENTER FOR INFORMATION TECHNOLOGY SOCIETY AND LAW — ITSL" above a graphic of stylized human figures connected by a network of lines.

Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 1: Einführung

Worum es in Kursbaustein 15 geht

- Was ist Recht?
- Welche Rechtsgebiete lassen sich unterscheiden?
- Welches sind die wichtigsten Grundbegriffe des Rechts?
- Was sind Daten?
- Wem gehören Daten?
- Welche Rechtsnormen kommen zur Anwendung?

Seite 2

Lernziele Datenrecht

- Sie kennen die wichtigsten Rechtsgebiete
- Sie kennen die Grundbegriffe des Rechts
- Sie kennen die Grundsätze der Zuordnung von Daten
- Sie sind in der Lage, Daten anhand der wichtigsten Zuordnungsinstrumente einer natürlichen oder juristischen Person zuzuordnen

Seite 3

Datenrecht

- Gegenstand: Zuordnung von Daten zu einem Rechtsträger (Personen oder Unternehmen)
- Relevanz: Daten als Gold der heutigen Zeit
- «Querschnittsmaterie»: Verschiedene Rechtsgebiete kommen zur Anwendung

Seite 4



CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 2: Recht und Gesellschaft

Was ist Recht?

Konstituierende Elemente

- Normative Ordnung
- Durchsetzbarkeit (u.U. mit Zwang)
- Legitimation

Seite 6

Legitimation von Recht

Rechtssubjekte

Träger von Rechten

Objekte des Rechts

Gegenstände, an denen
Rechte begründet
werden können

Beispiel: Natürliche oder
juristische Person (Unternehmen)

Beispiel: Sache (Tisch, Stuhl),
Immateriagüter (Patent)

Seite 7

Funktionen von Recht

Luhmann Niklas (1927 – 1998)

Funktion des Rechts: **Stabilisierung einer Erwartungshaltung**

- Schutz von Erwartungshaltungen, auch wenn sie einmal enttäuscht werden
- Z.B. Strafrecht

Seite 8



Digital Society Initiative

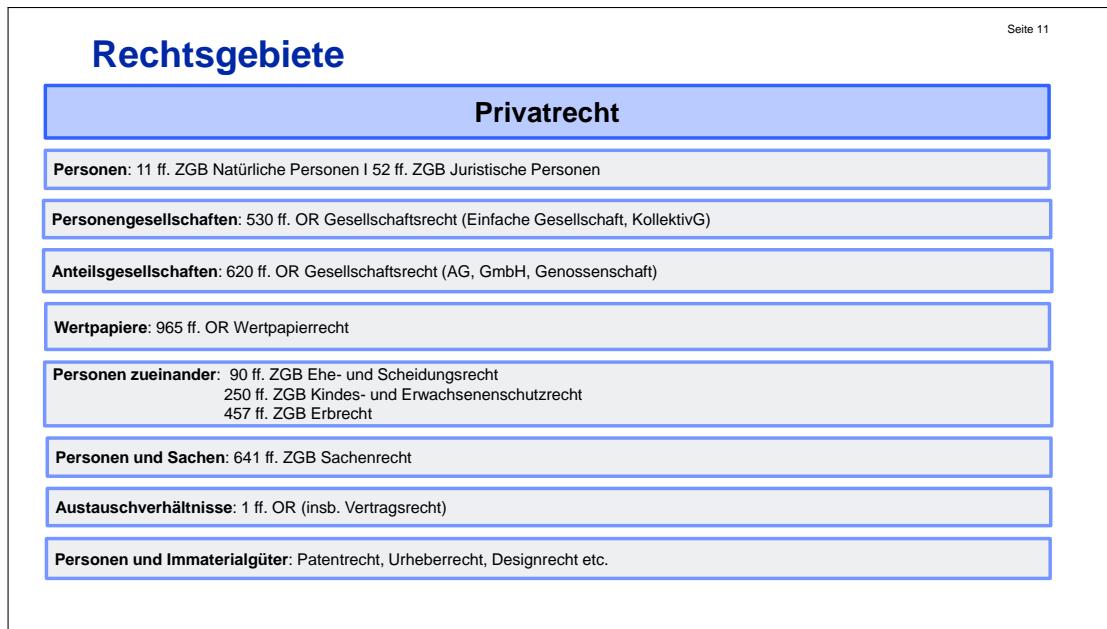
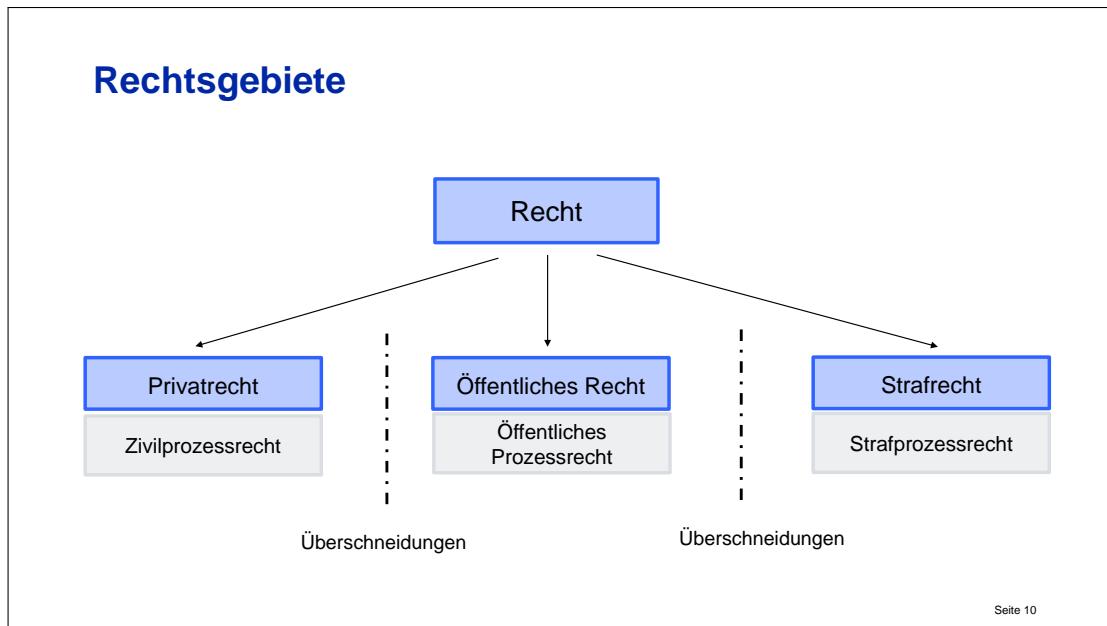
CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 3: Rechtsgebiete



The slide features the logo of the University of Zurich (UZH) and the Center for Information Technology Society and Law (ITSL). A banner at the bottom reads "Digital Society Initiative". The title "Studium Digitale" is prominently displayed in blue, followed by "Kursbaustein 15: Datenrecht" and "Lektion 4: Grundbegriffe".

Grundbegriffe

Rechtssubjekte	Objekte des Rechts
Träger von Rechten	Gegenstände, an denen Rechte begründet werden können

Beispiel: Natürliche oder juristische Person (Unternehmen)

Beispiel: Sache (Tisch, Stuhl), Immaterialgüter (Patent)

Seite 13

Grundbegriffe

Gesetz im materiellen Sinn

Alle Rechtsnormen (generell-abstrakt)

Beispiel: Verfassung, Kreisschreiben der Steuerbehörden

Gesetz im formellen Sinn

Rechtsnormen, die in einem formellen Gesetzgebungsverfahren, das in der Verfassung vorgesehen ist, erlassen worden sind (Legislative, Parlament)

Beispiel: Datenschutzgesetz, kantonales Gesundheitsgesetz des Kantons Zürich

Seite 14

Grundbegriffe

Objektives Recht

Für alle geltendes Recht, konkrete Rechtsnormen

Beispiel: Datenschutzrecht, Arbeitsrecht

Subjektives Recht

Individuelle Rechte einer Person, die sich aus Rechtsnormen ableiten

Beispiel: Datenschutzrechtlicher Auskunftsanspruch, Recht auf Lohn

Seite 15

Grundbegriffe

Absolutes Recht

Subjektives Recht, das gegenüber Jedermann (erga omnes) geltend gemacht werden kann

Beispiel: Eigentumsrecht

Relatives Recht

Subjektives Recht, das nur gegenüber bestimmten Personen (inter partes) geltend gemacht werden kann

Beispiel: Vertragliche Rechte (Kündigungs- oder Rücktrittsrecht)

Seite 16

Grundbegriffe

Norm

Bestimmung des objektiven Rechts

Beispiel: Art. 641 ZGB

Anspruch

Berechtigung, die sich aus einer Norm ergibt

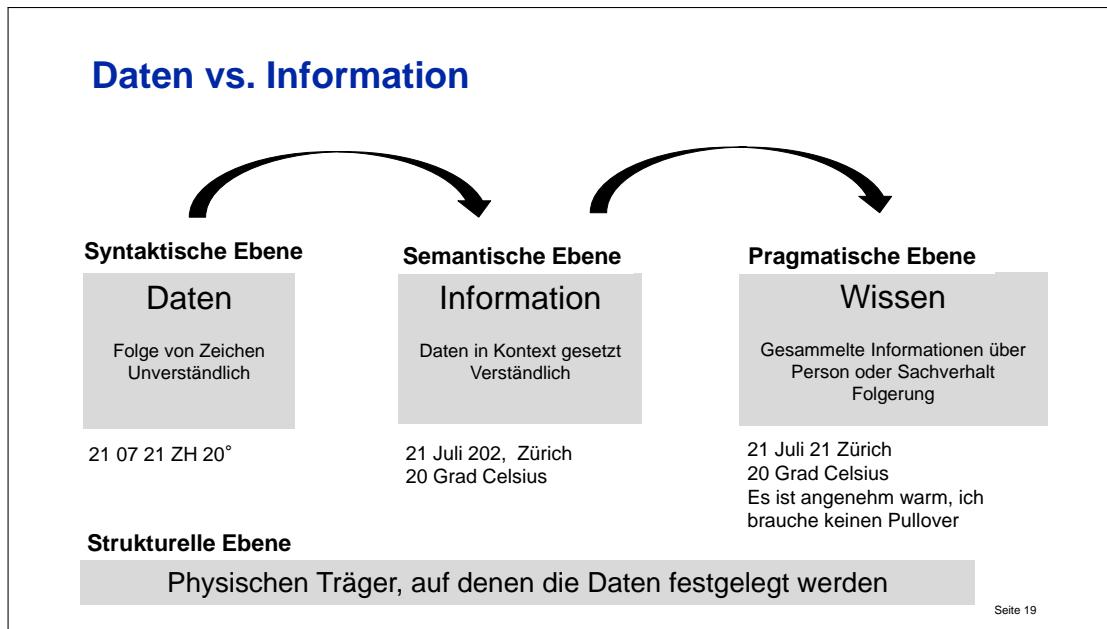
Beispiel: Herausgabeanspruch des Eigentümers, Art. 641 Abs. 2 ZGB

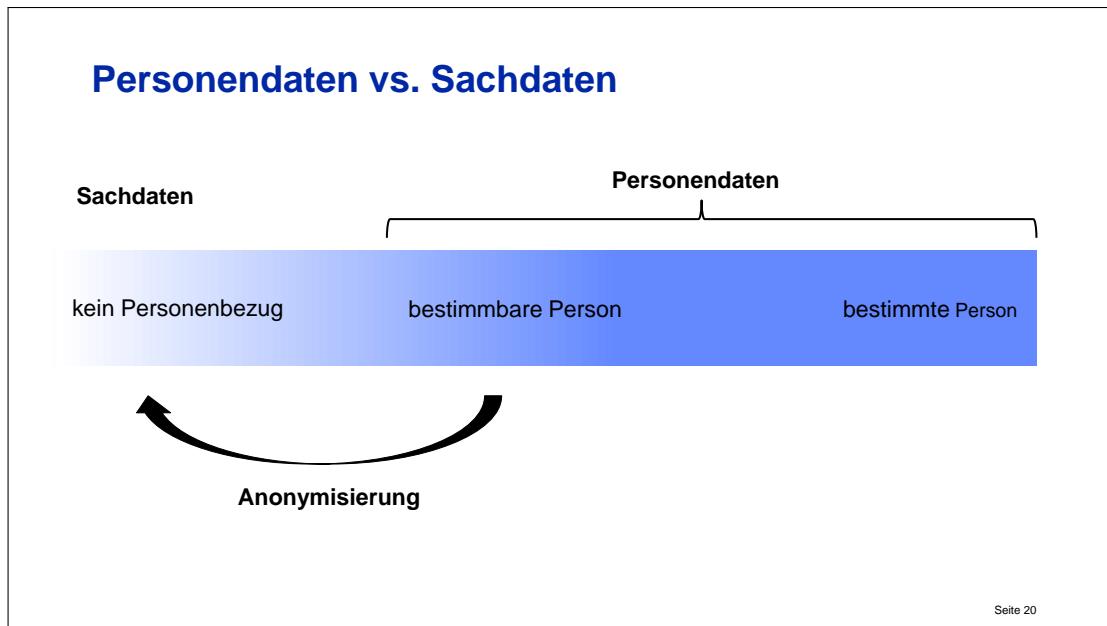
Seite 17

Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 5: Daten als Gegenstand des Rechts





Seite 20

Universität Zürich
Digital Society Initiative

CENTER FOR INFORMATION TECHNOLOGY SOCIETY AND LAW — ITSL

Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 6: Grundsätze der Zuordnung von Daten

Interessensausgleich der Zuordnung

Einzelner

Allgemeinheit



Je weniger die Daten einer Einzelperson zugeordnet werden, desto mehr kann die Allgemeinheit davon profitieren.

Beispiel: Zitierfreiheit und Eigengebrauch im Urheberrecht.

Einzelner

Allgemeinheit



Je mehr die Daten einer Einzelperson zugeordnet werden, weniger kann die Allgemeinheit davon profitieren.

Beispiel: Patentrecht führt dazu, dass Dritte die Erfindung i.d.R. nicht nutzen können.

Seite 22

Zuordnung von Daten zu einem Rechtsträger

Rechtliche Zuordnung



Kontrolle über das Schutzgut

- Wiedererlangung des Schutzgutes bei Abhandenkommen
- Dritten kann die Nutzung verboten werden

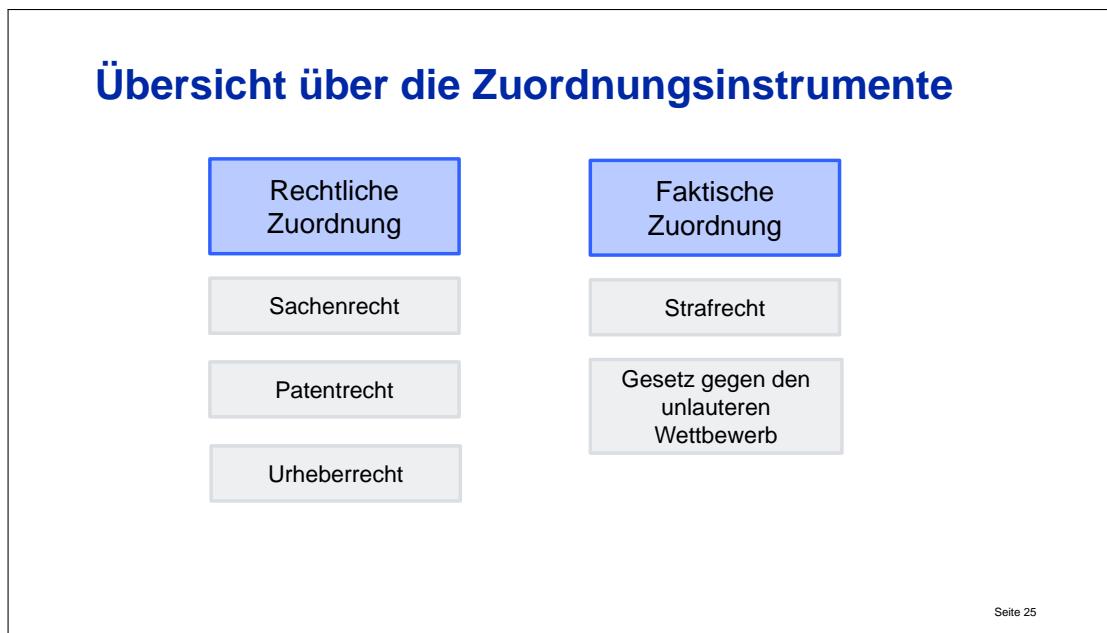
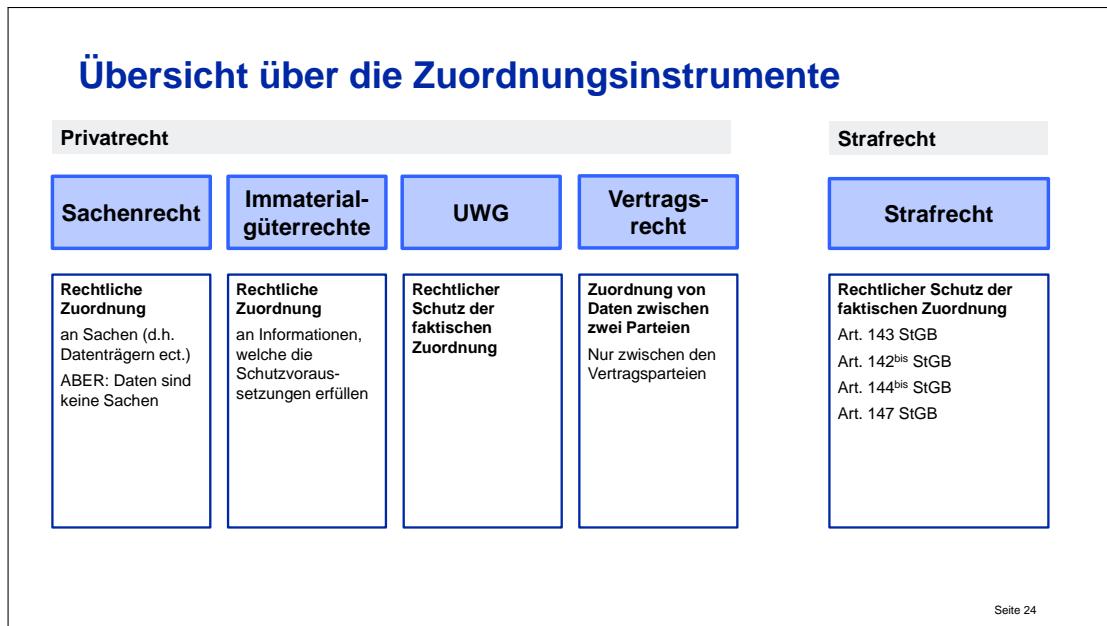
Rechtlicher Schutz der Faktische Zuordnung



Keine Kontrolle über das Schutzgut

- Sanktionierung bei Verstößen gegen den faktischen Gewahrsam
- Kein Anspruch, Daten wieder zurückzuerlangen

Seite 23



Sachenrecht

Inhalt: Umfassendes Herrschafts- und Abwehrrecht ans Sachen, welches gegenüber jedermann (erga omnes) geltend gemacht werden kann.

Was sind Sachen?

Sachen sind unpersönliche, Körperliche, für sich bestehende Gegenstände, die der menschlichen Herrschaft unterworfen werden können.

Daten: fehlende Körperlichkeit, keine Sachen

Seite 26



Digital Society Initiative

CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 7: Patentrecht

Patentrecht

Ziel: Förderung von Innovation

Geschütztes Gut: Erfindung, d.h. Lehre vom technischen Handeln

Schutzvoraussetzungen

- Neuheit
- Nicht-Naheliegen
- Gewerbliche Anwendbarkeit

Seite 28



Universität
Zürich^{UZH}

Digital Society Initiative

CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 8: Urheberrecht

Urheberrecht

Ziel: Förderung der Kreativität

Geschütztes Gut: Werk, d.h. Werke der Literatur und Kunst

Schutzvoraussetzungen

- Geistige Schöpfung
- Individueller Charakter

Ausnahme: Computerprogramme | Fotografien

Seite 30

Urheberrecht

Schutz von Software

Patentrecht

Grundsatz:



Urheberrecht



Ausnahme:



Art. 2 Abs. 3 URG

CII: Computerimplementierte Erfindungen

Seite 31



Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 9: Wettbewerbsrecht (UWG)

Gesetz gegen den unlauteren Wettbewerb (UWG)

Ziel: Schutz der Qualität des Wettbewerbs

Grundsatz: Prinzip der Nachahmungsfreiheit

Ausnahmen

- Schutz von Geschäftsgeheimnissen
- Verbot der Übernahme einer fremden Leistung (Art. 5 Abs. 1 lit. c UWG)

Gesetz gegen den unlauteren Wettbewerb (UWG)

Art. 5 Abs. 1 lit. c UWG Übernahme einer fremden Leistung

Tatbestandsmerkmale

- Marktreifes Arbeitsergebnis
- Übernahme und Verwertung "als solche"
- Technisches Reproduktionsverfahren
- Ohne angemessenen eigenen Aufwand

Seite 34



Universität
Zürich^{UZH}

Digital Society Initiative

CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 15: Datenrecht

Lektion 10: Vertragsrecht

Vertragsrecht

Vertragsrecht = Wirkung inter partes

- Immaterialgüterrechte: Übertragung und Lizenzierung von
Immaterialgüterrechten
- Vertrag über faktisch kontrollierte Daten: Keine Möglichkeit, gegen Dritte
vorzugehen

11 Datenschutzrecht

DR. ALFRED FRÜH

Studium Digitale Kursbaustein Datenschutzrecht

Begleit-Skript

Dieses Skript bereitet die Inhalte des Kursbausteins Datenschutzrecht schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

11.1 Lektion 1: Einführung

Das Rechtsgebiet des **Datenschutzrechts**, welches in den Kontext des Datenrechts gehört, ist in den vergangenen Jahren immer wichtiger geworden. Dies liegt zum einen daran, dass die Digitalisierung fast alle Lebensbereiche erfasst hat. Zum anderen sind in den Datenschutzgesetzen heute – anders als früher – Sanktionen vorgesehen, weswegen die Regeln mehr Beachtung finden.

Heute wird das Datenschutzrecht oft als Instrument zum Schutz der Individuen vor den übermächtigen privaten Internetunternehmen wahrgenommen, obwohl es ursprünglich eigentlich zur Abwehr hoheitlicher – d.h. staatlicher – Eingriffe geschaffen wurde. Tatsächlich deckt es beide Bereiche ab und regelt sowohl Datenbearbeitungen durch Private als auch durch Behörden. Damit gehört das Datenschutzrecht gleichzeitig zum *Privatrecht* und zum *öffentlichen Recht*.

11.2 Lektion 2: Geltungsbereich

Das Datenschutzrecht hat einen sehr weiten Geltungsbereich. In sachlicher Hinsicht ist es anwendbar, sobald eine *Bearbeitung von Personendaten* vorliegt.

Als **Bearbeiten** gilt jeder *Umgang mit Personendaten*.

Personendaten sind alle Angaben, die sich auf eine *bestimmte oder bestimmbar* natürliche Person beziehen. Bestimmt ist eine Person, wenn sie mit den Daten zweifelsfrei identifiziert werden kann. Bestimmbar ist eine Person, wenn sie zwar nicht aufgrund eines Datums allein, aber aufgrund weiterer vorhandener Informationen (ohne unverhältnismässigen Aufwand) identifiziert werden kann.

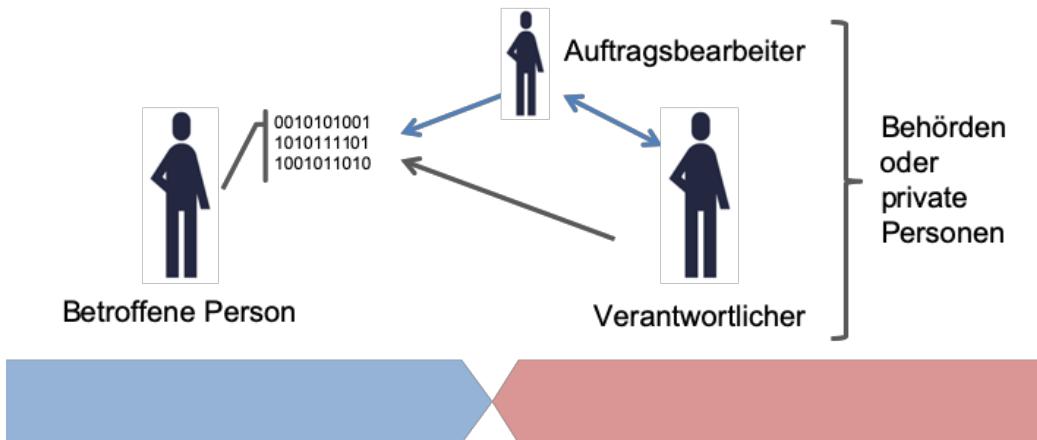


Abbildung 3: Involvierte Personen

Als besonders schützenswert gelten zudem Personendaten über religiöse, weltanschauliche, politische oder gewerkschaftliche Ansichten oder Tätigkeiten, Daten über die Gesundheit, die Intimsphäre oder die Zugehörigkeit zu einer Rasse oder Ethnie, genetische bzw. biometrische Daten und Daten über verwaltungs- und strafrechtliche Verfolgungen oder Sanktionen und Sozialhilfe.

In *persönlicher* Hinsicht sind die Handlungen von **privaten Personen und Behörden** erfasst. Je nach Rolle, welche diese einnehmen, werden sie als **Verantwortliche** oder **Auftragsdatenbearbeiter** qualifiziert.

Verantwortlicher ist, wer über den Zweck oder die Mittel der Datenbearbeitung entscheidet. Mit einem Vertrag kann der Verantwortliche einen Auftragsbearbeiter beziehen. Aus der Zugehörigkeit zur einen oder anderen Kategorie folgen jeweils unterschiedliche Pflichten.

11.3 Lektion 3: Ziele des Datenschutzrechts

Als **Hauptzwecke** des Datenschutzrechts gelten der Schutz der Persönlichkeit der betroffenen Personen und der Schutz von deren Grundrechten. In der Schweiz wird das Datenschutzgesetz beispielsweise als Konkretisierung des (gesetzlich in Art. 28 ZGB geregelten) Persönlichkeitsschutzes und der (auf Verfassungsstufe geregelten) Grundrechte – insbesondere des Schutzes vor Missbrauch der persönlichen Daten (Art. 13 Abs. 2 BV) – verstanden.

Im Einzelnen sind diese Zielsetzungen aber teilweise unscharf und häufig wer-

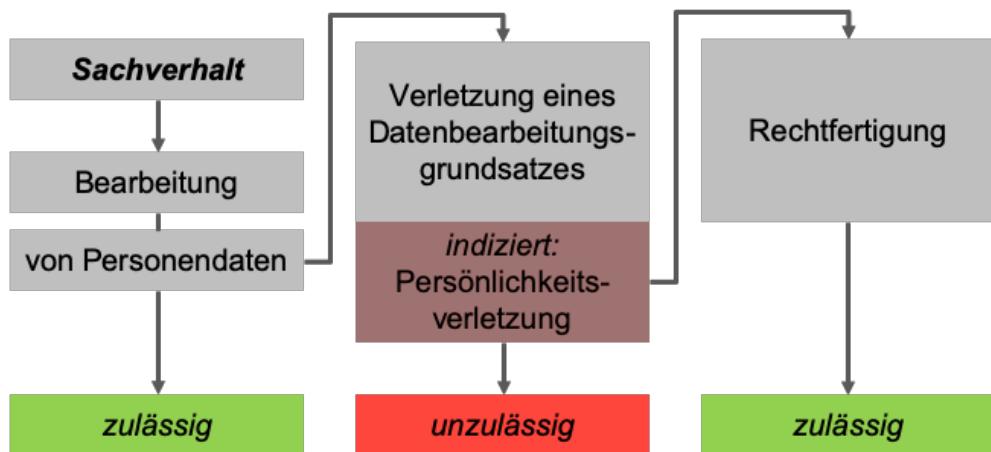


Abbildung 4: Prüfschema des DSG

den auch noch andere Zwecke genannt. Beides erschwert es, bei der Auslegung der einzelnen Normen auf den Gesetzeszweck abzustellen.

11.4 Lektion 4: Rechtsquellen

Im Datenschutzrecht existieren auf verschiedenen Ebenen Bestimmungen, die relevant sein können. Auf internationaler Ebene ist für die Schweiz beispielsweise die sog. Konvention 108 des Europarats massgebend. Die Schweiz hat diese Konvention unterzeichnet und ist dadurch *rechtlich* verpflichtet, deren Bestimmungen sinngemäss im nationalen Recht umzusetzen. Eher *wirtschaftliche* Auswirkungen auf die Schweiz hat demgegenüber die Datenschutzgrundverordnung (DSGVO) der Europäischen Union. Weil die DSGVO über das Gebiet der EU hinauswirkt, müssen auch viele Schweizer Unternehmen deren Regeln einhalten. Ausserdem ist die Schweiz bestrebt, ihr Datenschutzrecht dem Schutzniveau der DSGVO anzupassen. Neben dem Eidgenössischen Datenschutzgesetz (DSG) auf Bundesebene gibt es auch auf kantonaler Ebene Datenschutzgesetze. Sie regeln das Handeln der kantonalen Behörden.

11.5 Lektion 5: Funktionsweise des Datenschutzrechts

Das Schweizer Datenschutzgesetz (DSG) ist nach dem folgenden Schema aufgebaut:

Für die Zulässigkeit einer Datenbearbeitung ist also entscheidend, ob die Datenbearbeitungsgrundsätze eingehalten werden und – falls dies nicht zutrifft – ob Rechtfertigungsgründe vorliegen. In der Tatsache, dass das DSG Verletzungen von Datenbearbeitungsgrundsätzen mit Persönlichkeitsverletzungen gleichsetzt, zeigt sich der präventive Charakter des Datenschutzrechts. Im Vergleich zu anderen Gesetzen ist die Regelungstechnik des Datenschutzrechts eher untypisch, weil sehr viele Verhaltensweisen zunächst einmal unzulässig sind, von diesen aber wiederum sehr viele gerechtfertigt werden können. Andere Gesetze erfassen weniger Verhaltensweisen und brauchen entsprechend auch weniger breite Ausnahmetatbestände.

11.6 Lektion 6: Datenbearbeitungsgrundsätze

Sowohl im DSG als auch in der DSGVO gibt es zahlreiche Datenbearbeitungsgrundsätze. Sie sind zwar im Grossen und Ganzen deckungsgleich, unterscheiden sich aber teilweise in den Details und in der Bezeichnung.

Datenbearbeitungs- grundsatz (DSG)	Artikel (DSG)	Datenbearbeitungs- grundsatz (DSGVO)	Artikel (DSGVO)
Rechtmäßigkeit	Art. 5 Abs. 1	"	Art. 5 Abs. 1 Bst. a
Bearbeitung nach Treu und Glauben	Art. 5 Abs. 2	"	Art. 5 Abs. 1 Bst. a
Verhältnismäßigkeit	Art. 5 Abs. 2 und 4	Datenminimierung / Speicherbegrenzung	Art. 5 Abs. 1 Bst. c und e
Zweckbindung	Art. 5 Abs. 3	"	Art. 5 Abs. 1 Bst. b
Erkennbarkeit	Art. 5 Abs. 3	Transparenz	Art. 5 Abs. 1 Bst. a
Richtigkeit der Da- ten	Art. 5 Abs. 5	"	Art. 5 Abs. 1 Bst. d

Datenbearbeitungs- grundsatz (DSG)	Artikel (DSG)	Datenbearbeitungs- grundsatz (DSGVO)	Artikel (DSGVO)
Datenschutz durch Technik	Art. 6	"	Art. 25
Datensicherheit	Art. 7 Abs. 1	Integrität und Ver- traulichkeit	Art. 5 Abs. 1 Bst. f

Tabelle 6:

Für die moderne Datenwirtschaft sind besonders zwei dieser Grundsätze wichtig: Der Grundsatz der **Verhältnismässigkeit** verlangt, dass eine bestimmte Datenbearbeitung tatsächlich erforderlich ist. Der Grundsatz der **Zweckbindung** besagt, dass Daten nur zu demjenigen Zweck bearbeitet werden können, der zum Zeitpunkt der Erhebung der Daten angegeben wurde. Beide Grundsätze können die Unternehmen insoweit einschränken, als sie Datensammlungen auf Vorrat und die Verwendung der Daten für neu entdeckte Zwecke zunächst einmal verunmöglichen.

11.7 Lektion 7: Rechtfertigungsgründe

Liegt ein Verstoss gegen einen Datenbearbeitungsgrundsatz vor, kommen drei Rechtfertigungsgründe in Frage:

- Die Einwilligung
 - Die betroffene Person kann in eine unzulässige Datenbearbeitung einwilligen, sofern
 - die Einwilligung gültig ist (d.h. mit angemessener Information, freiwillig und eindeutig erfolgt),
 - sie bei normalen Personendaten stillschweigend, bei besonders schützenswerten Personendaten aber ausdrücklich erfolgt
 - sie vor der Datenbearbeitung erfolgt ist und nicht widerrufen wird (was grundsätzlich jederzeit möglich ist).
- Eine gesetzliche Grundlage

- Eine Bearbeitung wird ausdrücklich von einem Gesetz erlaubt.
- Überwiegende private oder öffentliche Interessen
 - Es braucht eine Einzelfallabwägung der involvierten Interessen, die gegebenenfalls eine unzulässige Bearbeitung rechtfertigen.

11.8 Lektion 8: Rechtsdurchsetzung

In Bezug auf die Durchsetzung datenschutzrechtlicher Ansprüche ist die private von der hoheitlichen Rechtsdurchsetzung zu unterscheiden. In beiden Varianten kann sich die Durchsetzung gegen private Personen oder gegen Behörden richten.

Gehen **Private gegen andere Private** (meist Unternehmen) vor, haben sie eine Reihe von Ansprüchen. Dazu gehören beispielsweise:

- *Klagen auf Unterlassung oder Beseitigung* der Verletzung, oder auch bloss (als zu den genannten Klagen subsidiäres Mittel, d.h. wenn diese nicht möglich sind) auf *Feststellung* der Verletzung;
- *Klagen auf Schadenersatz, Genugtuung oder Gewinnherausgabe*;
- ein Anspruch auf *Vernichtung* der Daten;
- ein Anspruch auf *Verbot der Bekanntgabe der Daten an Dritte*.

Die betroffenen Personen haben gegenüber dem Verantwortlichen zudem:

- Ein Berichtigungsrecht,
- ein Auskunftsrecht und sogar
- ein Portabilitätsrecht, mit dem sie die Übertragung der Daten auf einen Dritten erwirken können.

Gehen **Behörden**, namentlich der Eidgenössische Datenschutz- und Öffentlichkeitsbeauftragte (EDÖB) oder die kantonalen Strafverfolgungsbehörden, **gegen Private** vor, haben sie ebenfalls eine Reihe von Möglichkeiten. Insbesondere können sie Bussgelder verhängen.

11.9 Lektion 9: Datenschutzrecht als Zuordnungsinstrument

Wie andere Rechtsgebiete kann auch das Datenschutzrecht ein Zuordnungsinstrument für Daten sein (zum Ganzen s. Kursbaustein 15). Tatsächlich ist das Datenschutzrecht ein **rechtliches Zuordnungsinstrument**, weil mehrere Elemente des Datenschutzrechts eine gewisse Zuordnung bewirken:

- Das *Auskunftsrecht* und das *Datenportabilitätsrecht* erlauben der betroffenen Person auch dann einen Zugriff auf die Personendaten, wenn diese von jemand anderem kontrolliert werden;
- mittels *Einwilligung* kann die betroffene Person zu einem gewissen Grad über ihre Personendaten verfügen und
- das jederzeitige *Widerrufsrecht* (der Einwilligung) sowie das *Widerspruchsrecht* geben der betroffenen Person ebenfalls ein Kontrollmittel in die Hand.

Die **Wirkung** des Zuordnungsinstruments «Datenschutzrecht» ist allerdings **beschränkt**, denn:

- es erfasst nur Daten natürlicher Personen;
- es ist auf die jeweils eigenen Personendaten beschränkt und
- es gibt keine Übertragung der datenschutzrechtlichen Ansprüche auf Dritte, wie dies bei anderen Zuordnungsinstrumenten (Sachenrecht, Immaterialgüterrecht) möglich ist.

© Digital Society Initiative

Anhang

Abs. Absatz (in einem Gesetz)

Art. Artikel (in einem Gesetz)

Bst. Buchstabe (in einem Gesetz)

BV Bundesverfassung

bzw. beziehungsweise

d.h. das heisst

DSG Schweizer Datenschutzgesetz

DSGVO Europäische Datenschutzgrundverordnung

EDÖB Eidgenössischer Datenschutz- und Öffentlichkeitsbeauftragter

EU Europäische Union

ZGB Zivilgesetzbuch

Tabelle 7: Glossar

11.10 Folien

The slide features a header bar with logos and text. On the left is the logo of Universität Zürich (UZH) with the text "Universität Zürich" and "Digital Society Initiative". In the center is the logo for the "CENTER FOR INFORMATION TECHNOLOGY SOCIETY AND LAW — ITSL". To the right is a graphic depicting a network of people connected by lines.

Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 1: Einführung

Worum es in Kursbaustein 16 geht

- Was ist Datenschutzrecht ?
- Wann findet Datenschutzrecht Anwendung?
- Welche Ziele verfolgt das Datenschutzrecht ?
- Welche Rechtsquellen liegen vor?
- Wie funktioniert das Datenschutzrecht ?
- Welches sind die Grundsätze der Datenbearbeitung?
- Wie kann Datenschutzrecht durchgesetzt werden?

Seite 2

Lernziele Datenschutzrecht

- Sie kennen die wichtigsten Ziele und die Rechtsquellen des Datenschutzes
- Sie kenne die Funktionsweise des Datenschutzrechts
- Sie kennen die wichtigsten Datenbearbeitungsgrundsätze
- Sie wissen, wie sich datenschutzrechtliche Ansprüche durchsetzen lassen
- Sie wissen, weshalb man Datenschutzrecht auch als Zuordnungsinstrument betrachten kann

Seite 3

Datenschutzrecht

- Relevanz: (Personen-)Daten als Gold der heutigen Zeit
- Sanktionen bei Nichtbeachtung
- Abwehr von hoheitlichen Eingriffen
- Schutz der Individuen vor privaten Internetriesen

Seite 4



Universität
Zürich^{UZH}

Digital Society Initiative

CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



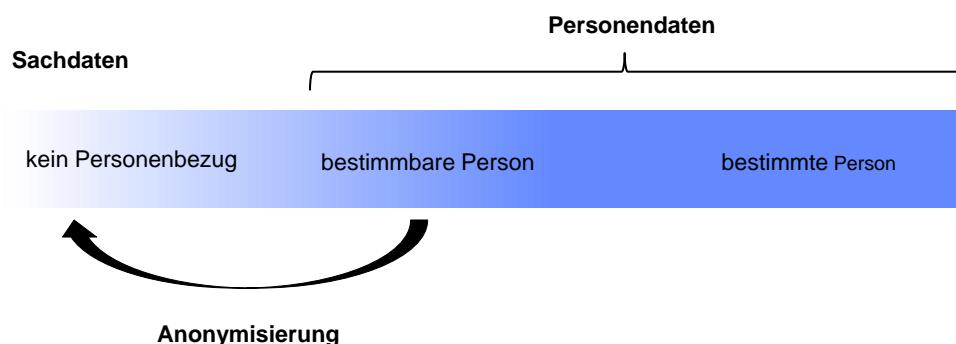
Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 2: Geltungsbereich

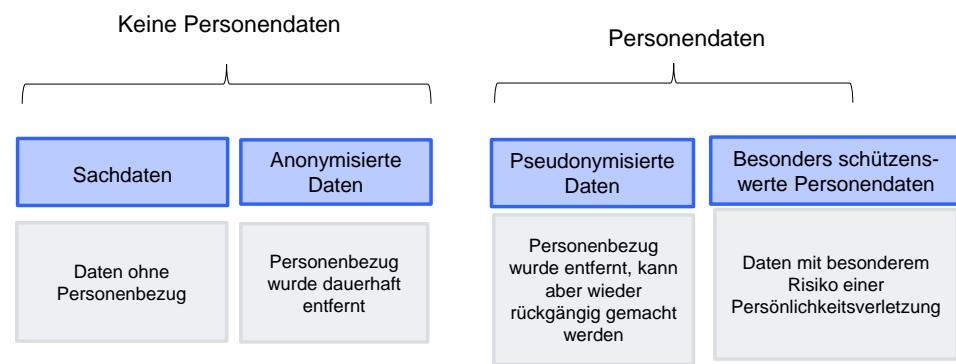
Geltungsbereich des Datenschutzrechts

Grundsatz: Datenschutzrecht findet immer Anwendung, wenn eine **Bearbeitung** von **Personendaten** vorliegt.



Seite 6

Sachlicher Geltungsbereich: Personendaten



Seite 7

Sachlicher Geltungsbereich: Bearbeiten

Grundsatz: Datenschutzrecht findet immer Anwendung, wenn eine **Bearbeitung von Personendaten** vorliegt.

Bearbeiten (Art. 5 lit. d DSG)

- Jeder **Umgang** mit Personendaten, unabhängig von den angewandten Mitteln und Verfahren
- Beschaffen, Speichern, Aufbewahren, Verwenden, Verändern, Bekanntgeben, Archivieren, Löschen oder Vernichten von Daten etc.

Seite 8

Persönlicher Geltungsbereich

Private

Bundesbehörden

Kantonale Behörden

Datenschutzgesetz (DSG)
Allgemeine Bestimmungen, Art. 5 ff. DSG
(z.B. Datenbearbeitungsgrundsätze)

Bestimmungen für
Privatpersonen, 30 ff.
DSG

Bestimmungen für
Bundesbehörden, 33 ff.
DSG

Kantonale Regelungen
ZH: IDG

Seite 9

The slide features the logos of the University of Zurich (UZH) and the Center for Information Technology Society and Law (CITS). The UZH logo is on the left, showing a circular seal with a building and the text "UNIVERSITÄT ZÜRICH". Below it is the text "Digital Society Initiative". The CITS logo is in the center, featuring the text "CENTER FOR INFORMATION TECHNOLOGY SOCIETY AND LAW — ITSL". To the right is a graphic of a network of people connected by lines.

Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 3: Ziele des Datenschutzrechts

Ziele des Datenschutzrechts

Persönlichkeitsschutz, Art. 28 ff. ZGB

Schutz vor Missbrauch von persönlichen Daten, Art. 13 Abs. 2 BV

Art. 1 DSG: Dieses Gesetz bezweckt den **Schutz der Persönlichkeit** und der **Grundrechte** von Personen, über die Daten bearbeitet werden.

Seite 11

Ziele des Datenschutzrechts

Art. 1 DSG: Dieses Gesetz bezweckt den **Schutz der Persönlichkeit** und der **Grundrechte** von Personen, über die Daten bearbeitet werden.

Weitere Ziele?

Grundrecht auf «Informationelle Selbstbestimmung»?

Schutz der Privatsphäre?

Problem: Unklare Zielsetzung(en)

Relevanz: Auslegung von Gesetzesbestimmungen stellen auf den Gesetzeszweck ab

Seite 12



Universität
Zürich^{UZH}
Digital Society Initiative

CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 4: Rechtsquellen

Rechtsquellen des Datenschutzes

International

Bindend

- Europarat: Konvention 108

Nicht bindend

- Datenschutzgrundverordnung (DSGVO)

National

- Datenschutzgesetz (DSG)

- Datenschutzverordnung (VDSG)

- Kantonale Datenschutzgesetz, z.B. ZH IDG

Seite 14



Universität
Zürich^{UZH}

Digital Society Initiative

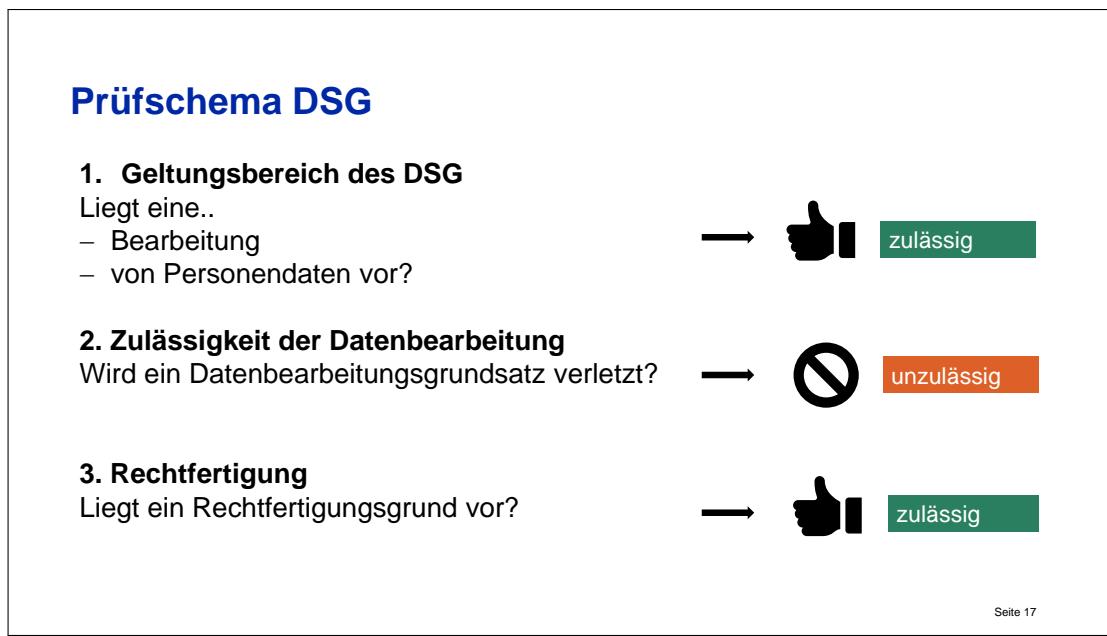
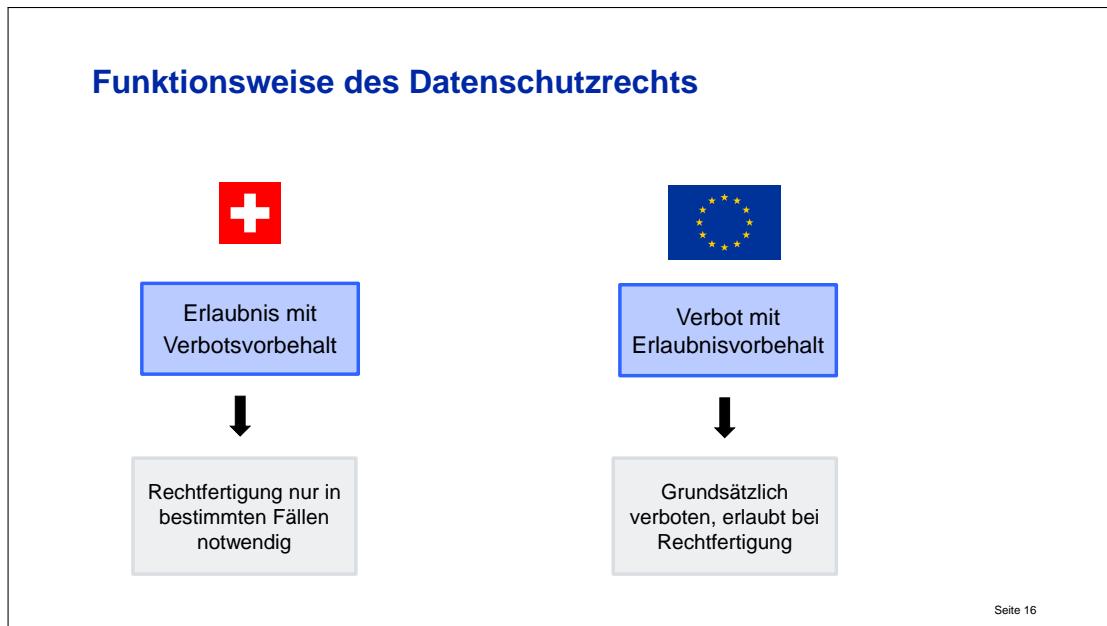
CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 5: Funktionsweise des Datenschutzrechts





Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 6: Datenbearbeitungsgrundsätze

Datenbearbeitungsgrundsätze

Rechtmässigkeit: 6 Abs. 1 DSG I 5 Abs. 1 lit. a DSGVO

Treu und Glauben: 6 Abs. 2 DSG I 5 Abs. 1 lit. a DSGVO

Verhältnismässigkeit: 6 Abs. 2 und 4 DSG I 5 Abs. 1 lit. c und d DSGVO
Datenminimierung und Speicherbegrenzung

Zweckbindung: 6 Abs. 3 DSG I 5 Abs. 1 lit. b DSGVO

Erkennbarkeit: 6 Abs. 3 DSG I 5 Abs. 1 lit. a DSGVO Transparenz

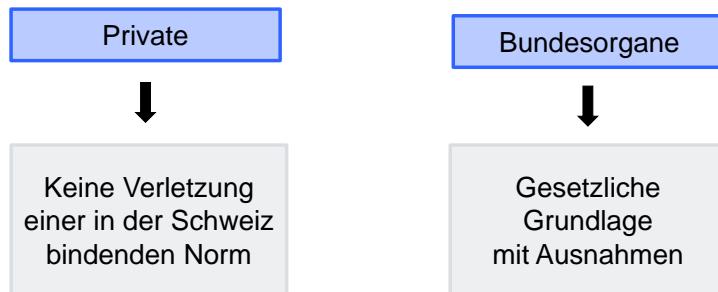
Datenrichtigkeit: 6 Abs. 5 DSG I 5 Abs. 1 lit. d DSGVO

Datenschutz durch Technik: 6 Abs. 1 DSG I 5 Abs. 1 lit. a DSGVO

Datensicherheit: 8 Abs. 1 DSG I 5 Abs. 1 lit. f DSGVO Integrität und Vertraulichkeit

Datenbearbeitungsgrundsätze

1. Rechtmässigkeit



Seite 20

Datenbearbeitungsgrundsätze

2. Verhältnismässigkeitsprinzip

Erforderlichkeit

- Datenbearbeitung nur, wenn und soweit erforderlich
- Datenminimierung / Speicherbegrenzung: Anonymisierung oder Löschung, sobald die Daten nicht mehr erforderlich sind
- Keine Datensammlung auf Vorrat

Seite 21

Datenbearbeitungsgrundsätze

3. Zweckbindung

- Datenbearbeitung nur zum Zweck, der bei der Beschaffung erkennbar war (Primärzweck) oder mit diesem Zweck vereinbar ist (Sekundärzweck)
- Unzulässigkeit der Datenbearbeitung, falls sich der Zweck im Lauf der Zeit geändert hat – Neue Datenbearbeitung
- Unzulässigkeit der Vorratsdatenbeschaffung

Seite 22

Datenbearbeitungsgrundsätze

4. Grundsatz der Erkennbarkeit

- Beschaffung und Zweck der Datenbearbeitung müssen erkennbar sein
- Informationspflicht: Private müssen die betroffene Person über die Datenbearbeitung sowie deren Zweck informieren

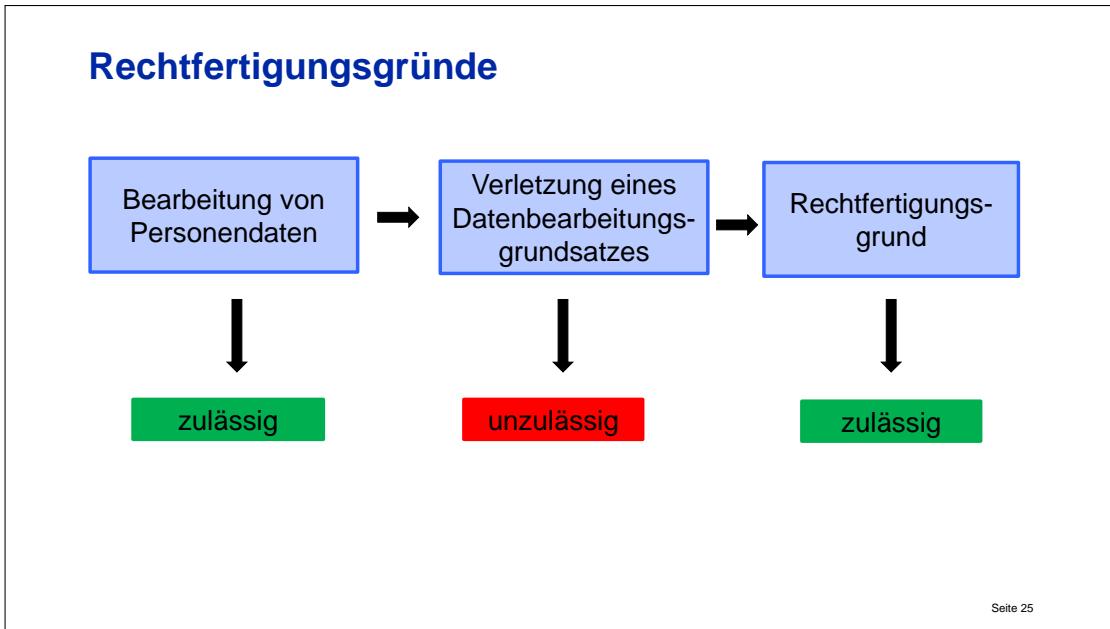
Seite 23

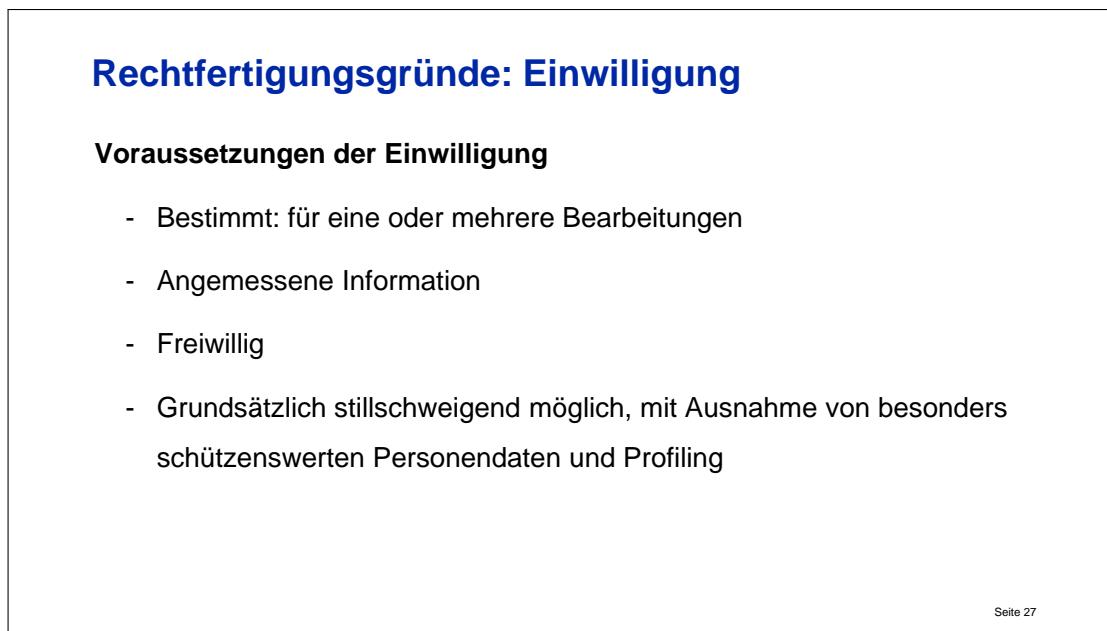
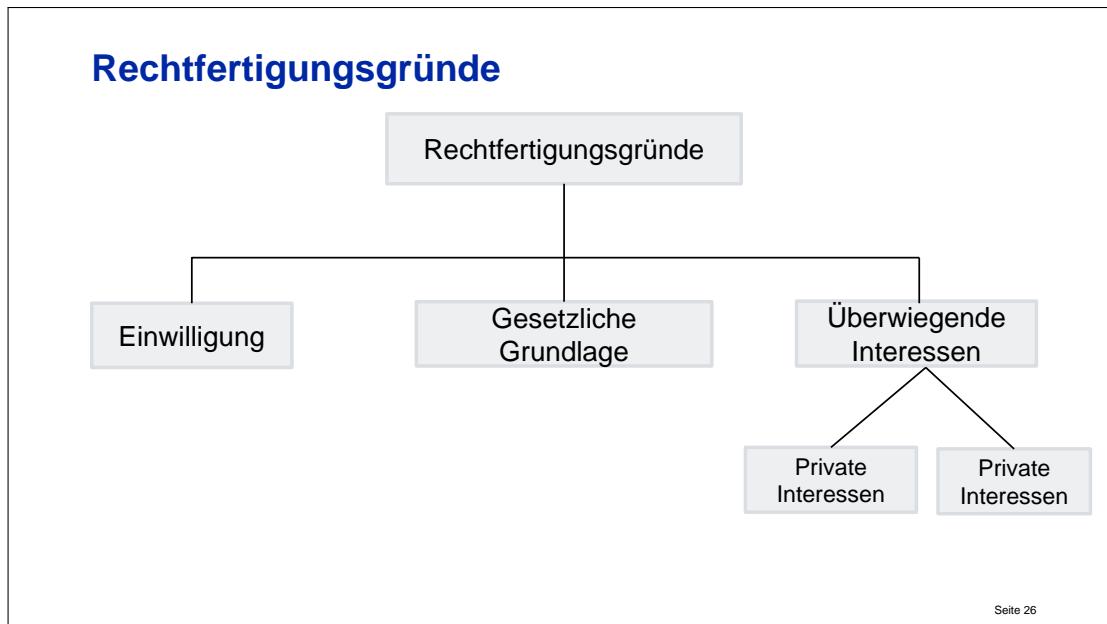
The slide features the logos of the University of Zurich (UZH) and the Center for Information Technology Society and Law (ITSL). The UZH logo is on the left, showing a circular seal with a building and the text "UNIVERSITÄT ZÜRICH". To its right is the ITSL logo, which includes the text "CENTER FOR INFORMATION TECHNOLOGY SOCIETY AND LAW — ITSL" and a graphic of stylized human figures.

Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 7: Rechtfertigungsgründe







Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 8: Rechtsdurchsetzung

Rechtsdurchsetzung

Private – Private (Unternehmen)

Ansprüche

- Klage auf **Unterlassung** der Verletzung – bei drohender Verletzung
- Klage auf **Beseitigung** der Verletzung – bei bestehender Verletzung
- Klage auf **Feststellung** – Subsidiär zur Unterlassungs- und Beseitigungsklage

- Klage auf **Schadenersatz, Genugtuung** oder **Gewinnherausgabe**
- Anspruch auf **Vernichtung** der Daten
- Anspruch auf **Verbot der Bekanntgabe** an Dritte

Seite 29

Rechtsdurchsetzung

Datenschutzrechtliche Ansprüche

- **Recht auf Datenberichtigung:** Berichtigung falscher und Ergänzung unvollständiger Personendaten
- **Auskunftsrecht:** Auskunft darüber, ob Daten über eine betroffene Person bearbeitet werden
- **Recht auf Datenportabilität:** Herausgabe der Daten an die betroffene Person

Seite 30

Rechtsdurchsetzung

Behörden | EDÖB

- **Untersuchung** gegen Bundesorgan oder private Personen
- **Verwaltungsmassnahmen** (insb. Verfügungen): z.B. Unterlassung von Datenbearbeitungen, Vernichten von Personendaten
- **Strafrechtliche Massnahmen:** Bussen bis CHF 250'0000

Seite 31



Universität
Zürich^{UZH}

Digital Society Initiative

CENTER FOR
INFORMATION
TECHNOLOGY
SOCIETY AND
LAW — ITSL



Studium Digitale

Kursbaustein 16: Datenschutzrecht

Lektion 9: Datenschutz als Zuordnungsinstrument

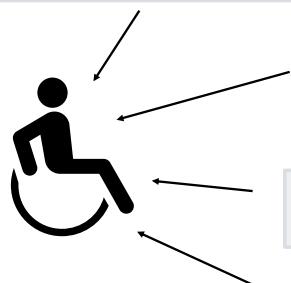
Datenschutz als Zuordnungsinstrument

Rechtliche Zuordnung	Faktische Zuordnung
Sachenrecht	Strafrecht
Patentrecht	Gesetz gegen den unlauteren Wettbewerb
Urheberrecht	
Datenschutzrecht	

Seite 33

Datenschutz als Zuordnungsinstrument

Auskunfts- und Datenportabilität: Zugriff auf von Dritten kontrollierte Personendaten



Einwilligung: Entscheid, wer eigene Daten in welcher Form nutzen darf

Widerrufsrecht: Jederzeitige Möglichkeit Einwilligung zu widerrufen

Widerspruchsrecht: Widerspruch gegen Bearbeitung der Personendaten

Seite 34

Datenschutz als Zuordnungsinstrument

Datenschutzrechtliche Ansprüche können eine gewisse **Zuordnung oder Kontrolle** bewirken

Beschränkungen:

- Erfasst sind nur Daten von natürlichen Personen
- Erfasst nur die jeweils eigenen Personendaten
- Keine Übertragung der datenschutzrechtlichen Ansprüche auf Dritte

Seite 35

12 KI und Machine Learning

DR. SARAH EBLING

Studium Digitale Kursbaustein Künstliche Intelligenz und maschinelles Lernen
Begleit-Skript

Dieses Skript bereitet die Inhalte des Kursbaustein “Künstliche Intelligenz und maschinelles Lernen” schriftlich auf. Die Inhalte sind nicht identisch mit den Videos; sie wurden angepasst für die Lesbarkeit als Dokument.

12.1 Lektion 1: Begrüßung

In der Computerlinguistik geht es um die automatische Verarbeitung natürlicher Sprache. Dazu gehören u.a. die maschinelle Übersetzung (z.B. Google Translate, DeepL) und persönliche Assistenten (z.B. Siri). Maschinelles Lernen und künstliche Intelligenz werden heutzutage viel diskutiert. Der Kursbaustein beschäftigt sich in zwei Teilen mit diesen Themen.

Die Lernziele zur künstlichen Intelligenz sind:

- Sie sind in der Lage, verschiedene Definitionen des Begriffs *künstliche Intelligenz* voneinander abzugrenzen.
- Sie können die wichtigsten Meilensteine der künstlichen Intelligenz darlegen.
- Sie können den Turing-Test erklären.
- Sie können Risiken künstlicher Intelligenz darlegen und Beispiele dazu nennen.

Die Lernziele zum maschinellen Lernen sind:

- Sie sind in der Lage, den Begriff *Maschinelles Lernen* zu definieren.
- Sie können erläutern, welche Aufgaben sich für maschinelles Lernen eignen.
- Sie können zentrale Konzepte des maschinellen Lernens erläutern.
- Sie sind in der Lage, die Begriffe *Maschinelles Lernen*, *Deep Learning* und *(Künstliche) neuronale Netze* voneinander abzugrenzen.

- Sie können die Kernbestandteile von neuronalen Netzen beschreiben.

12.2 Lektion 2: Was ist künstliche Intelligenz?

Nach Poole und Mackworth (2017, S.3) wird künstliche Intelligenz wie folgt definiert: "Artificial Intelligence, or AI, is the field that studies the synthesis and analysis of computational agents that act intelligently." *Computational agents* bezeichnet hier Entitäten, die in einer Umgebung handeln.

Es gab und gibt unterschiedliche Ansätze beim Thema der künstlichen Intelligenz. Einerseits geht es darum, ob man den Unterschied auf Denkprozesse oder das Verhalten einer Maschine legt. Andererseits ist die Frage, wie der Erfolg einer Maschine gemessen wird; dies geschieht entweder anhand der Nähe zur menschlichen oder einer idealtypischen (rationalen) Leistung. Eine Maschine handelt rational, wenn sie das aus Sicht ihres bestehenden Wissens Richtiges tut.

Daraus ergeben sich folgende vier Ansätze der künstlichen Intelligenz:

- Menschlich denken
- Rational denken
- Menschlich handeln
- Rational handeln

Künstliche Intelligenz beruht auf Beiträgen verschiedener Disziplinen:

- Philosophie (Denkfähigkeit von Maschinen)
- Mathematik (Berechenbarkeit, Wahrscheinlichkeiten)
- Wirtschaftswissenschaften (Spieltheorie, Entscheidungstheorie)
- Neurowissenschaften (Informationsverarbeitung des Gehirns)
- Psychologie (menschliches Denken)
- Informatik (Systementwicklung)
- Linguistik (Zusammenhang zwischen Sprache und Denken)

In der Philosophie wird unterschieden zwischen der schwachen und der starken künstlichen Intelligenz (*weak bzw. strong AI*). Bei der ersten tun Maschinen so, als wären sie intelligent, während sie es bei der zweiten tatsächlich sind und ein Bewusstsein haben. Die meisten bisherigen Ansätze konzentrieren sich auf die schwache KI. Die Möglichkeit zu starker KI wird von einigen Wissenschaftlern infrage gestellt. Searle (1980, S.423) schreibt dazu: “No one supposes that computer simulations of a five-alarm fire will burn neighborhood down or that a computer simulation of a rainstorm will leave us all drenched. Why on earth would anyone suppose that a computer simulation of understanding actually understood anything?”

Einführung des Begriffs der künstlichen Intelligenz durch McCarthy et al. (1955, S.2): “We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it.”

Der Begriff *simulate* macht deutlich, dass es hier um schwache KI geht.

12.3 Lektion 3: Meilensteine der künstlichen Intelligenz

Wichtige Meilensteine im Zusammenhang mit der künstlichen Intelligenz sind:

- 1943: erstes Modell künstlicher Neuronen (McColluch und Pitts)
- 1950: Turing-Test
- 1955: Begriff *artificial intelligence*
- 1958: Perceptron (Rosenblatt)
- 1960er-/1970er-Jahre: Systeme für Sprachverstehen
- 1970er-/1980er-Jahre: Experten-Systeme
- 1974: erster KI-Winter
- 1980er-Jahre: erste erfolgreiche kommerzielle Systeme
- 1987-1993: zweiter KI-Winter
- 1990er-Jahre: Aufschwung

Eines dieser ersten Sprachverständssysteme war CHAT-80, das Fragen zur Geographie (Grenzen, Hauptstädte, Lage von Ozeanen...) beantworten konnte. Es basierte auf einer Datenbank, die viele Fakten speicherte und darauf zurückgriff. Wie alle Sprachverständssysteme dieser Zeit war es in Vokabular und Satzstrukturen stark beschränkt.

Wichtige Teilbereiche der künstlichen Intelligenz sind:

- Automatische Sprachverarbeitung
- Wissensrepräsentation
- *Reasoning*
- Maschinelles Lernen
- *Computer vision*
- Robotik

Diese Bereiche sind nicht gleichwertig. Das maschinelle Lernen kommt z.B. in den anderen Teilbereichen vor.

12.4 Lektion 4: Wissen, Schlussfolgern, Planen

Wissen wird in der KI typischerweise in Wissensbasen (Knowledge Bases) gespeichert. Diese bestehen aus Fakten und Regeln. Aus Fakten und Regeln lassen sich Schlussfolgerungen treffen. Mit einer Inferenzmaschine oder einem automatischen Algorithmus lassen sich Fakten und Regeln verwenden, um Abfragen zu beantworten. Je mehr Fakten und Regeln in einer Wissensbasis enthalten sind, desto besser können auch die Abfragen beantwortet werden.

Früher musste Wissen mühsam zusammengetragen werden. Dies änderte sich mit dem Internet. Im Jahr 2007 wurden die Tabellen von DBpedia mit der Datenbank Geo-names zusammengeführt. Diese Datenbasis ist über die Jahre stark gewachsen und enthält heute hunderte von Datensätzen und Milliarden von Aussagen. Die so zu findenden Informationen sind allerdings nicht immer korrekt.

Wissen und automatisches Schlussfolgern sind grundlegende Bausteine der künstlichen Intelligenz. Sie kommen auch heute noch stark zum Einsatz.

Fakten	Regeln
Robin ist ein Rotkehlchen	?X ist eine Art ?Y &
Rotkehlchen ist eine Art Vogel	?Y ist eine Art ?Z
Vogel ist eine Art Tier	-> ?X ist eine Art ?Z
Tier ist eine Art Lebewesen	
Vogel bewegt sich Fliegen	?X ist ein ?Y & ?Y ist eine Art ?Z
Flix ist ein Fledermaus	-> ?X ist ein ?Z
Fledermaus bewegt sich Fliegen	?X bewegt sich Fliegen
Flugtier ist eine Art Tier	-> ?X ist eine Art Flugtier
Pingu ist ein Pinguin	...
Pinguin ist eine Art Vogel	
...	

Tabelle 8: Beispiel einer Wissensbasis

12.5 Lektion 5: Prototypische KI-Anwendungen

Die Objekterkennung ist Teil der *Computer vision* (maschinelles Sehen). Dabei sollen Objekte in einem Bild erkannt und mit einem entsprechenden Label versehen werden. Beispiele auf dem Bild in der Präsentation sind Objekte im Straßenverkehr (Autos, Fahrräder, Ampeln...).

Beim *Image captioning* sollen nicht mehr nur einzelne Objekte erkannt, sondern das gesamte Bild beschrieben werden (z.B. "man in black shirt is playing guitar"). Hier ist wichtig, dass ein zusammenhängender Text in Form einer Phrase generiert wird.

Bei der Spracherkennung werden die Schallwellen der gesprochenen Sprache aufgezeichnet und z.B. mithilfe eines neuronalen Netzes in geschriebene Sprache umgewandelt.

Künstliche Intelligenz wird auch in der medizinischen Diagnostik genutzt, um Hinweise auf Krankheiten, Tests und Behandlungsformen zu erhalten. Aus dem Alltag bekannt sind Empfehlungssysteme, bei denen man zum Beispiel aufgrund der ausgewählten Antworten auf die Fragen eine Empfehlung erhält.

Computational creativity stellt synthetisch Musik, bildende Kunst und Literatur her.

12.6 Lektion 6: Der Turing-Test

In Turings *Imitation Game* hat ein menschlicher Fragesteller/eine menschliche Fragestellerin die Aufgabe, einem Menschen und einer Maschine Fragen zu stellen und am Ende der Unterhaltung zu entscheiden, wer der Mensch und wer die Maschine ist. Die Maschine besteht den Turing-Test, wenn es dem Fragesteller/der Fragestellerin nicht gelingt, seine/ihre beiden Unterhaltungspartner zu unterscheiden.

Beim Turing-Test handelt es sich um einen Verhaltenstest auf Intelligenz. Damit entspricht er dem Ansatz, dass KI menschlich handeln soll.

Dem Turing-Test liegt ein Dialogsystem, eine Anwendung der automatischen Sprachverarbeitung, zugrunde. Dabei handelt sich um ein System, bei dem ein Mensch einen Dialog mit einer Maschine führt.

Beim *total Turing test* werden die Fertigkeiten der Maschine auf Sehen und Greifen, also *Computer vision* und Robotik, ausgeweitet.

12.7 Lektion 7: Dialogsysteme, Teil I

Das erste Dialogsystem war ELIZA, welches 1966 am Massachusetts Institute of Technology (MIT) entwickelt wurde. ELIZA wurde als maschinelle Psychotherapeutin eingesetzt.

Typisch für Dialogsysteme der damaligen Zeit war, dass sie einen Teil der Eingabe des Benutzers nahmen und zurückspielten.

Weitere Meilensteine im Bereich Dialogsysteme:

- 1991: Einführung Loebner-Preis
- 2006: Watson (IBM)
- 2011: Siri (Apple)
- 2012: Google Now
- 2014: Cortana (Microsoft), Alexa (Amazon), Eugene Goostman
- 2016: Google Assistant, Jill Watson
- 2018: Google Duplex

Es hat noch kein Dialogsystem wirklich den Turing-Test bestanden.

12.8 Lektion 8: Dialogsysteme, Teil II

Das Dialogsystem Jill Watson ist eine maschinelle Tutorin in einem Online-Kurs zum Thema künstliche Intelligenz am Georgia Institute of Technology. Zusammen mit menschlichen Tutoren betreut Jill Watson Studierende bei formalen Anliegen.

Die Herausforderungen heutiger Dialogsysteme hängen damit zusammen, dass sie auf maschinellem Lernen basieren. Sie konsultieren grosse Datenmengen. Die häufigste Antwort ist oft eine sehr unspezifische Aussage wie z.B. "I don't know". Eine weitere Herausforderung ist, dass Dialogsysteme Daten aus sehr unterschiedlichen Quellen verwenden, was zu einem Phänomen namens Sprecherinkonsistenz führen kann. Dabei gibt das System je nachdem, wie eine Frage formuliert ist, unterschiedliche Antworten.

12.9 Lesson 9: Introduction to Robotics

The Oxford Dictionary defines robotics as follows: “The branch of technology that deals with the design, construction, operation and application of robots.”

According to their working environment, robots are grouped into different types:

- Aerial (e.g. drones and airplanes)
- Grounded
- Water
- Manipulators (e.g. robots in industries)

Important historical milestones are:

1495: Da Vinci’s Knight

1700: The Century of Mechanical Automata

1920: First Use of Term Robot

1938: The First Manipulator

1984: CMU NavLab

1989: Genghis

1991: Honda P1

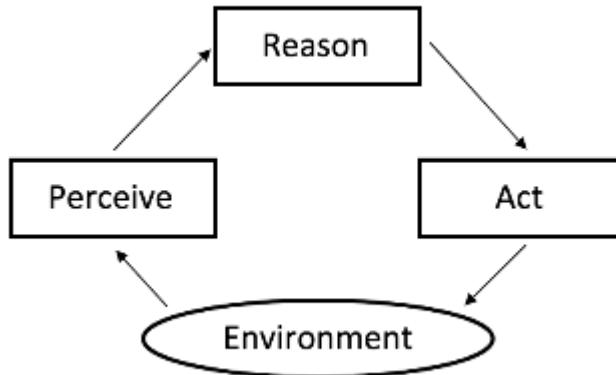
1996: Robo-Tuna

2001: First Aerial Robot

Robots work based on the Perception-Action Cycle: Their perception of the environment leads to a reasoning determining what course of action to pursue. This cycle will continue until the mission is accomplished.

In classical robotics, the Perception-Action Cycle is based on if-else-conditions. For example, “if you see an obstacle, turn right or left. If not, continue forward.”

In modern robotics, the Perception-Action Cycle is learned through interaction with the physical world. To manage to huge amount of data generated in this process, artificial intelligence is used.



Environment leads to Perceive leads to Reason leads to Act leads to Environment etc.

12.10 Lektion 10: Auswirkungen der künstlichen Intelligenz

Man geht davon aus, dass künstliche Intelligenz u.a. ethische, psychologische, soziale, ökonomische und rechtliche Auswirkungen hat.

Die künstliche Intelligenz hat auch Risiken, z.B.:

- Verlust von Arbeitsplätzen
- Verwendung von künstlicher Intelligenz für unbeabsichtigte oder unerwünschte Zwecke
- Frage oder Rechenschaft

Da Systeme aus vorhandenen Daten lernen, kann es je nach Zusammensetzung der Daten sein, dass sie Merkmale lernen, die wir mit gesellschaftlichen Stereotypen verbinden. Das nennt man einen *Bias*.

12.11 Lektion 11: Was ist maschinelles Lernen?

Mitchell (1997, S.2) definiert maschinelles Lernen folgendermassen: “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E.”

Eine Übersetzungsaufgabe könnte gelöst werden, indem man ein zweisprachiges Wörterbuch konsultiert, Grammatikregeln der Quell- und Zielsprache hinzunimmt und Regeln hinzuzieht, wie die Grammatik der Quellsprache in diejenige der Zielsprache übertragen werden muss. Das nennt man ein *regelbasierter Vorgehen*. Dieses hat diverse Nachteile wie z.B. die Schwierigkeit, ein System auf neue Kontexte anzupassen. Je nach Problemstellung kann es auch eine Herausforderung sein, überhaupt Regeln zu definieren.

Beim maschinellen Lernen wird von bereits erarbeiteten Lösungen ausgegangen, im Beispiel von Übersetzungen, und daraus gelernt, ähnliche Problemstellungen zu bewältigen.

12.12 Lektion 12: Zentrale Konzepte maschinellen Lernens, Teil I

Aufgrund vorhandener Daten lernt das System. Die Daten werden in drei Sets unterteilt:

- Trainingsset (*training set*), etwa 80% der Daten
- Entwicklungsset (*validation/development set*), etwa 10% der Daten
- Testset (*test/evaluation set*), etwa 10% der Daten

Experience E

Beim Lernen kommt es darauf an, wie auf die Daten zugegriffen wird. Dafür gibt es diverse Ansätze, die hier allerdings auf supervisierte (überwachtes) und unvsupervierte Lernen beschränkt sind. Beim supervisierten Lernen werden Labels verwendet. Beim unvsupervierten Lernen gibt es keine Labels und es wird einzig aufgrund der Datenstruktur der Beispiele etwas über die Daten gelernt.

Tasks lassen sich in Klassen zusammenfassen. Beispiele für solche Klassen sind Regression und Klassifikation.

Ein Beispiel für die Regression ist, den Preises eines neuen Hauses in einem bestimmten Gebiet bei bekannter Hausgröße vorherzusagen. Dafür schaut man sich bestehende Häuser, deren Größe und Preis an. Aus diesen Informationen lässt sich eine lineare Regressionsfunktion $\hat{y} = wx + b$

instanziieren. Das Ziel des Lernvorgangs ist das Bestimmen der Parameter w und b.

Das Ziel beim Training ist, einen minimalen Verlust zu erreichen, wenn wir \hat{y} vorhersagen, während die echte Ausgabe y ist. Um dies zu modellieren, gibt es die Verlustfunktion (*loss function*). Ein Beispiel für eine Verlustfunktion ist der *mean squared error* (MSE) (mittlerer quadratischer Fehler). Dieser lässt sich wie folgt berechnen: $\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$

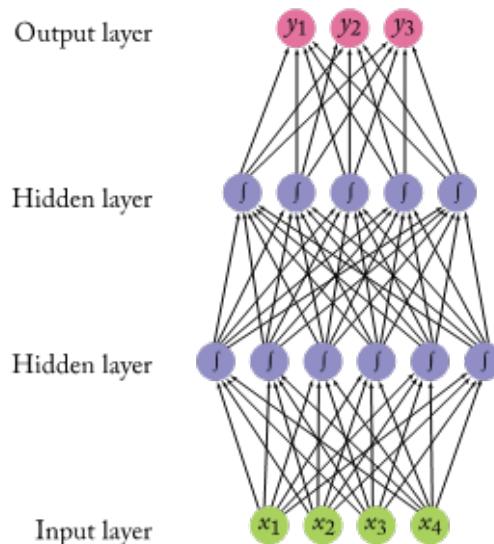
Um die Performanz des Lernalgorithmus auf ungesiehenen Daten zu beurteilen, wird er dann an einem Testset erprobt.

Bei der Klassifikation wird die logistische Regression verwendet. Dabei handelt es sich um eine lineare Regression mit zusätzlicher Sigmoid-Funktion:

$$\hat{y} = \sigma(w^\top x + b)$$

Beim unsuperviseden Lernen gibt es keine Labels wie den Preis y , sondern aus den Daten soll etwas über ihre Struktur gelernt werden. Das kann z.B. durch Clustering geschehen. Dabei wird eine Menge von Elementen in Gruppen eingeteilt, sodass die Ähnlichkeit zwischen Elementen desselben Clusters maximal und zwischen solchen unterschiedlicher Cluster minimal ist.

12.13 Lektion 14: Deep Learning und künstliche neuronale Netze



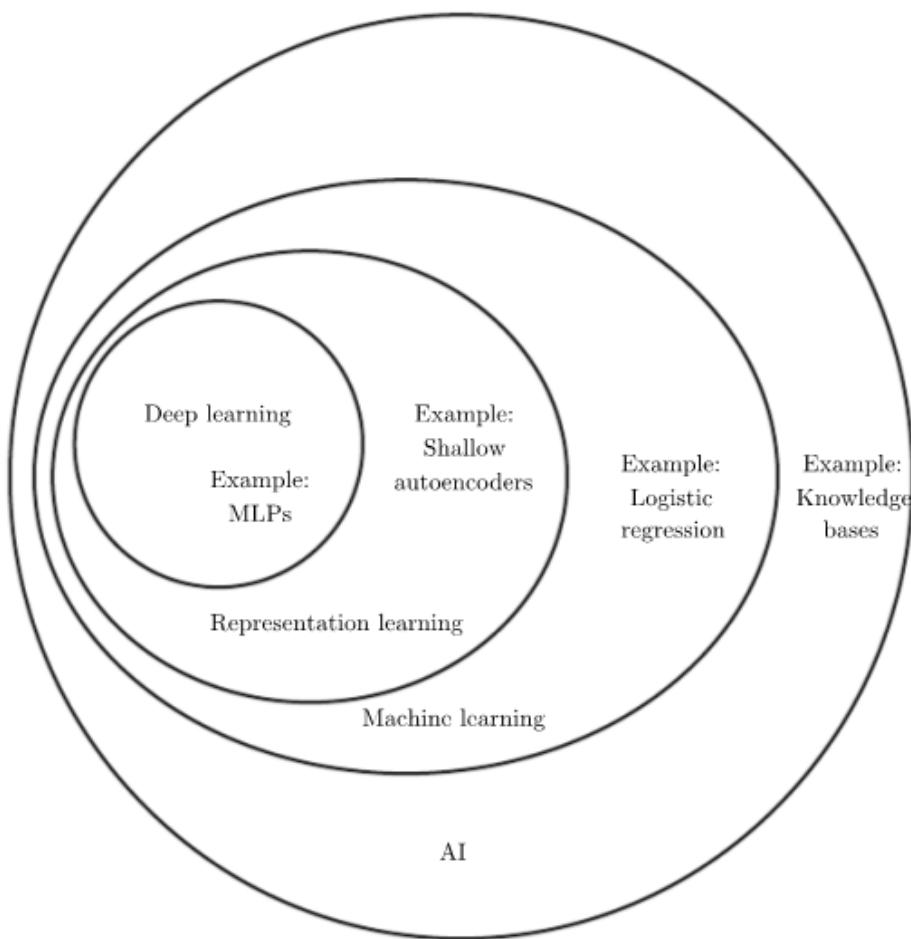
Das neuronale Netz ist in Schichten (*layers*) angeordnet. Es besitzt einen Eingabe- und einen Ausgabe-Layer sowie zwei *hidden layer*. Jeder Layer

besitzt eine bestimmte Anzahl Neuronen, das sind Platzhalter für einzelne Zahlen. Die Anzahl *Layer* und Anzahl Neuronen pro *Layer* sind Beispiele für Hyperparameter, die über ein Entwicklungsset festgelegt werden. Parameter werden im Gegensatz dazu während des Trainingsvorgangs gelernt.

Das abgebildete neuronale Netz beschreibt die folgende Funktion:

$\text{NN}(x) = W^3(g^2(W^2(g^1(W^1x + b^1)) + b^2))$ Ein solches neuronales Netz könnte bspw. Zeitungsartikel sortieren. Dabei gäbe es drei Ausgabeklassen, da der Ausgabe-Layer drei Neuronen hat.

Dieses neuronale Netz ist ein *deep neural net*. Davon spricht man, wenn ein neurales Netz mehr als einen *hidden layer* hat. Solche Netze werden im Rahmen von *deep learning* gelernt.



Die Abbildung zeigt, dass *deep learning* ein Teil des maschinellen Lernes ist. Dieses wiederum ist ein Teilgebiet der künstlichen Intelligenz.

Zu den Herausforderungen im Zusammenhang mit neuronalen Netzen gehören:

- grosse benötigte Datenmengen
- Interpretierbarkeit
- Robustheit

12.14 Lektion 15: Abschluss

Die künstliche Intelligenz beschäftigt sich mit dem Verstehen und Entwickeln von intelligent handelnden Maschinen. Die Ansätze künstlicher Intelligenz sind dabei:

- Menschlich denken
- Rational denken
- Menschlich handeln (Turing-Test)
- Rational handeln

Künstliche Intelligenz wird in schwache und starke KI unterteilt. Geprägt wurde sie durch die schwache KI.

Ein Teilbereich der künstlichen Intelligenz ist die automatische Sprachverarbeitung. Prototypische Anwendungen davon sind die maschinelle Übersetzung, die Spracherkennung, *Image captioning* und Dialogsysteme. Dialogsysteme können auf Text - wie beim Turing-Test - oder gesprochener Sprache - wie bei persönlichen Assistenten - basiert sein.

Im Bereich der *Computer vision* sind prototypische Anwendungen die Objekterkennung und *Image captioning*.

Die behandelten Bestandteile des maschinellen Lernens sind:

- *Experience E*: z.B. supervisiert, unsupervisiert
- *Task T*: z.B. Regression (lineare Regression), Klassifikation (logistische Regression), Clustering
- *Performance measure P*: z.B. *Mean squared error*

Künstliche neuronale Netze sind komplexe nicht-lineare Funktionen, die aus kleineren Einheiten zusammengesetzt sind.

Deep learning ist ein Teil des maschinellen Lernens und dieses wiederum ein Teil der künstlichen Intelligenz.

12.15 Folien



Universität
Zürich^{UZH}

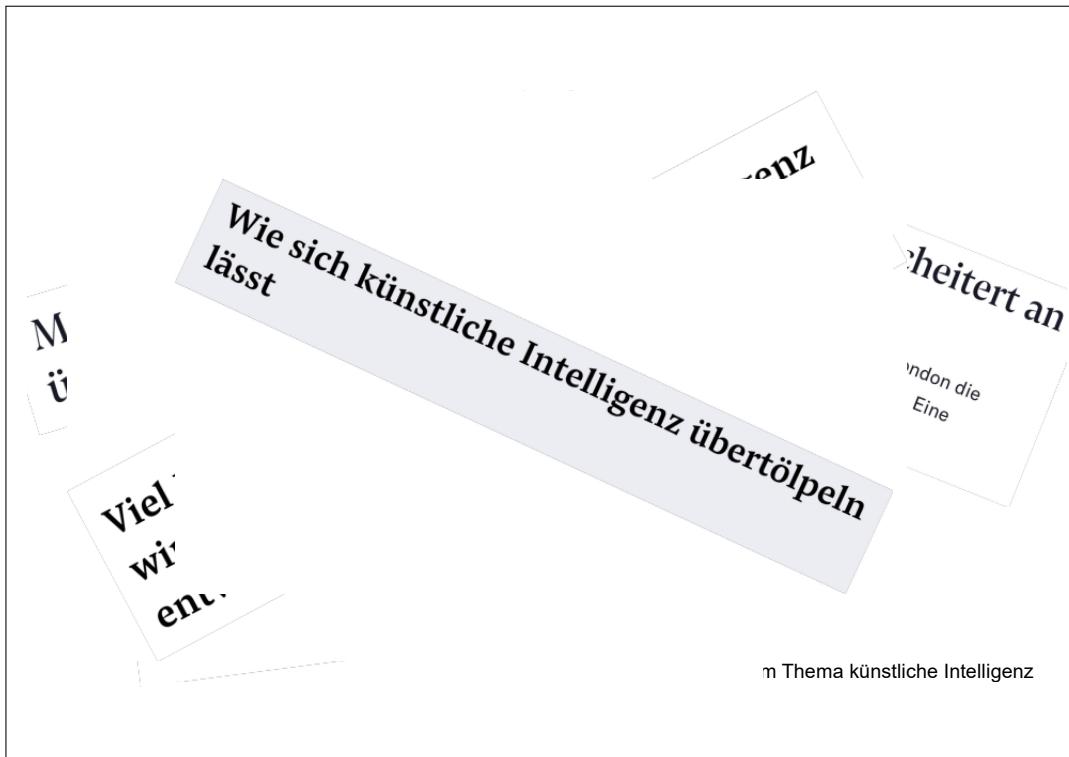
Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und
maschinelles Lernen

Lektion 1: Begrüssung



Künstliche Intelligenz: Lernziele

- Sie sind in der Lage, verschiedene Definitionen des Begriffs *Künstliche Intelligenz* voneinander abzugrenzen.
- Sie können die wichtigsten Meilensteine der künstlichen Intelligenz darlegen.
- Sie können den Turing-Test erläutern.
- Sie können Risiken künstlicher Intelligenz darlegen und Beispiele dazu aus der Praxis nennen.

Maschinelles Lernen: Lernziele

- Sie sind in der Lage, den Begriff *Maschinelles Lernen* zu definieren.
- Sie können erläutern, welche Aufgaben sich für maschinelles Lernen eignen.
- Sie können zentrale Konzepte des maschinellen Lernens erläutern.
- Sie sind in der Lage, die Begriffe *Maschinelles Lernen*, *Deep learning* und *(Künstliche) neuronale Netze* voneinander abzugrenzen.
- Sie können die Kernbestandteile von neuronalen Netzen beschreiben.



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und
maschinelles Lernen

Lektion 2: Was ist künstliche Intelligenz?

«Artificial intelligence, or AI, is the field that studies the **synthesis** and **analysis** of computational agents that act intelligently.» (Poole und Mackworth 2017, S. 3)

Ansätze künstlicher Intelligenz

- Menschlich denken
 - Rational denken
 - Menschlich handeln
 - Rational handeln
- (Russell und Norvig 2016)

Grundlagen der künstlichen Intelligenz

- Philosophie
- Mathematik
- Wirtschaft
- Neurowissenschaft
- Psychologie
- Informatik
- Linguistik
- ...

(Russell und Norvig 2016)

Arten künstlicher Intelligenz

- Schwache KI (*weak AI*)
- Starke KI (*strong AI*)

«No one supposes that computer simulations of a five-alarm fire will burn the neighborhood down or that a computer simulation of a rainstorm will leave us all drenched. Why on earth would anyone suppose that a **computer simulation of understanding actually understood anything?**» (Searle 1980, S. 423)

Dartmouth-Workshop

«We propose that a 2 month, 10 man study of **artificial intelligence** be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to **simulate** it»
(McCarthy u. a. 1955, S. 2)

Referenzen

- McCarthy, J. u. a. (1955). *Proposal for the Dartmouth summer research project on artificial intelligence*. Techn. Ber. Dartmouth College.
- Poole, David L. und Alan K. Mackworth (2017). *Artificial Intelligence: Foundations of Computational Agents*. 2nd. New York, NY, USA: Cambridge University Press. ISBN: 110719539X, 9781107195394.
- Russell, Stuart und Peter Norvig (2016). *Artificial Intelligence: A Modern Approach*. Pearson Education Limited.
- Searle, John (1980). "Minds, Brains, and Programs". In: *Behavioral and Brain Sciences* 3, S. 417–424.

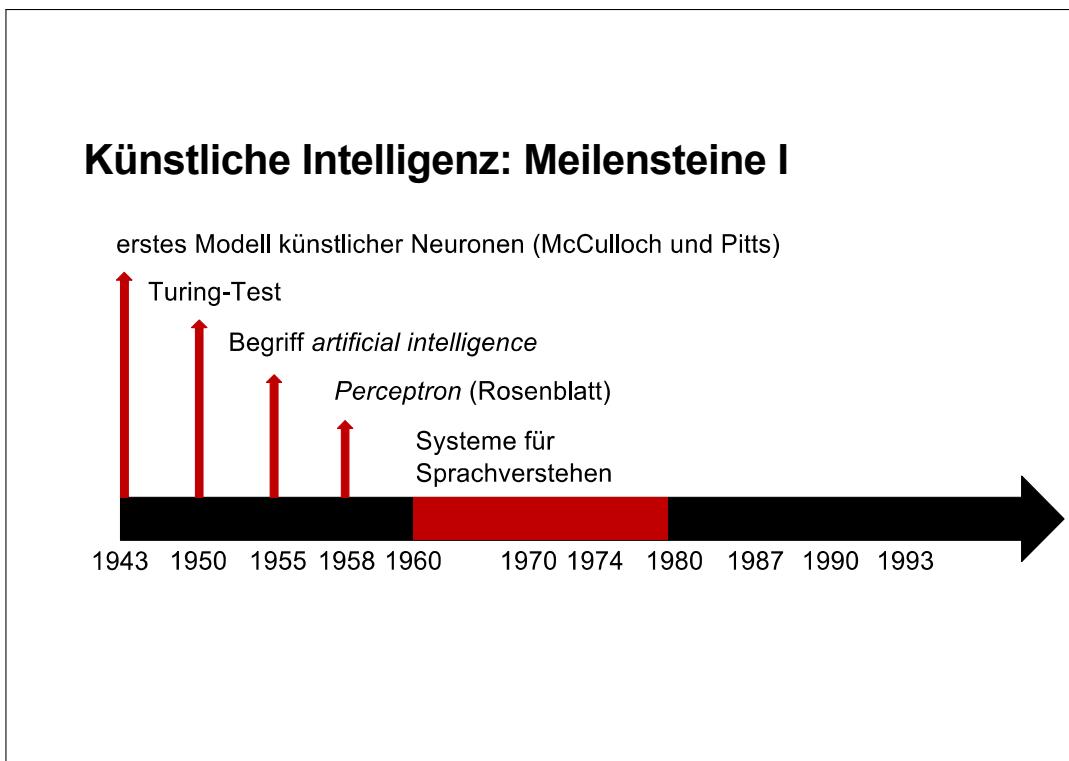


**Universität
Zürich^{UZH}**
Digital Society Initiative

Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

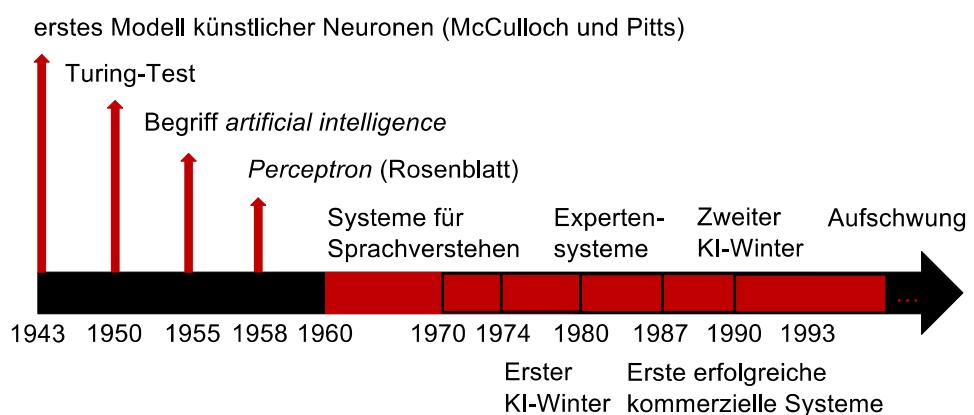
Lektion 3: Meilensteine der künstlichen Intelligenz



CHAT-80 (Warren und Pereira 1982)

Does Afghanistan border China?
 which country's capital is Ouagadougou?
 which is the ocean that borders African countries and that borders Asian countries?
 what is the capital of each country bordering the Baltic?
 what are the latitudes of the countries north of the United Kingdom?
 (Warren und Pereira 1982, S. 120)

Künstliche Intelligenz: Meilensteine II



Künstliche Intelligenz: Teilbereiche

- Automatische Sprachverarbeitung
- Wissensrepräsentation
- *Reasoning*
- Maschinelles Lernen
- *Computer vision*
- Robotik
- ...

(Russell und Norvig 2016)

Referenzen

- Russell, Stuart und Peter Norvig (2016). *Artificial Intelligence: A Modern Approach*. Pearson Education Limited.
- Warren, David H. D. und Fernando C. N. Pereira (Juli 1982). “An Efficient Easily Adaptable System for Interpreting Natural Language Queries”. In: *Comput. Linguist.* 8.3–4, 110–122. ISSN: 0891-2017.



**Universität
Zürich**
UZH
Digital Society Initiative



Studium Digitale

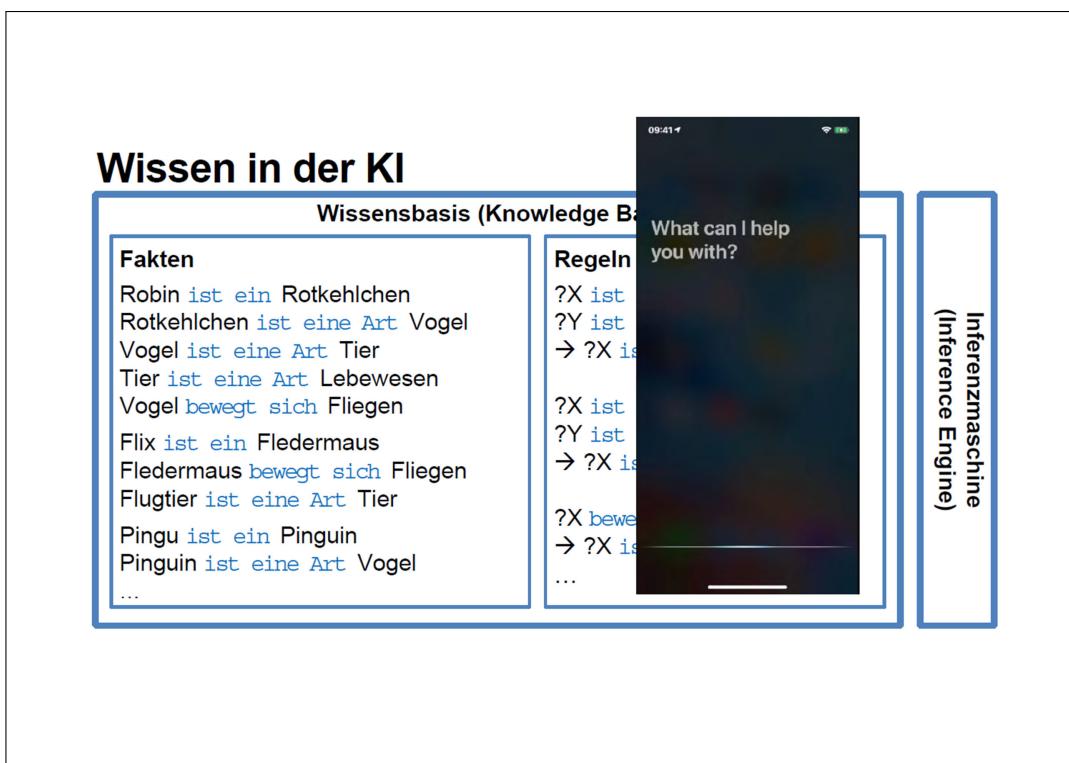
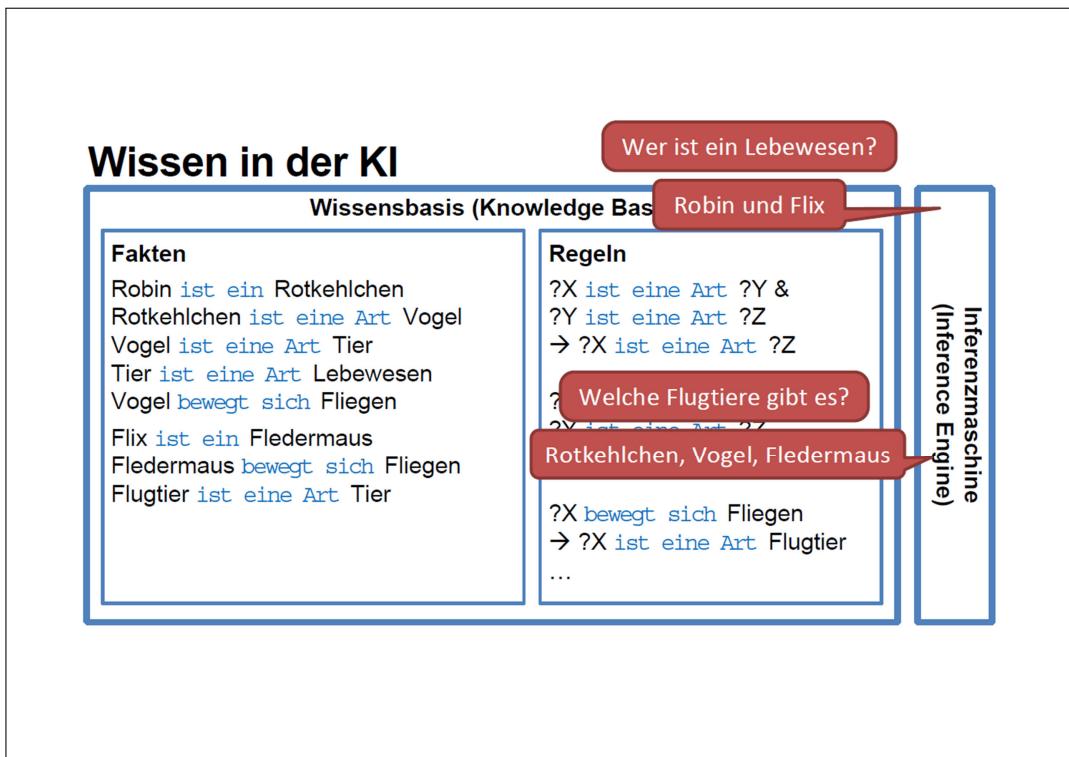
Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

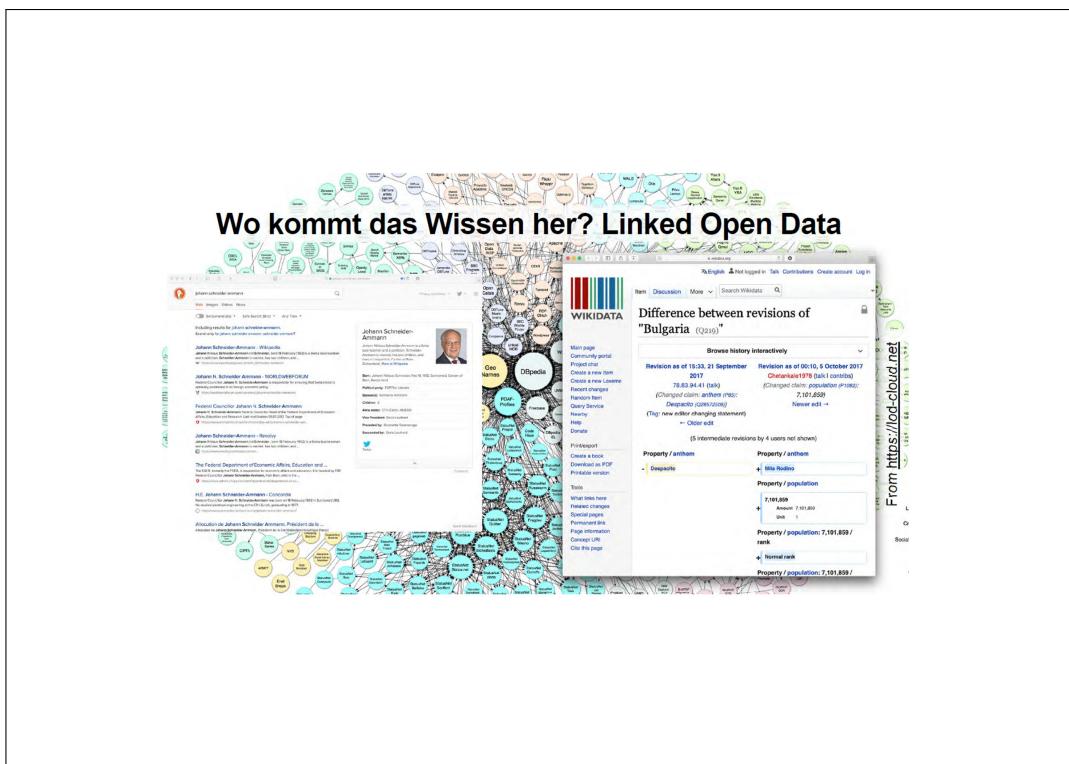
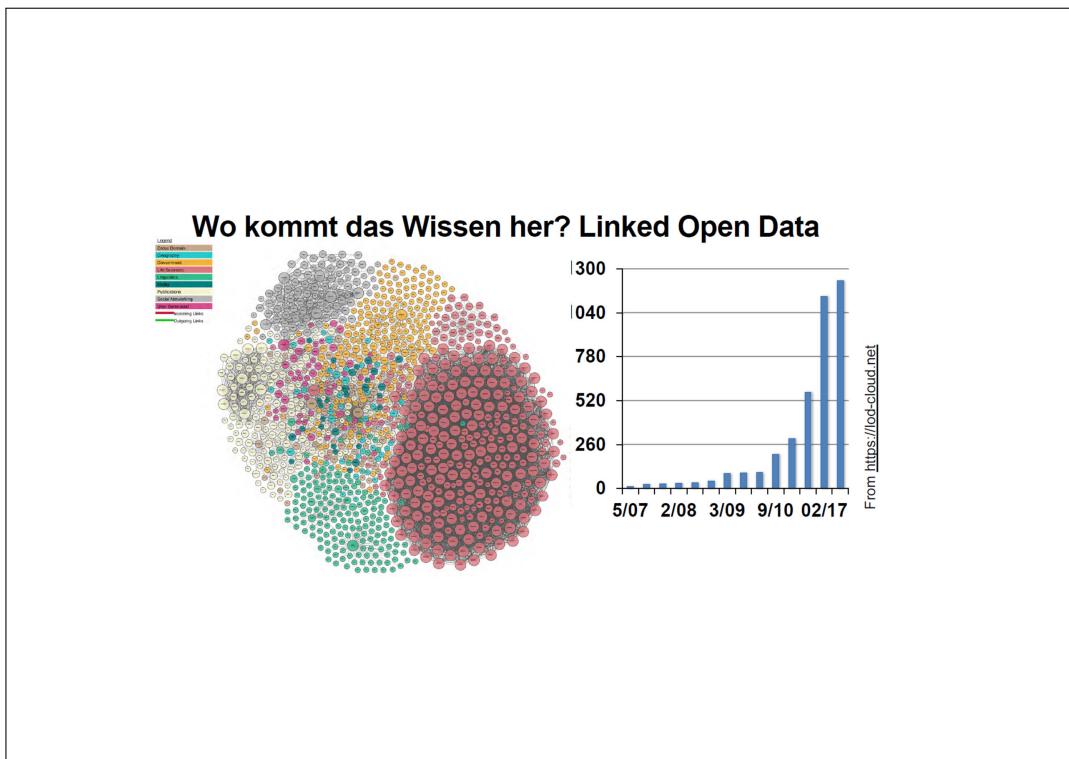
Lektion 4: Wissen und Schlussfolgern

Wissen in der KI

Wissensbasis (Knowledge Base)

Fakten Robin ist ein Rotkehlchen Rotkehlchen ist eine Art Vogel Vogel ist eine Art Tier Tier ist eine Art Lebewesen Vogel bewegt sich Fliegen	Regeln $?X \text{ ist eine Art } ?Y \&$ $?Y \text{ ist eine Art } ?Z$ $\rightarrow ?X \text{ ist eine Art } ?Z$ Was ist Robin? Robin ist ein Rotkehlchen Was ist ein Rotkehlchen? Ein Rotkehlchen ist ein Vogel, ein Tier und ein Lebewesen	Inferenzmaschine (Inference Engine)
---	---	--





Referenzen

Inhalte

- Linked Open Data Cloud, <https://lod-cloud.net>



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und
maschinelles Lernen

Lektion 5: Prototypische KI-Anwendungen

Objekterkennung

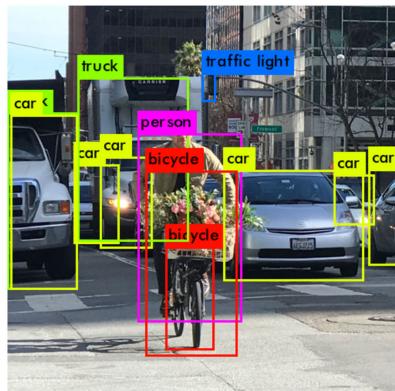


Abbildung: <https://towardsdatascience.com/r-cnn-3a9beddf55a>

Image captioning



"man in black shirt is playing guitar."



"construction worker in orange safety vest is working on road."



"two young girls are playing with lego toy."

Abbildung: <https://towardsdatascience.com/image-captioning-in-deep-learning-9cd23fb4d8d2>

Spracherkennung

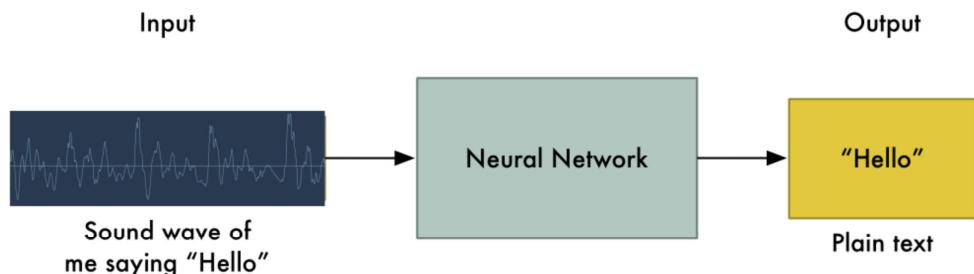


Abbildung: <https://medium.com/@ageitgey/>

Übersetzung

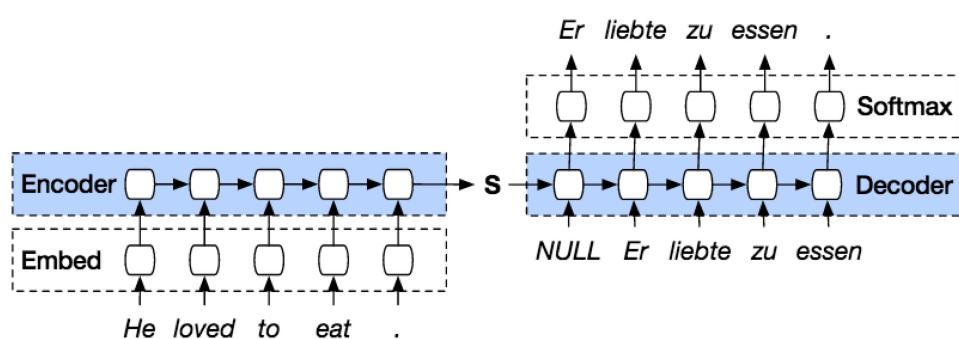


Abbildung: <https://medium.com/@gautam.karmakar/>

Medizinische Diagnostik

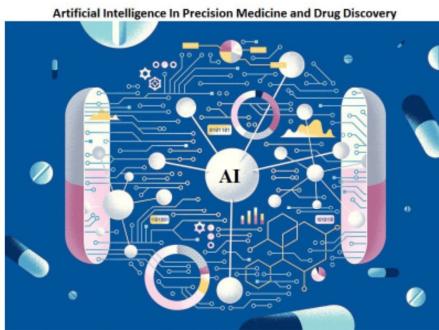


Abbildung: <https://medium.com/@mahajanprachi256/>

Empfehlungssysteme

A screenshot of a web-based recommendation system. On the left, there is a survey section with four questions and radio button options: "Would you like to write handwritten notes on your screen?", "Do you have a desktop PC at home?", "Do you have an external monitor at home for use with your laptop or are planning to buy one?", and "Will you move abroad in the foreseeable future or do a semester abroad?". The last question has "Yes" selected. Below the survey is a green "Find my Device!" button. On the right, a blue header bar says "NILA recommends for you:". Below it are two product cards: "MacBook Pro 15" Touch Bar (Mid 2018)" and "MacBook Pro 15" Touch Bar High-End (Mid 2018)". Each card has a "Give Feedback" button and a "See Details" link.

Abbildung: <https://nila-1.projektneptun.ch/advisor/alpha/2>

Computational creativity



Abbildung: <https://www.ibm.com/blogs/research/2018/10/ai-creativity/>



**Universität
Zürich UZH**

Digital Society Initiative



Studium Digitale

**Kursbaustein 10: Künstliche Intelligenz und
maschinelles Lernen**

Lektion 6: Der Turing-Test

Turings «Imitation Game»

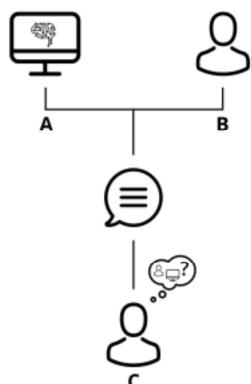


Abbildung: Turing-Test
(<https://medium.com/@vanessa.friebe/>)

Turing-Test: Beispieldialog

HUMAN: In the first line of your sonnet which reads "shall I compare thee to a summer's day," would not a "spring day" do as well or better?

MACHINE: It wouldn't scan.

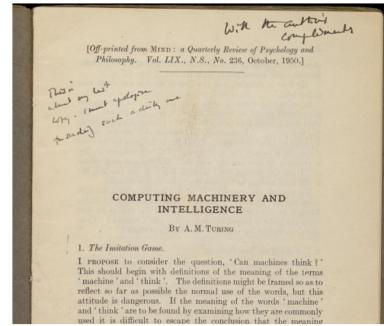
HUMAN: How about "a winter's day." That would scan all right.

MACHINE: Yes, but nobody wants to be compared to a winter's day.

(Turing 1950, S. 443f.)

Turing-Test und Dialogsysteme

- Operationale Definition von künstlicher Intelligenz
- Textbasiertes Dialogsystem
- *Total Turing test*: mit *Computer vision* und Robotik



Referenzen

Turing, A. M. (1950). "Computing Machinery and Intelligence". In: *Mind* 59, S. 433–460.



**Universität
Zürich^{UZH}**
Digital Society Initiative



Studium Digitale

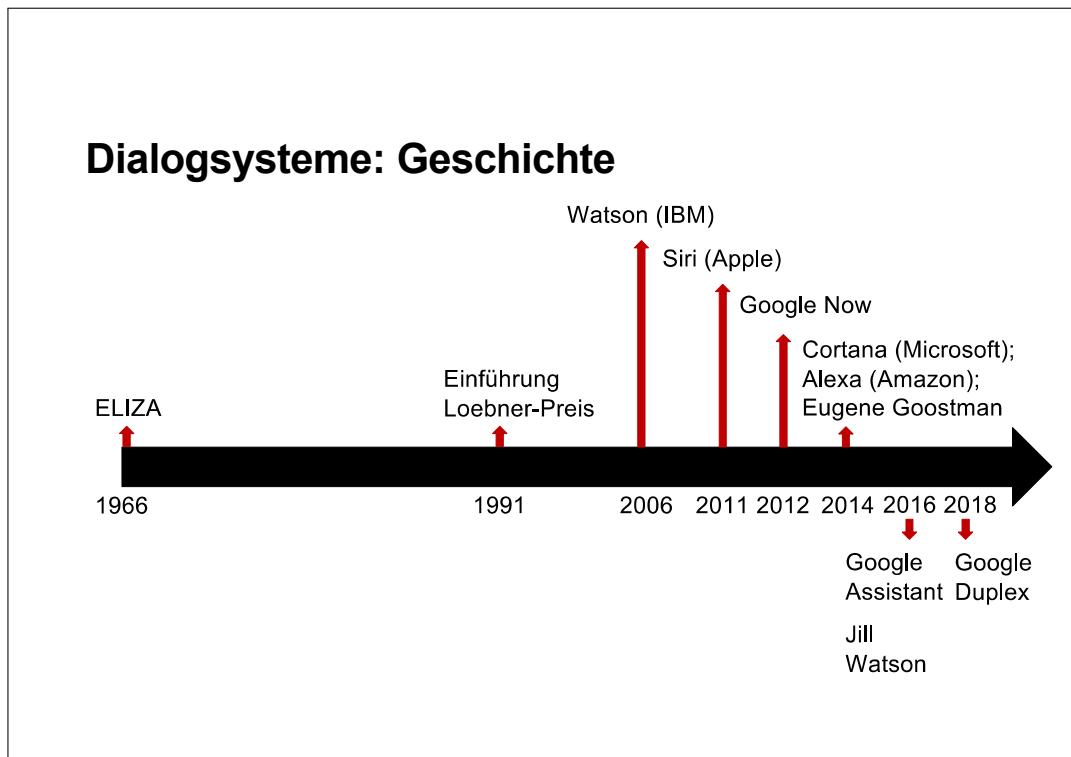
Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

Lektion 7: Dialogsysteme, Teil I

```
Welcome to
      EEEEEE  LL      IIII  ZZZZZZ  AAAAAA
      EE      LL      II     ZZ  AA  AA
      EEEEEE  LL      II     ZZZ  AAAAAAA
      EE      LL      II     ZZ  AA  AA
      EEEEEE  LLLLLL  IIII  ZZZZZZ  AA  AA

Eliza is a mock Rogerian psychotherapist.
The original program was described by Joseph Weizenbaum in 1966.
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?
YOU: Men are all alike.
ELIZA: What is the connection, do you suppose ?
YOU: They're always bugging us about something or other.
ELIZA: Can you think of a specific example ?
YOU: Well, my boyfriend made me come here.
ELIZA: Is it important to you that your boyfriend made you come here ?
YOU: He says I'm depressed much of the time.
ELIZA: I am sorry to hear that you are depressed.
YOU: It's true. I am unhappy.
ELIZA: Can you explain what made you unhappy ?
YOU: ■
```



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

Lektion 8: Dialogsysteme, Teil II

«Jill Watson»

The screenshot shows a messaging interface. On the left, there is a user icon (blue person) and a message box containing text from 'Jill Watson'. On the right, there is another user icon and a message box containing text from the user. A text input field and a 'Send' button are at the bottom.

I didn't find the title of the PDF file for submission. Can we just call it 'Assignment 1.pdf'?

Jill Watson

Please submit as a PDF. There isn't a specific format for the file name, so you can name it what you'd like.

Enter your message... Send

Abbildung: Ashok Goel,
TEDxSanFrancisco: „A
teaching assistant named Jill
Watson“
(<https://www.youtube.com/watch?v=WbCguICyfTA>)

«Jill Watson»: Sie sind dran!

The screenshot shows a messaging interface with two messages from 'Jill Watson'. Each message has an icon of a blue person and a text box.

Is it the case that only basic problems have the problem text data available?

There are no verbal representations of Challenge problems. They'll only be run as visual problems. But you're welcome to write your own verbal representations to try them out!

Is it permissible for Project 3 to be in a different language than our Project 1 and 2 submissions? (from Java to Python).

Yes, you may switch between the 2 languages from one project to the next with no problem.

Dialogsysteme: Herausforderungen I

Informationsgehalt
der Antworten

Input: What are you doing?

-
- 0.86 I don't know. - 1.09 Get out of here.
 - 1.03 I don't know! - 1.09 I'm going home.
 - 1.06 Nothing. - 1.09 Oh my god!
 - 1.09 Get out of the way. - 1.10 I'm talking to you.
-

Input: what is your name?

-
- 0.91 I don't know. ...
 - 0.92 I don't know! - 1.55 My name is Robert.
 - 0.92 I don't know, sir. - 1.58 My name is John.
 - 0.97 Oh, my god! - 1.59 My name's John.
-

[Abbildung:](#) Uninformative Antworten (Li u. a. 2015)

Dialogsysteme: Herausforderungen II

Sprecherkonsistenz

Einbindung von externem
Faktenwissen

-
- | | |
|-----------------|-----------------------------------|
| <i>message</i> | Where do you live now? |
| <i>response</i> | I live in Los Angeles. |
| <i>message</i> | In which city do you live now? |
| <i>response</i> | I live in Madrid. |
| <i>message</i> | In which country do you live now? |
| <i>response</i> | England, you? |
-
- | | |
|-----------------|------------------------|
| <i>message</i> | Where were you born? |
| <i>response</i> | I was born in Canada. |
| <i>message</i> | Where are you from? |
| <i>response</i> | England, you? |
| <i>message</i> | Where did you grow up? |
| <i>response</i> | I grew up in Texas. |
-

[Abbildung:](#) Sprecherinkonsistenz (Li u. a. 2016)

Referenzen

- Li, Jiwei u. a. (2015). "A Diversity-Promoting Objective Function for Neural Conversation Models". In: *CoRR abs/1510.03055*. arXiv: 1510.03055. URL: <http://arxiv.org/abs/1510.03055>.
- Li, Jiwei u. a. (2016). "A Persona-Based Neural Conversation Model". In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany, S. 994–1003.



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

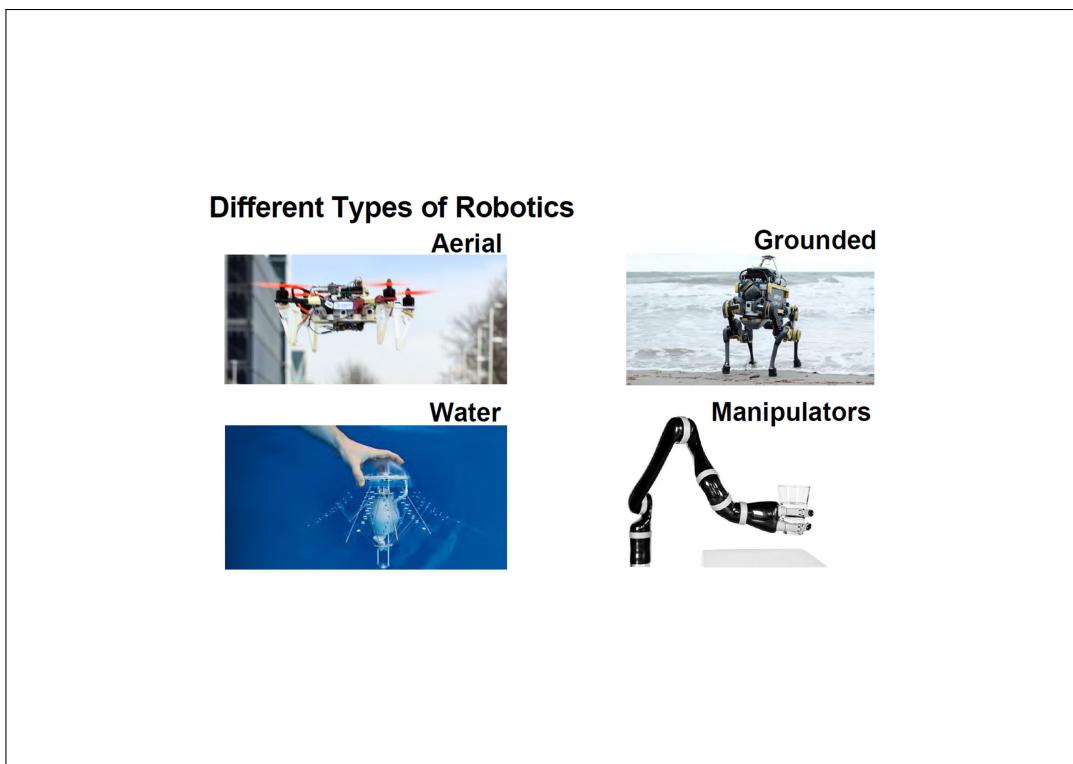
Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

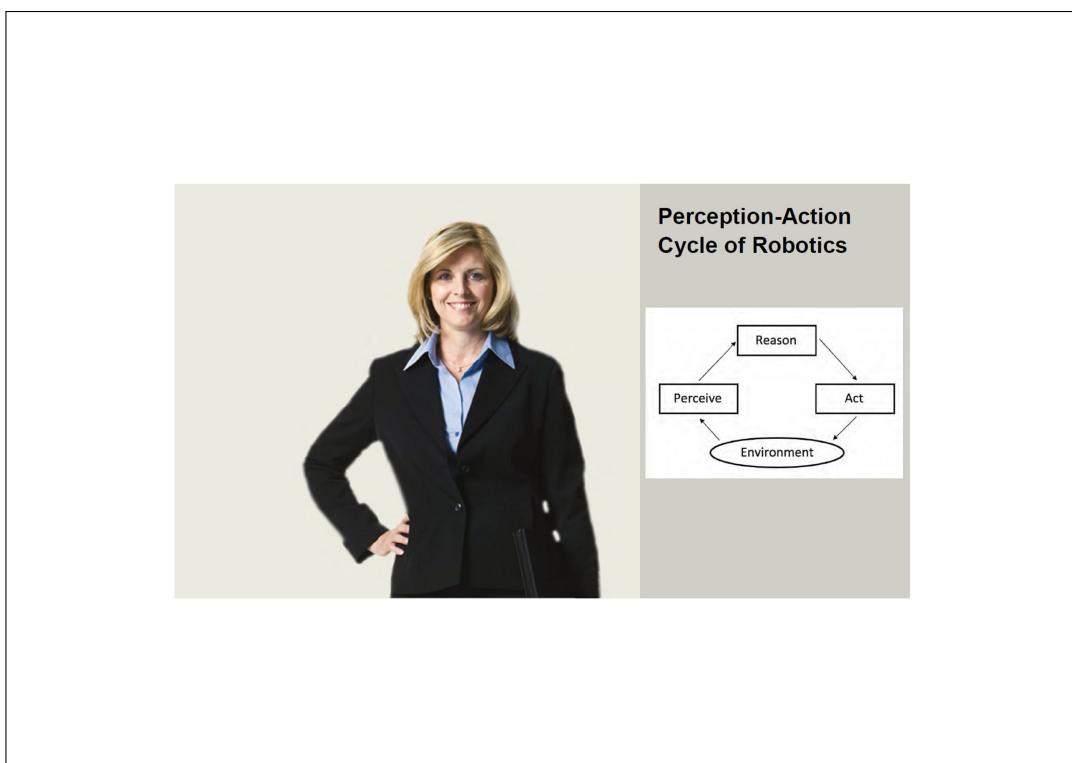
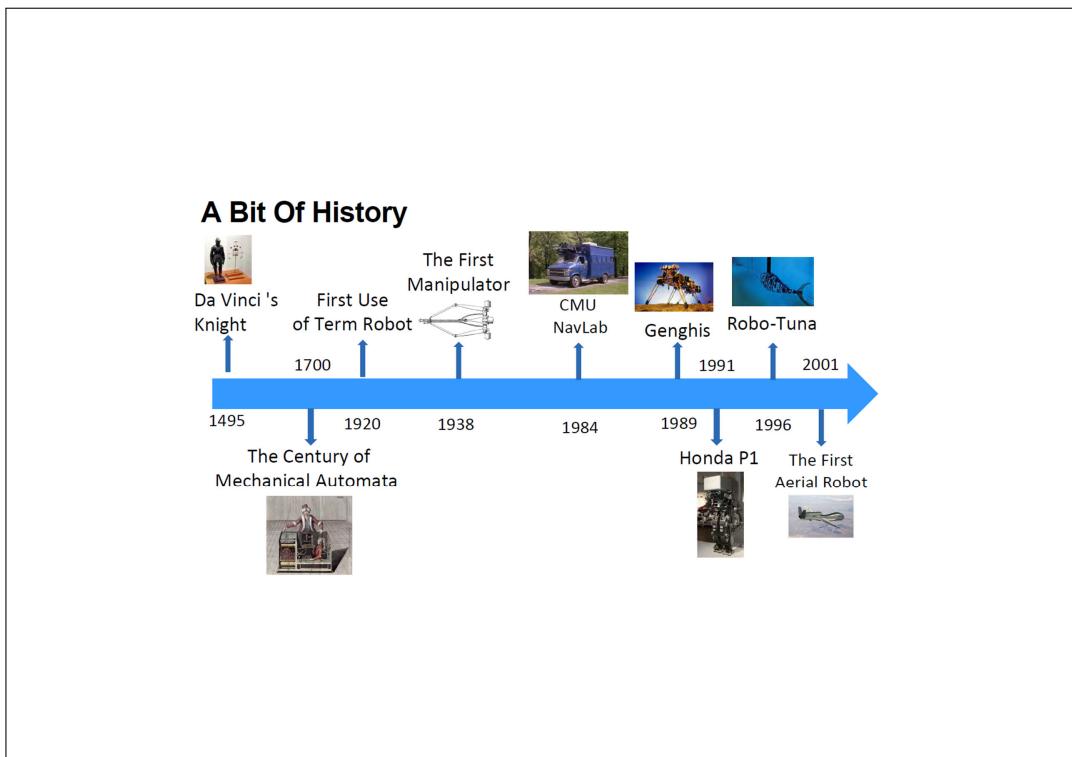
Lektion 9: Introduction to Robotics

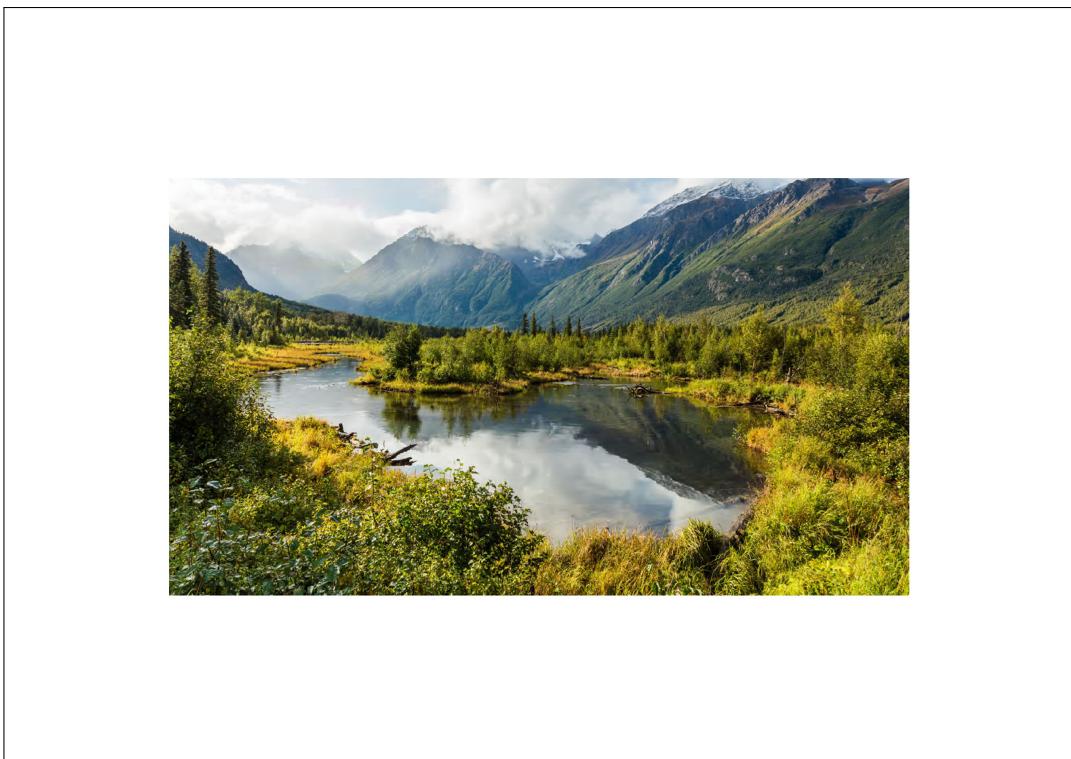
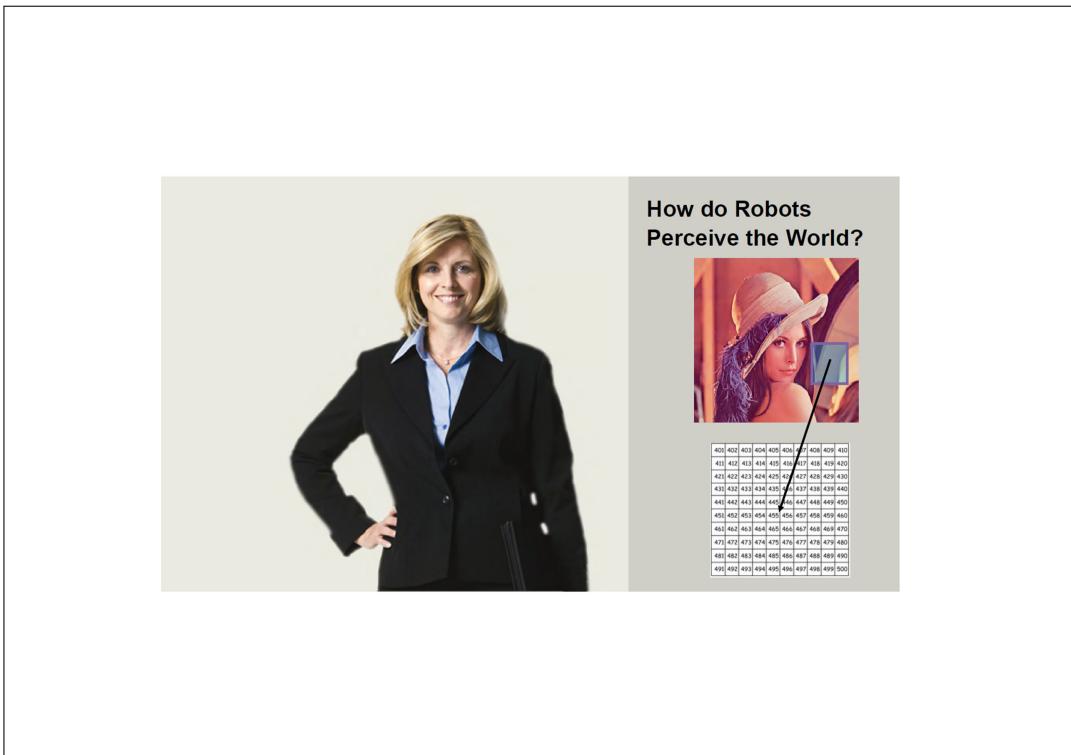
**What is Robotics?**

"The branch of technology that deals with the design, construction, operation, and application of robots."

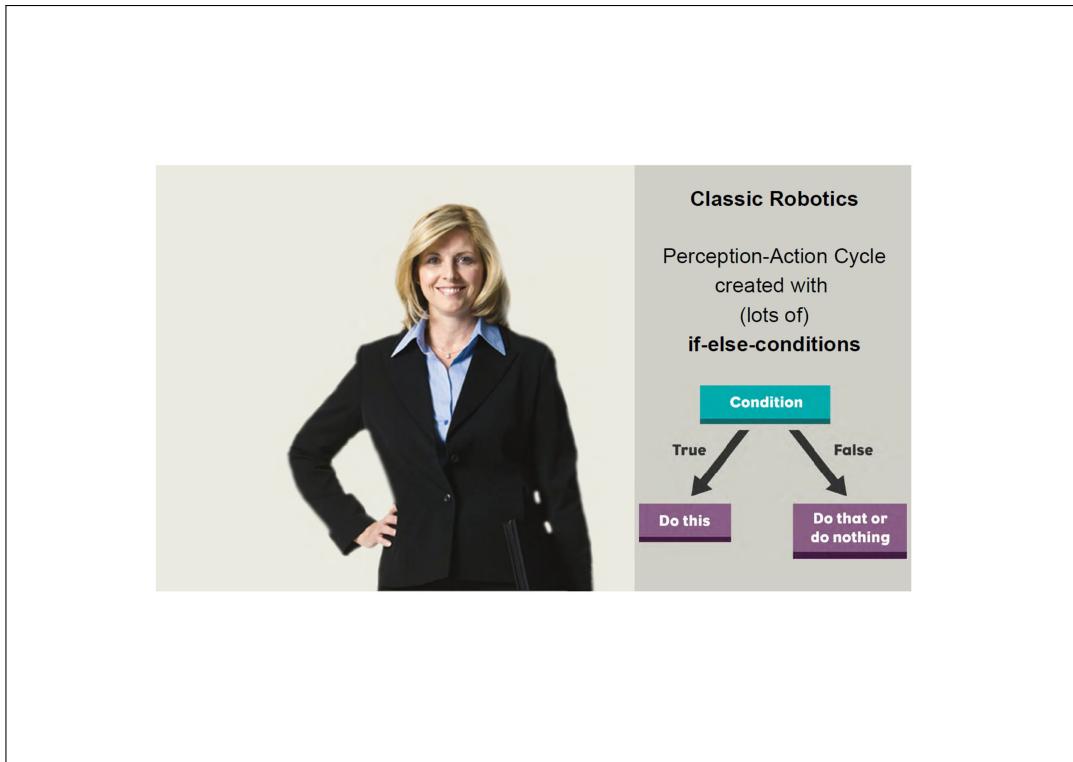
Oxford Dictionary

**Different Types of Robotics****Aerial****Grounded****Water****Manipulators**





12	12	12	12	12	12	40	40	12	12	26	12	7	12	8	12
33	97	12	24	37	93	33	8	8	8	7	12	7	18	12	7
12	24	8	26	24	17	37	37	11	38	43	60	46	33	26	18
60	69	117	61	37	37	43	18	28	28	19	33	18	37	37	17
17	17	117	179	113	12	18	43	28	101	89	80	80	53	51	53
19	33	33	19	26	12	12	31	33	7	12	8	8	14	8	8
57	74	26	8	69	60	31	14	31	8	8	8	8	33	31	8
8	8	8	8	8	8	31	65	74	65	61	40	12	26	48	33
60	18	53	33	140	97	33	28	12	28	206	60	26	48	12	31
33	40	42	40	31	8	26	12	8	12	48	46	8	14	8	12
12	8	12	12	8	116	42	8	8	31	31	12	12	14	14	12
14	14	18	12	7	11	8	8	81	19	26	31	31	8	26	31
36	4	33	46	35	24	28	37	9	35	160	35	37	36	24	11
89	79	87	69	50	37	37	57	57	80	80	50	53	37	50	32
36	50	69	43	37	37	37	32	53	43	53	51	37	50	37	37
37	38	53	30	57	87	9	31	113	213	37	37	32	28	40	40
17	32	169	141	66	7	17	12	19	4	43	164	32	32	32	143
24	17	43	40	43	43	37	30	32	37	60	43	28	48	37	43
43	32	138	69	51	43	31	60	79	6	37	108	163	80	40	28
24	37	51	108	48	35	5	189	117	7	43	31	32	35	35	13
13	30	13	13	18	17	13	11	22	29	13	32	31	31	48	13
24	30	24	24	32	17	43	43	37	53	37	6943	37	53	43	43
17	5	1	6	6	1	6	11	4	7	4	3	3	3	11	24
24	31	31	31	13	50	59	32	32	31	61	31	32	32	61	79
31	31	39	66	31	39	24	32	39	35	46	29	39	5	4	32
29	5	48	51	66	48	66	50	32	68	24	13	13	14	6	29
5	7	9	39	39	60	37	3	3	6	7	4	7	7	75	35
47	7	7	14	6	11	13	14	5	11	13	13	9	7	14	5
5	66	7	7	7	6	6	80	119	119	127	141	141	127	127	127
141	141	153	108	80	89	97	89	89	109	127	127	109	109	109	109
127	109	127	127	109	109	127	109	109	127	127	127	141	127	127	127
141	140	127	127	141	140	127	109	109	109	109	109	127	127	127	127





Modern Robotics:

Perception-Action
Cycle learned via
**interaction with
the physical world**

Take-Home Messages

- The **Perception-Action Cycle** defines a robot's operation.
- There is a **Perception-Gap** between robots and humans.
- Robots learn to act by **Interaction with the World**.

Referenzen

- ANYmal - a highly mobile and dynamic quadrupedal robot
By M. Hutter , C. Gehring , et al.
- Collective Robot Reinforcement Learning with Distributed Asynchronous Guided Policy Search
By Ali Yahya, Adrian Li, Mrinal Kalakrishnan, Yevgen Chebotar, and Sergey Levine
- Dronet: Learning to Fly By Driving
By A. Loquercio, A. Maqueda, R. Del Blanco, and D. Scaramuzza
- Probabilist Robotics
By Sebastian Thrun, Wolfram Burgard, Dieter Fox
- Introduction to Autonomous Mobile Robots
By R Siegwart, IR Nourbakhsh, D Scaramuzza



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und
maschinelles Lernen

Lektion 10: Auswirkungen der künstlichen Intelligenz

Auswirkungen künstlicher Intelligenz

- Ethisch
 - Psychologisch
 - Sozial
 - Ökonomisch
 - Rechtlich
- (Poole und Mackworth 2017)

Risiken im Zusammenhang mit KI

- Verlust von Arbeitsplätzen
 - **Verwendung für unbeabsichtigte oder unerwünschte Zwecke**
 - Frage der Rechenschaft
 - ...
- (Russell und Norvig 2016)

Bias in maschinellem Lernen

O bir doktor. O bir hemşire. ! He is a doctor. She is a nurse.

(Caliskan Islam, Bryson und Narayanan 2016, S. 11)



Abbildung: Twitter-Account von
Microsofts Chatbot Tay
([https://www.zeit.de/digital/internet/
2016-03/
microsoft-tay-chatbot-twitter-rassistisch](https://www.zeit.de/digital/internet/2016-03/microsoft-tay-chatbot-twitter-rassistisch))

Human parity?

Superhuman intelligence?

Singularity?

(Kurzweil 2005)

Referenzen

- Caliskan Islam, Aylin, Joanna J. Bryson und Arvind Narayanan (2016). "Semantics derived automatically from language corpora necessarily contain human biases". In: *CoRR abs/1608.07187*. arXiv: 1608.07187. URL: <http://arxiv.org/abs/1608.07187>.
- Kurzweil, Ray (2005). *The Singularity is near: When Humans Transcend Biology*. New York: Viking.
- Poole, David L. und Alan K. Mackworth (2017). *Artificial Intelligence: Foundations of Computational Agents*. 2nd. New York, NY, USA: Cambridge University Press. ISBN: 110719539X, 9781107195394.
- Russell, Stuart und Peter Norvig (2016). *Artificial Intelligence: A Modern Approach*. Pearson Education Limited.



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

Lektion 11: Was ist maschinelles Lernen?

Maschinelles Lernen

«A computer program is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P** if its performance at tasks in T, as measured by P, improves with experience E.» (Mitchell 1997, S. 2)

Maschinelles Lernen: Motivation

Beispiel ohne maschinelles Lernen: A girl eats an apple. →

Wörterbuch:

a : ein
girl : Mädchen
eat : essen
an : ein
apple : Apfel

Quellsprachgrammatik:

NP → Det N
VP → V

Zielsprachgrammatik:
NP → Det N
VP → V

Transferregeln:

NP → NP
Det N → Det N
VP → VP
V → V

(https://wikivisually.com/wiki/Rule-based_machine_translation)

Maschinelles Lernen: Motivation

Beispiel mit maschinellem Lernen: *A girl eats an apple.* →

Wir verwenden existierende Übersetzungen:

- *A girl eats two pears.* → *Ein Mädchen isst zwei Birnen.*
- *A woman eats an apple.* → *Eine Frau isst einen Apfel.*
- ...

Nachteile regelbasiertes Vorgehen

- Anpassung einer Aufgabe auf neuen Kontext aufwändig
- Definition von Regeln manchmal schwierig

Referenzen

Mitchell, Thomas M. (1997). *Machine Learning*. 1. Aufl.
New York, NY, USA: McGraw-Hill, Inc. ISBN:
0070428077, 9780070428072.



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und
maschinelles Lernen

Lektion 12: Zentrale Konzepte maschinellen Lernens,
Teil I

Daten

- Trainingsset
(training set)
 - Entwicklungsset
(validation/development set)
 - Testset
(test/evaluation set)
- [Abbildung: https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7](https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7)
- 

Experience und Task

Experience E:

- **Supervisiertes Lernen**
- Unsupervisiertes Lernen
- ...

Task T (supervisiertes Lernen):

- **Regression**
- Klassifikation
- ...

Beispiel: Lineare Regression

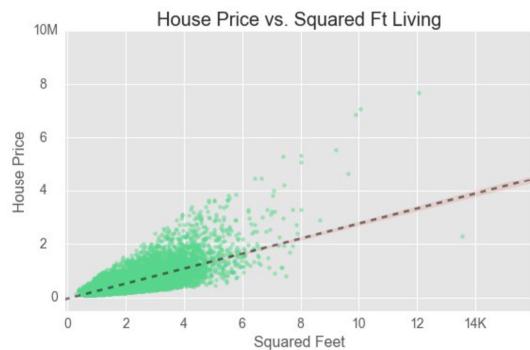


Abbildung: Lineare Regression
(<https://medium.com/@datalesdatales/>)

$$\hat{y} = wx + b$$



Universität
Zürich UZH

Digital Society Initiative



Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

Lektion 13: Zentrale Konzepte maschinellen Lernens,
Teil II

Performance measure P

- Training: minimaler Verlust, wenn wir \hat{y} vorhersagen, während die echte Ausgabe y ist
- Verlustfunktion (*loss function*)
- Beispiel: *Mean squared error* (MSE) (mittlerer quadratischer Fehler) $MSE = \frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2$
- Performanz des Lernalgorithmus auf ungesehenen Daten → Testset

Task T (supervisiertes Lernen)

- Regression
- **Klassifikation**
- ...

Klassifikation

Lineare Regression: $\hat{y} = w^T x + b$

Logistische Regression: $\hat{y} = \sigma(w^T x + b)$
mit $\sigma(x) = \frac{1}{1+e^{-x}}$

Unsupervisiertes Lernen

- Keine Labels
- z.B. Clustering

Clustering

- Menge von Elementen in Cluster aufteilen, sodass Mitglieder eines Clusters untereinander ähnlich und den Mitgliedern anderer Cluster unähnlich sind
- Bsp.: Clustering von Dokumenten anhand von linguistischen Merkmalen

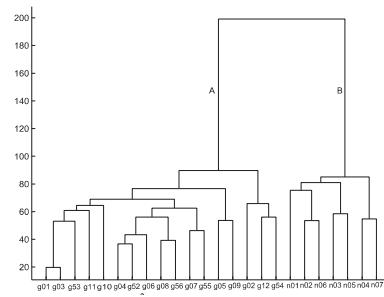


Abbildung: Cluster (Moisl 2015, S. 11)

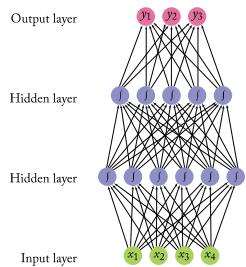


Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

Lektion 14: Deep Learning und künstliche neuronale Netze

Neuronale Netze



$$NN(x) = W^3(g^2(g^1(W^1x + b^1)) + b^2))$$

Abbildung: Multi-layer perceptron (Goldberg 2017, S. 42)

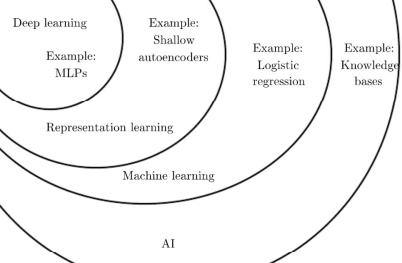


Abbildung: Terminologische Abgrenzung AI, machine learning, deep learning (Goodfellow, Bengio und Courville 2016, S. 9)

Neuronale Netze: Herausforderungen

- Grosse (!) Datenmengen benötigt
- Interpretierbarkeit
- Robustheit

Nachhaltigkeit
(Strubell, Ganesh und
McCallum 2019)



Teapot(24.99%)
Joystick(37.39%)



Hamster(35.79%)
Nipple(42.36%)

Abbildung: One pixel attack (Su, Vargas und Sakurai 2019, S. 829)

Referenzen

- Goldberg, Yoav (2017). *Neural Network Methods for Natural Language Processing*. Morgan & Claypool Publishers.
- Goodfellow, Ian, Yoshua Bengio und Aaron Courville (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Strubell, Emma, Ananya Ganesh und Andrew McCallum (2019). “Energy and Policy Considerations for Deep Learning in NLP”. In: *CoRR* abs/1906.02243. arXiv: 1906.02243. URL: <http://arxiv.org/abs/1906.02243>.
- Su, J., D. V. Vargas und K. Sakurai (2019). “One Pixel Attack for Fooling Deep Neural Networks”. In: *IEEE Transactions on Evolutionary Computation* 23.5, S. 828–841. ISSN: 1941-0026. DOI: 10.1109/TEVC.2019.2890858.



**Universität
Zürich^{UZH}**
Digital Society Initiative



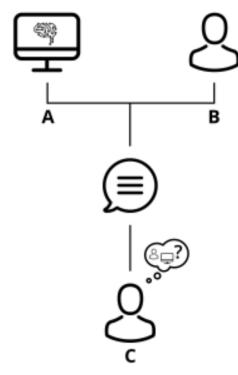
Studium Digitale

Kursbaustein 10: Künstliche Intelligenz und maschinelles Lernen

Lektion 15: Abschluss

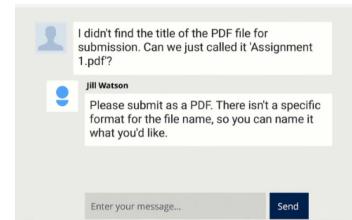
Künstliche Intelligenz

- Künstliche Intelligenz: Verstehen und Entwickeln von intelligent handelnden Maschinen
- Ansätze:
 - Menschlich denken
 - Rational denken
 - **Menschlich handeln** (Turing-Test)
 - Rational handeln
- Schwache KI vs. starke KI → Begriff *artificial intelligence* 1955 im Sinne von schwacher KI geprägt



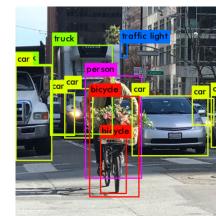
Teilbereiche künstlicher Intelligenz

- Automatische Sprachverarbeitung
 - Prototypische Anwendungen:
 - Maschinelle Übersetzung
 - Spracherkennung
 - ***Image captioning***
 - Dialogsysteme
 - Turing-Test: textbasiert
 - Persönliche Assistenten:
 - gesprochene Sprache
 - Wissensrepräsentation



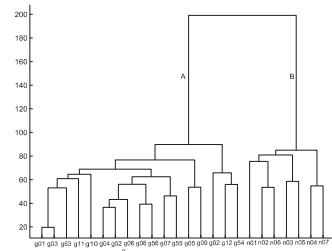
Teilbereiche künstlicher Intelligenz

- ***Reasoning***
- Maschinelles Lernen
- ***Computer vision***
 - Prototypische Anwendungen:
 - Objekterkennung
 - ***Image captioning***
 - Robotik
 - ...



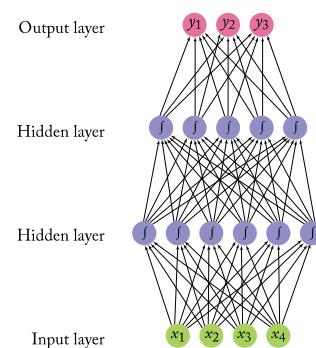
Bestandteile maschinellen Lernens

- *Experience E*: z.B. supervisiert, unsupervisiert
- *Task T*: z.B. Regression (lineare Regression), Klassifikation (logistische Regression), Clustering
- *Performance measure P*: z.B. *Mean squared error*



Neuronale Netze, *Deep learning*

- Künstliche neuronale Netze: komplexe nicht-lineare Funktionen, die aus kleineren Einheiten zusammengesetzt sind
- Künstliche Intelligenz > maschinelles Lernen > *Deep learning*



Referent*innen

Dr. Sarah Ebling

Prof. Abraham Bernstein

Dr. Antonio Loquercio

© Universität Zürich

Digital Society Initiative

A Skript: Mikroökonomie



Skript: Mikroökonomie

(Stand: 15. September 2025)

Autoren dieses Moduls:

Prof. Abraham Bernstein Ph.D., Universität Zürich

Dr. Katharina Reinecke, Universität Zürich
Andrea Kunz, Universität Zürich

A.1 Einleitung

A.1.1 Lehrziele

Das Selbstlernmodul Mikroökonomie" besteht aus folgenden vier Lerneinheiten:

- LE 1: Angebot und Nachfrage
- LE 2: Produktions- und Kostentheorie
- LE 3: Vollkommene Konkurrenz

Zu jeder Lerneinheit steht Ihnen ein Selbsttest zur Verfügung, mit dem Sie Ihren Lernerfolg überprüfen können.

Nach der Bearbeitung dieses Moduls sollten Sie Folgendes können:
die Nachfrage- und Angebotskurve beschreiben und aufzeichnen.
einfache komparative Statik betreiben.

A.1.2 Aufbau des Selbstlernmoduls Mikroökonomie

Mikroökonomie wird Ihnen nicht nur im Studium, sondern auch im alltäglichen Leben immer wieder begegnen. Dieses Modul soll Ihnen deshalb grundlegendes Wissen vermitteln. Aufbauend darauf werden später in der Vorlesung Aspekte der Mikroökonomie in Zusammenhang mit der Informatik wieder aufgegriffen und vertieft.

Die Volkswirtschaft beschäftigt sich vorwiegend mit Märkten. Sicherlich haben Sie schon von Märkten gehört oder sich gefragt, wie überhaupt Angebot und Nachfrage entstehen. Dies wollen wir in den nächsten Kapiteln genauer betrachten. Starten werden wir mit dem Thema "Angebot und Nachfrage". Danach werden wir uns mit Unternehmen beschäftigen und uns mit der Frage auseinandersetzen, wann die gewinnbringendste Menge erreicht ist. Wir werden Gewinn genauer definieren und untersuchen, wie dieser maximiert

werden kann. Um das Ganze verständlich zu machen, werden wir unsere gewonnenen Erkenntnisse immer mit Grafiken illustrieren.

A.2 Lerneinheit 1: Angebot und Nachfrage

A.2.1 Lehrziele

In dieser Lerneinheit erhalten Sie eine kurze Einführung zu Angebot und Nachfrage.

Nach der Bearbeitung der Lerneinheit sollten Sie folgende Fragen beantworten können:

Wie kann man Nachfrage in einem Diagramm darstellen?

Was ist ein Reservationspreis?

Was wird unter Angebot verstanden und wie verläuft die Gerade in einem Preis-Mengen-Diagramm?

Wie entstehen der Gleichgewichtspreis und die Gleichgewichtsmenge?

Was ist Wohlfahrt und wie entsteht die Konsumentenrente?

A.2.2 Märkte

Die Mikroökonomie beschäftigt sich mit Märkten. Ein Markt besteht aus Käufern und Verkäufern von einem bestimmten Gut und kann beispielsweise in Gütermarkt, Finanzmarkt und Arbeitsmarkt unterteilt werden. In einem Markt gibt es somit eine Nachfrage (Käufer) und ein Angebot (Verkäufer). Natürlich werden wir nicht immer den gesamten Gütermarkt betrachten, sondern diesen in kleinere Märkte, zum Beispiel den Markt für Hardware oder Lebensmittel unterteilen. Da auch diese Märkte noch sehr vielfältig sind, werden wir in den kommenden Kapiteln meist nur kleine Märkte betrachten, zum Beispiel einen Markt für Smartphones.

A.2.3 Die Nachfrage

Um die Nachfrage genauer beschreiben zu können, benötigen wir den Begriff "Reservationspreis". Der Reservationspreis steht für den Preis, den ein Individuum für ein bestimmtes Gut zu zahlen bereit ist (Ch. Ewerhart, 2007, S.18). Diese Zahlungsbereitschaft ist natürlich von Individuum zu Individuum verschieden. Ein Apple-Fan ist beispielsweise bereit, viel Geld für das neuste Gerät auszugeben. Eine andere Person, der solche Technik nicht viel bedeutet, wird einen tieferen Reservationspreis haben.

Daraus kann man ableiten, dass je höher ein Preis ist, desto weniger Käufer an einem Kauf interessiert sind. Wir können diesen Sachverhalt in einem Diagramm, genauer einem Preis-Mengen-Diagramm (Abbildung 1), darstellen. Es resultiert eine fallende Gerade. Die Menge kann auch mit einem Q (für das englische Wort Quantity) bezeichnet werden. Anstelle von Preis steht oft auch einfach ein P (englisch, price).

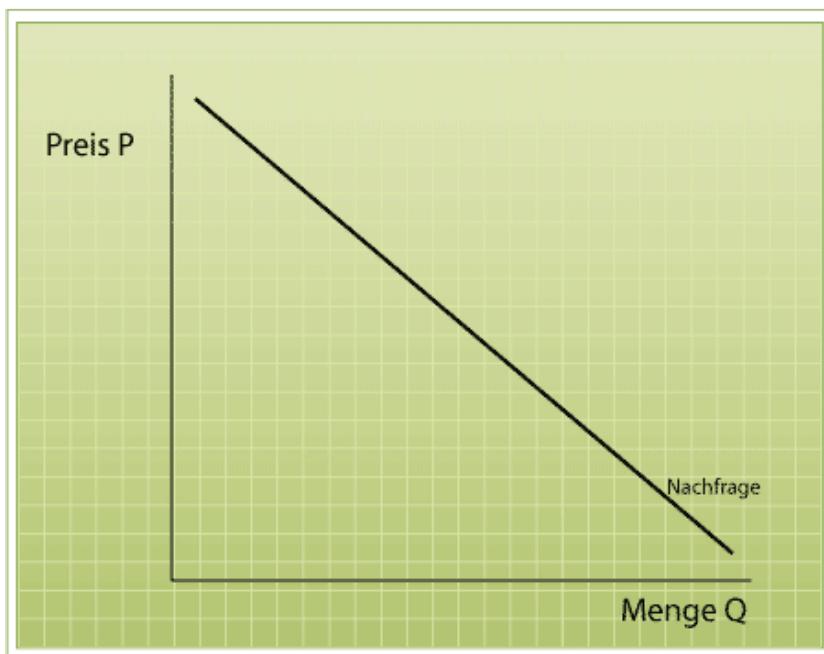


Abb.1 Die Nachfrage

Beispiel:**Reservationspreis**

Angenommen, eine Person irrt durch die Wüste und stösst nach langer Zeit auf einen Getränkestand. Da die Person sehr durstig ist (was in der Wüste bei grosser Hitze durchaus möglich ist), wird diese den Preis des angebotenen Wassers nicht gross interessieren, das heisst der Reservationspreis ist sehr hoch. Mitten in der Stadt, mit vielen Einkaufsmöglichkeiten, wird dieselbe Person vermutlich einen tieferen Reservationspreis haben. Der Reservationspreis ist daher nicht nur von Person zu Person unterschiedlich, sondern kann von vielen verschiedenen Einflüssen abhängen.

A.2.4 Das Angebot

Nicht nur Käufer, sondern auch Verkäufer haben einen Reservationspreis. Während dieser beim Käufer dem maximal akzeptierten Preis entspricht, ist der Reservationspreis beim Verkäufer der minimale Preis, zu dem er ein Gut verkaufen will.

Im Gegensatz zur Nachfrage, nimmt die angebotene Menge mit dem Preis zu (Abbildung 2). Je höher ein Preis für ein bestimmtes Gut ist, desto mehr wird von diesem Gut angeboten (B. Beck, 2008, S.33ff). Ob dieses Gut von einem einzigen Anbieter oder von mehreren angeboten wird (das heisst, ob mit steigendem Preis ein Anbieter mehr anbietet oder ob sich mehr Anbieter im Markt einfinden), ist momentan noch nicht wichtig.

Beispiel:

Wenn zum Beispiel für ein externes Blueray-Laufwerk nur ein Franken vom Käufer verlangt werden kann, wird wahrscheinlich kaum ein Hersteller bereit sein, ein solches Produkt zu produzieren. Erhält man jedoch für ein Laufwerk 1000 Franken, werden viel mehr Anbieter im Markt erscheinen. Je höher der Preis, desto mehr wird auf dem Markt angeboten.

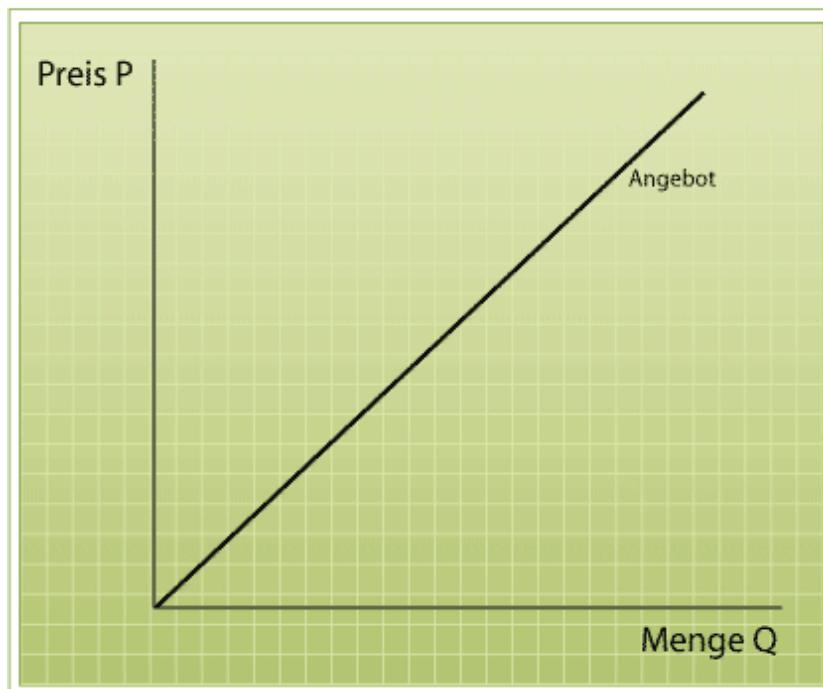


Abb.2 Angebotskurve

Beispiel:**Angebotsfunktion**

Peter hat eine kleine Backstube mit einem kleinen Backofen in dem er verschiedene Produkte wie Brote, Kuchen, Gipfel etc. backen kann. Gemäss der in der Grafik dargestellten Angebotskurve, wird Peter keine Brote backen, wenn er da-

für 0 oder nur 1 Franken bekommt. Dann backt er lieber andere Produkte oder arbeitet gar nicht und geniesst die Freizeit.

Für 2 Franken ist Peter bereit 10 Brote zu backen. Mehr aber nicht, denn es ist für ihn lukrativer seinen kleinen Ofen auch noch dafür zu nutzen andere Gebäcke herzustellen. Mit steigendem Marktpreis für das Brot wird Peter immer mehr Brot backen und dafür zum Beispiel weniger Kuchen oder Gipfel. Denn am Brot verdient er inzwischen mehr. So backt er bei 3 Franken 20 Brote oder bei 4 Franken 30 Brote.

Mit zunehmendem Preis wird er den Backofen schliesslich nur noch dazu nutzen Brote zu backen und keine anderen Produkte mehr herstellen. Er wird sogar einen zweiten Ofen kaufen und teure Hilfskräfte einstellen. Deshalb steigt die Angebotskurve für ein Gut mit zunehmendem Preis an."(Quelle: <http://www.vitemis.ch/d/lexikon/15/Angebotskurve.html>)

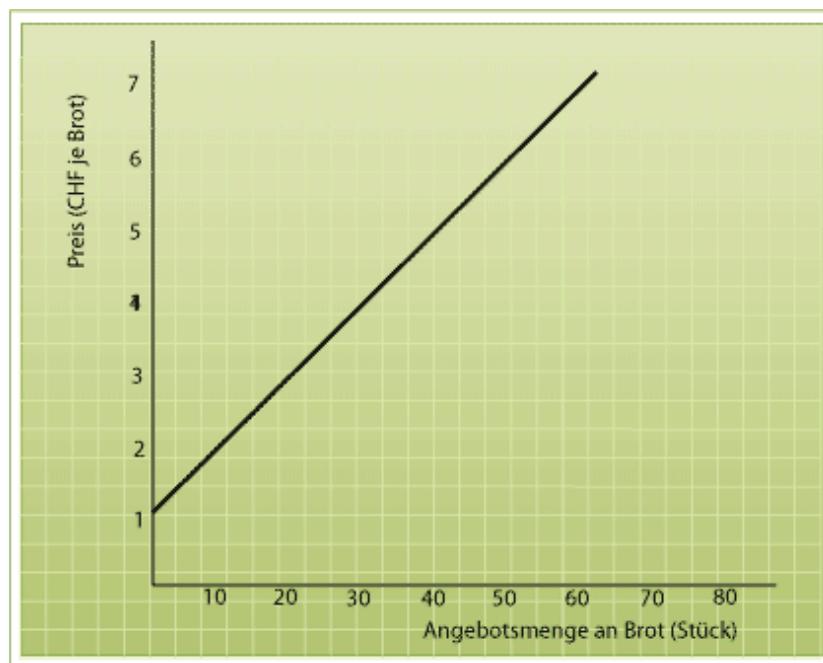


Abb.3 Peters Angebotskurve (in Anlehnung an www.vitemis.ch)

A.2.5 Marktgleichgewicht

Wir haben nun das Angebot und die Nachfrage je in einem Preis-Mengen-Diagramm eingezeichnet (Abbildungen 1 und 2). Es liegt nun nahe, diese beiden Kurven in einem Diagramm zu vereinen (Abbildung 4). Der Schnittpunkt der beiden Geraden bezeichnet das Gleichgewicht, das heisst, wir können beim Schnittpunkt sowohl die Gleichgewichtsmenge, als auch den Gleichgewichtspreis ablesen.

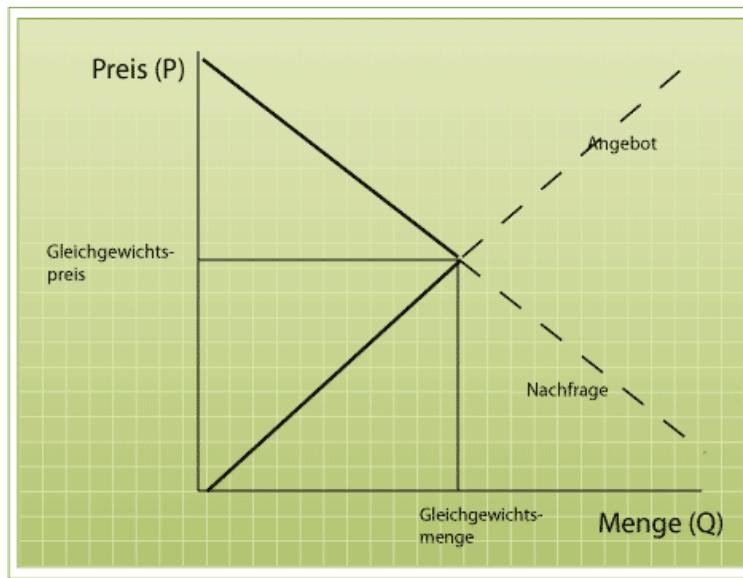


Abb.4 Mengen-Preis-Diagramm

Beim Betrachten der Abbildung 4 fällt auf, dass es ein Angebot als auch eine Nachfrage gibt, die nicht befriedigt werden. Alles, was rechts von der Gleichgewichtsmenge liegt (gestrichelte Gerade) wird nicht hergestellt, beziehungsweise nicht gekauft (B. Beck, 2008, S.38).

Aufgabe:

Die Nachfrage sei gegeben durch folgende Gerade: $Q^d = 6 - P$. Das Angebot wird durch folgende Gerade beschrieben: $Q^s = P/2$. Das hochgestellte *d* bezieht sich dabei auf den englischen Begriff demand (Nachfrage") und das *s* auf supply (Angebot").

Berechnen Sie die Gleichgewichtsmenge und den Gleichgewichtspreis.

Lösung

$$Q^d = Q^s$$

$$6 - P = P : 2 \quad | * 2$$

$$12 - 2P = P \quad | + 2P$$

$$12 = 3P \quad | : 3$$

$$P = 4 \text{ und } Q = 2$$

A.2.6 Komparative Statik

Wir haben nun das Markgleichgewicht berechnet. Dieses bleibt aber nicht immer gleich, denn Märkte können sich über die Zeit verändern, indem sich die Nachfrage- und Angebotskurven verschieben. Solche Veränderungen können wir ebenfalls im Angebots-Nachfrage-Diagramm darstellen.

Verschieben der Nachfragekurve

Beispielsweise kann in einem strengen Winter die Nachfrage nach Holz steigen, d.h. die Individuen sind bereit, für die gleiche Menge an Holz einen höheren Preis zu zahlen. Die Nachfrage nach einer bestimmten Kleidermarke kann beispielsweise steigen, nachdem ein Superstar diese an einem seiner Konzerte getragen hat. Es gibt eine grosse Vielfalt an Gründen warum die Nachfrage steigt. Gemäss Abbildung 5 hat dies jedoch immer die gleichen Auswirkungen. Die Gleichgewichtsmenge und der Gleichgewichtspreis steigen beide, d.h. es wird eine grösse Menge zu einem höheren Preis verkauft. Eine sinkende Nachfrage führt demnach analog zu einem niedrigeren Preis und einer niedrigeren Menge (Ch. Ewerhart, 2007, S.74).

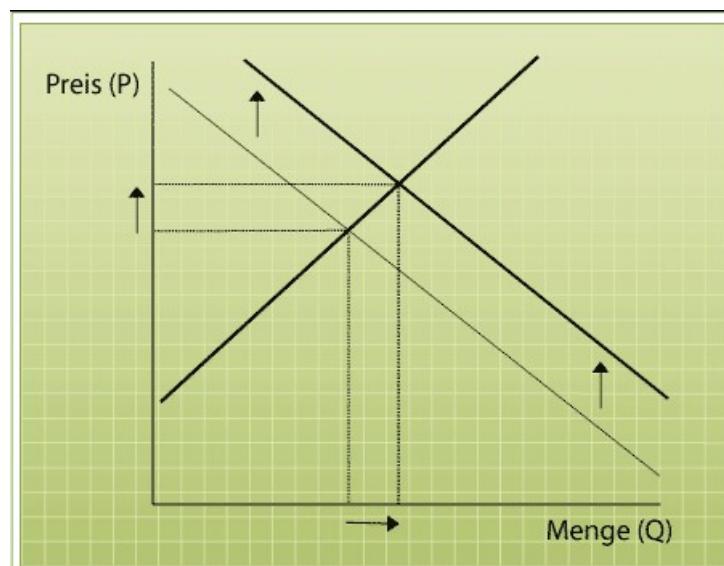


Abb.5 Verschiebung der Nachfragekurve nach oben

Verschieben der Angebotskurve

Wenn sich die Angebotskurve nach oben verschiebt, bedeutet dies, dass für die gleiche Menge ein höherer Preis verlangt wird. Dies kann beispielsweise geschehen, wenn sich Rohstoffpreise verteuern (z.B. Kakao für Schokolade) oder bei saisonalen Produkten (im Winter können z.B. keine Äpfel in der Schweiz geerntet werden, weshalb sich die Preise für Äpfel erhöhen). Wenn die Angebotskurve steigt, dann erhöht sich der Preis, während die gehandelte Menge sinkt (siehe Abbildung 6). Analog sinken die Preise und die Menge steigt, wenn die Angebotskurve nach unten verschoben wird (Ch. Ewerhart, S.75).

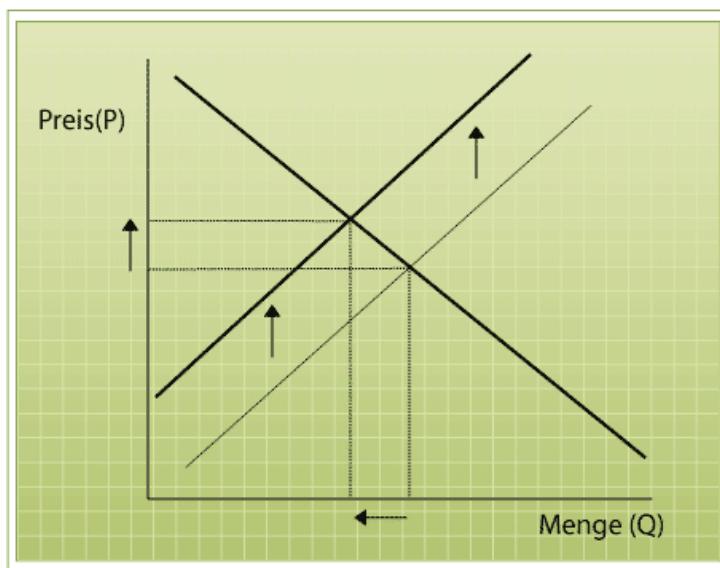


Abb.6 - Verschiebung der Angebotskurve nach oben

A.2.7 Konsumentenrente

Stellen Sie sich vor, es gäbe genau eine Sorte Brot zu einem immer gleichbleibenden Preis von 5 Franken. Stellen Sie sich nun weiter vor, Sie erhalten die Möglichkeit, ein Brot der gleichen Sorte für nur 3 Franken zu kaufen. Was wäre dabei Ihre Einsparung? Richtig, zwei Franken. Dies nennt man die Konsumentenrente. Erinnern wir uns zurück an das erste Kapitel. Hier haben wir die Nach-

fragekurve hergeleitet. Dabei sind wir davon ausgegangen, dass je tiefer der Preis ist, desto mehr Käufer sich finden lassen.

Betrachten wir nun drei verschiedene Personen und ein bestimmtes Gut zu einem Preis von P^* (das Sternchen bezeichnet in diesem Falle den Gleichgewichtspreis). Betrachten Sie hierfür auch die Abbildung 13.

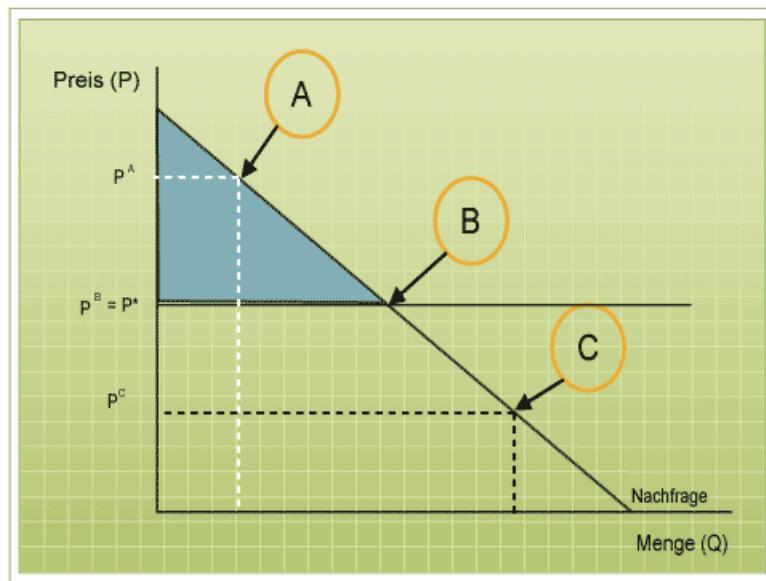


Abb.13 Konsumentenrente

Person A: Person A wäre bereit einen Preis der höher als der tatsächliche Preis P^* ist zu bezahlen. Die Konsumentenrente von Person A wäre somit $P^A - P^*$.

Person B: Person B ist gerade bereit den geforderten Preis P^* zu bezahlen. Die Konsumentenrente beträgt somit 0.

Person C: Person C ist bereit einen niedrigeren Preis P^C als den tatsächlichen Preis P^* zu bezahlen. Person C wird daher das Gut nicht kaufen. Es existiert keine Konsumentenrente für C.

Konsumentenrente: Die Konsumentenrente setzt sich zusammen aus allen

Konsumentenrenten von Individuen und ist damit gleich der markierten Fläche in der Abbildung 13.

A.2.8 Nachfrage- und Angebotsüberschuss

Wie wir bisher gesehen haben, passen sich die Preise und Mengen dem Angebot und der Nachfrage an. Was geschieht nun aber, wenn ein fixer Preis vom Staat festgelegt wird? Ist der fixe Preis über dem Gleichgewichtspreis, dann resultiert dies in einem Angebotsüberschuss. Das heisst, zu diesem fixen Preis ist die Nachfrage geringer als das Angebot. Analog dazu gibt es einen Nachfrageüberschuss, wenn der Preis unter dem Gleichgewichtspreis liegt (Löchel H., 2003, S.131).

Beispiel:

Nehmen wir an der Zuckerpreis sei in den USA künstlich (das heisst durch die Regierung) hochgehalten. Wie wir gesehen haben, resultiert daraus ein Angebotsüberschuss. Was geschieht aber mit den Nachfragern, welche nicht bereit sind, den hohen Preis zu bezahlen? Sie werden auf Produkte umsteigen, die denselben Nutzen haben, aber billiger als Zucker sind. Zum Beispiel Maissirup (HFCS). Produkte, welche einen ähnlichen Nutzen haben, nennt man Substitute.

In der Schweiz ist beispielsweise Fleisch sehr teuer. Daher konnten sich ebenfalls teure Substitute (beispielsweise Quorn-Produkte) einen Markt bei Nicht-Vegetariern verschaffen, da sie im Vergleich zum Fleisch "billig /dq sind.

Anhand dieser Beispiele sehen wir, dass künstlich hochgehaltene Preise Substitute fördern.

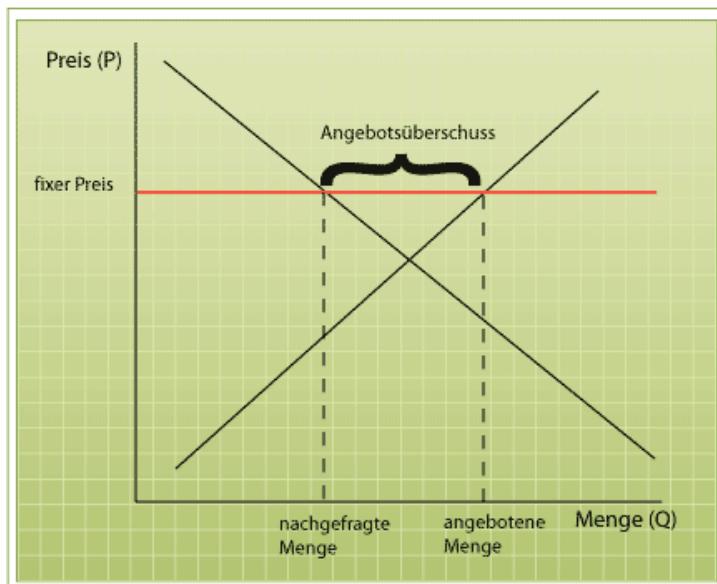


Abb.7 Angebotsüberschuss

A.2.9 Zusammenfassung

In dieser Lerneinheit haben Sie einen Einblick in Angebot und Nachfrage erhalten.

Folgende Themen wurden in der Lerneinheit behandelt:

Nachfrage: fallende Gerade im Preis-Mengen-Diagramm

Reservationspreis: Preis, den ein Individuum maximal zu zahlen bereit ist, resp. minimaler Preis zu dem ein Unternehmen bereit ist, ein Gut herzustellen.

Angebot: steigende Gerade im Preis-Mengen-Diagramm

Gleichgewichtspreis und die Gleichgewichtsmenge: Schnittpunkt der Angebots- und der Nachfragekurve

Konsumentenrente: Summe aus allen Konsumentenrenten von Individuen

A.3 Lerneinheit 2: Produktions- und Kostentheorie

A.3.1 Lehrziele

In dieser Lerneinheit erhalten Sie einen Überblick über die Produktions- und Kostentheorie.

Nach der Bearbeitung der Lerneinheit sollten Sie folgende Fragen beantworten können:

Was versteht man unter einer Produktionsfunktion und wie verläuft diese typischerweise?

Was versteht man unter Grenzproduktivität und Durchschnittsproduktivität?

Was ist eine Kostenfunktion und wie verläuft diese?

Was ist der Unterschied zwischen fixen und variablen Kosten?

A.3.2 Einleitung

Der Schnittpunkt der Angebots- und Nachfragekurve, die wir im letzten Kapitel vorgestellt haben, ergibt den Gleichgewichtspreis und das Gleichgewichtsangebot. Wie die Nachfrage entsteht, konnten wir anhand einiger Beispiele klären. Beim Angebot verhält es sich etwas komplizierter. Je höher der Preis, desto mehr wird angeboten. Wir haben offen gelassen, ob dieses größere Angebot von einer einzelnen Firma oder von mehreren kommt. In diesem Kapitel betrachten wir einzelne Unternehmen und ihre Produktionsfunktion. Wir werden sehen, dass es sich nicht immer lohnt, mehr zu produzieren.

A.3.3 Produktionsfunktion

Produktion ist der Prozess bei dem aus verschiedenen Inputs (zum Beispiel Rohstoffe) ein Output (das Produkt) generiert wird. Die Produktionsfunktion be-

schreibt dabei den Zusammenhang zwischen Input und Output und zeigt auf, wie viel mit einem gegebenen Input höchstens an Output produziert werden kann (Ch. Ewerhart, 2007, Seite 202).

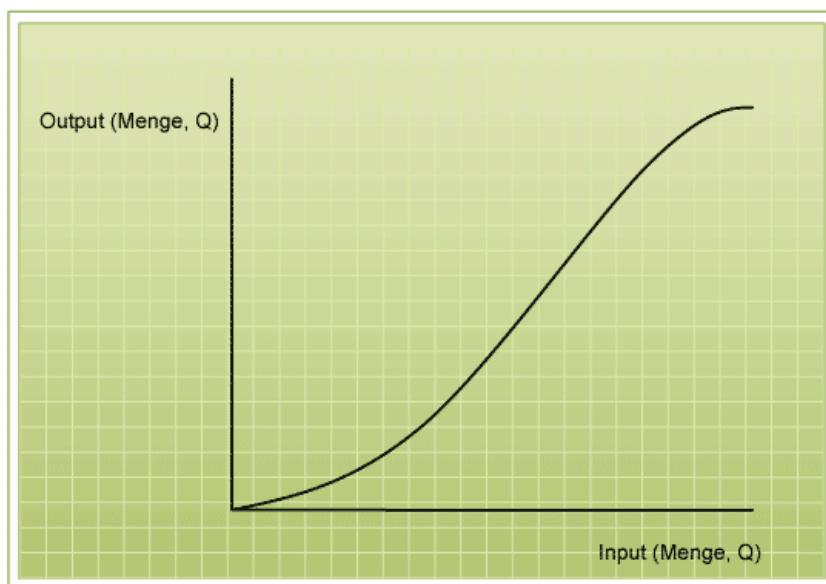


Abb.8 Produktionsfunktion

Wir werden nun versuchen, die Produktionsfunktion in einem Diagramm einzuziehen. Zunächst gehen wir davon aus, dass die Produktionsfunktion einer steigenden Geraden entspricht.

Ist dies wirklich der Fall? Nehmen wir als Beispiel einen Konditor, welcher sich auf Schokoladenkuchen spezialisiert hat. Für einen Kuchen braucht er 500 Gramm an diversen Zutaten und eine Stunde (inkl. der Backzeit). Zutaten sowie Arbeitseinsatz (Arbeitszeit) sind der Input. Da gewisse Zutaten nur in grösseren Mengen erhältlich sind, wiegen diese insgesamt 800 Gramm (300 Gramm werden nicht gebraucht). Damit kann der Konditor nun einen Output (einen Kuchen) generieren. Was ist nun, wenn er sowohl die Zutaten als auch die Arbeitszeit verdoppelt – kann er dann doppelt so viele Kuchen herstellen? Mit 1600 Gramm an Zutaten und zwei Stunden Input wären drei Kuchen möglich (wenn der erste Kuchen im Backofen ist (Backzeit 30 Minuten)

kann der Konditor bereits den nächsten Kuchen herstellen). Somit sind mit einer Verdoppelung des Inputs drei Kuchen möglich. Wir sehen, der Output ist überproportional grösser als der Input. Allerdings ist dies nur zu Beginn der Fall, wie wir der Kurve in Abbildung 9 entnehmen können. Grund dafür ist, dass der Konditor immer weniger Energie hat. Wenn er noch mehr Zeit investiert steigt der Output nicht mehr so stark, da der Konditor erschöpft und deswegen nicht mehr so produktiv ist.

Diesen Sachverhalt können wir auch mit Begriffen beschreiben. Die **Durchschnittsproduktivität** (AP, englisch für Average Productivity) beschreibt, wie viel Output (Kuchen) wir im Durchschnitt mit dem Input (Zeit) herstellen können. In der Abbildung 9 sehen wir, dass diese zunächst ansteigt (wir können mit mehr Input im Schnitt mehr herstellen) und ab einer gewissen Menge wieder sinkt.

Der zweite Begriff, den wir in diesem Zusammenhang gebrauchen können ist die **Grenzproduktivität** (MP, englisch für Marginal Productivity). Dieser Wert sagt, wie viel mehr Output wir mit *einer Einheit Input* mehr herstellen können. Zum Beispiel, wie viele Kuchen können wir mehr produzieren, wenn der Konditor eine Stunde länger arbeitet? Die Grenzproduktivität entspricht mathematisch gesehen der Steigung der Produktionsfunktion. Der Abbildung 9 können wir entnehmen, dass die Grenzproduktivität zunächst ansteigt und dann wieder sinkt.

Beispiel:

Nehmen wir an, Sie betreiben ein kleines Unternehmen, welches T-Shirts mit einem vom Kunden gewünschten Logo bedruckt. Sie selbst können am Tag drei T-Shirts bedrucken (der gesamte Arbeitsvorgang beinhaltet das Waschen des T-Shirts, das Ausdrucken auf Spezialfolie des Logos, das Aufbügeln des Logos und das anschliessende Verpacken, inkl. Rechnungserstellung). Wenn Sie einen weiteren Mitarbeiter einstellen, können Sie 8 T-Shirts am Tag bedrucken. Sie sehen, eine Erhöhung des Inputs (in diesem Falle die Anzahl der

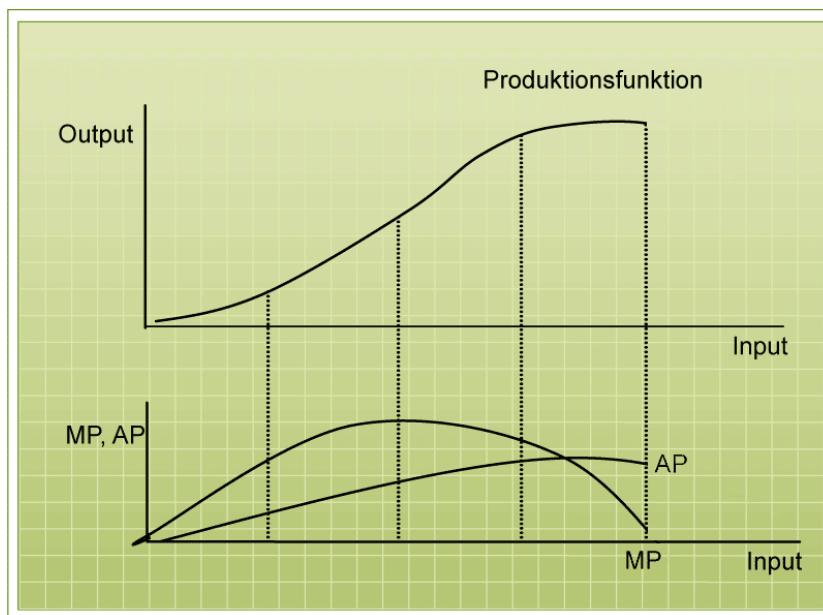


Abb.9 Produktionsfunktion, Durchschnittsproduktivitätskurve (AP), Grenzproduktivitätskurve (MP)

Mitarbeiter) erhöht die Anzahl des Outputs (die T-Shirts) überproportional. Die Grenzproduktivität ist somit grösser als 1 (die Erhöhung des Inputs um eine Einheit erhöht den Output um mehr als eine Einheit). Ab einer gewissen Anzahl an Mitarbeitern sinkt die Grenzproduktivität jedoch wieder. Zum Beispiel, weil die Kapazität des Bügeleisens erreicht ist. Selbstverständlich wäre es auch möglich, sich ein weiteres Bügeleisen zu beschaffen und die Produktivität somit wieder zu steigern.

Definition:

Die **Grenzproduktivität (MP)** gibt an, wie viel zusätzliche Einheiten an Output eine weitere Einheit Input generiert und entspricht somit der Steigung der Produktionsfunktion (erste Ableitung der Produktionsfunktion).

Die **Durchschnittsproduktivität (AP)** zeigt, wie viel Output mit einer Einheit

Input im Durchschnitt erzeugt werden kann.

A.3.4 Kostenfunktion

Mit dem Wissen über die Produktionsfunktion ist es nun möglich, eine Kostenfunktion aufzustellen. Im obigen T-Shirt-Beispiel haben wir gesehen, dass es verschiedene Arten von Kosten gibt. Die variablen Kosten verändern sich mit der Anzahl der produzierten Güter (Anzahl Mitarbeiter und ihr Lohn, Materialien wie die Folien zum Bedrucken der T-Shirts und so weiter). Daneben gibt es aber auch Kosten, die sich mit der Menge nicht verändern. Diese nennt man fixe Kosten (beispielsweise das Bügeleisen, die Miete, etc.).

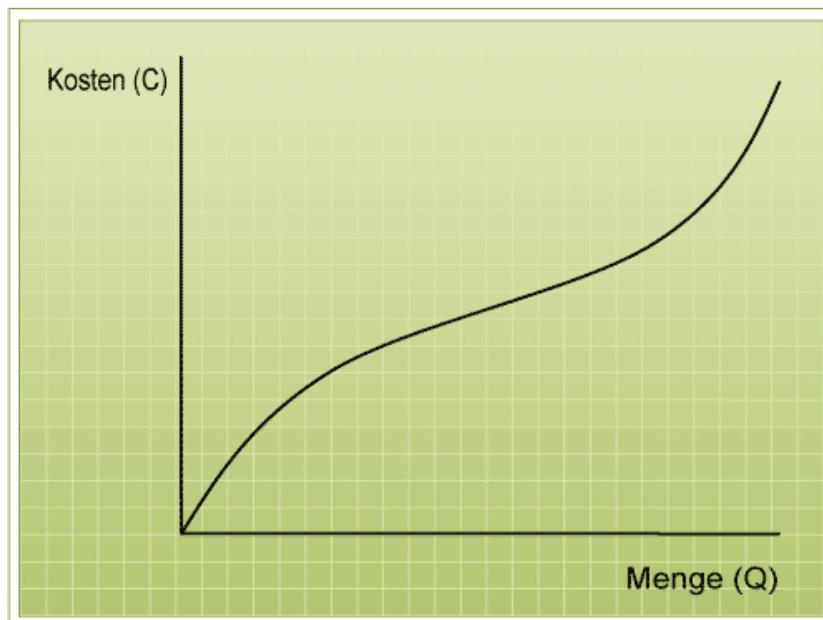


Abb.10 Kostenfunktion

- Die **totalen Kosten (TC)** sind gleich den fixen Kosten (FC) plus variablen Kosten (VC) ($TC = FC + VC$).

- Die **Grenzkosten (MC)** bezeichnen die zusätzlichen Kosten welche bei einer Mengenerhöhung (beispielsweise der Produktion einer zusätzlichen Einheit) anfallen.
- Die **Durchschnittskosten (ATC)** lassen sich aus den totalen Kosten dividiert durch die Menge berechnen.

Die Kostenfunktion in Abbildung 10 ist daher nur dann korrekt, wenn wir eine Produktion ohne Fixkosten betrachten. Wenn wir die fixen Kosten auch betrachten, sieht die Kostenfunktion wie in Abbildung 11 aus:

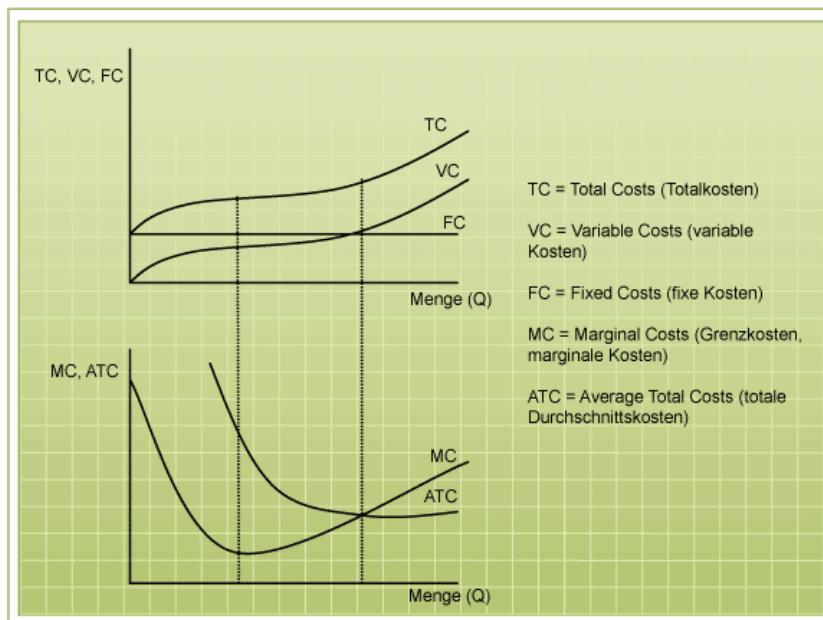


Abb.11 Kostenfunktionen

A.3.5 Zusammenfassung

Folgende Begriffe wurden in dieser Lerneinheit behandelt:

Produktionsfunktion: Grenzproduktivität, Durchschnittsproduktivität

Kostenfunktion: Grenzkosten, Durchschnittskosten

A.4 Lerneinheit 3: Vollkommener Wettbewerb

A.4.1 Lehrziele

In dieser Lerneinheit erhalten Sie eine Einführung zum Thema "Vollkommener Wettbewerb".

Nach der Bearbeitung der Lerneinheit sollten Sie folgende Fragen beantworten können:

Wie entsteht Gewinn?

Warum muss der Preis gleich den Grenzkosten sein?

In welchem Zusammenhang stehen die totalen Kosten, die Erträge und der Gewinn?

A.4.2 Einleitung

Um profitabel zu wirtschaften ist es für ein Unternehmen sehr wichtig, Gewinn zu erzielen. In dieser Lerneinheit werden wir daher genauer betrachten, woraus sich Gewinn zusammensetzt, wie dieser maximiert werden kann und ob es ein Gewinnmaximum gibt. In diesem Zusammenhang werden wir auch den vollkommenen Markt kennen lernen, ein vereinfachtes Modell eines Marktes, welches uns ermöglicht, komplexe Zusammenhänge relativ einfach darzustellen.

A.4.3 Gewinnmaximierung

In der Einleitung haben Sie gelesen, dass ein Unternehmen seinen Gewinn maximieren möchte. Gewinn (Π oder englisch *Profit*) entspricht dem Erlös (TR , englisch für Total Revenue) minus den Totalkosten (TC , englisch für Total Costs) ($\Pi = TR - TC$). Wir werden nun die Marksituation im **vollkommenen Wettbewerb** genauer betrachten. Vollkommener Wettbewerb bedeutet, dass

alle Unternehmen ein homogenes (d.h. genau das gleiche) Gut verkaufen und exakt über alle Preise der Konkurrenten informiert sind, ebenso besitzen die Konsumenten vollkommene Information. Zudem sind die Unternehmen Mengenanpasser, d.h. der Preis ist fix und jedes Unternehmen passt seine produzierte Menge so an, dass der maximale Gewinn erwirtschaftet werden kann. Der vollkommene Wettbewerb ist ein theoretisches Modell und entspricht daher nicht unbedingt der Wirklichkeit, ermöglicht uns aber, komplexe Zusammenhänge in einer vereinfachten Umgebung genauer zu untersuchen. Ein Unternehmen kann somit den Gewinn steigern indem es entweder die Totalkosten senkt oder den Erlös steigert.

Der Erlös ist der Gegenwert, den ein Unternehmen für den Verkauf von bestimmten Gütern (oder Dienstleistungen) erhält. Der Erlös lässt sich relativ einfach als "verkaufteMenge * Verkaufspreis" berechnen. Im Englischen wird der Erlös als Revenue bezeichnet.

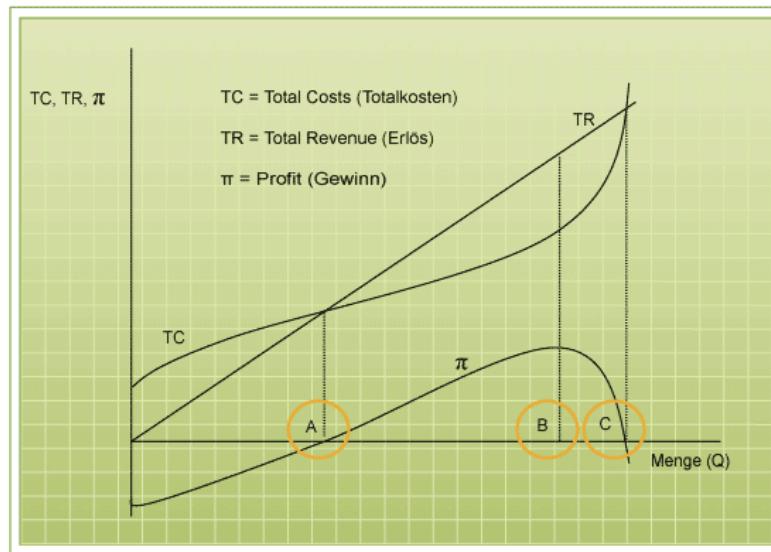


Abb.12 TC, TR und Gewinn

In der Abbildung 12 sehen wir, dass der Gewinn zunächst negativ ist, dies hängt damit zusammen, dass wenn nichts produziert wird, nur die fixen Kosten an-

fallen. Danach steigt der Gewinn stetig an. Sobald die Kosten (TC) gleich den Erträgen (TR) sind (Punkt A), geht der Gewinn (II) ins Positive. Der höchste Profit (das Gewinnmaximum) wird erreicht, wenn der Abstand der TR-Kurve und der TC-Kurve am grössten ist, sprich, wenn die Steigung der TR-Kurve gleich der Steigung der TC-Kurve ist (Punkt B). Sobald sich die TC und TR-Kurve ein zweites Mal schneiden (Punkt C) wird der Gewinn wieder negativ (G. Hildmann, 2005, S.111).

Wir wissen, dass die Steigung der totalen Kosten den marginalen Kosten (MC) entspricht. Wenn der Preis nun gleich den MC ist, ist der Gewinn maximal (Punkt B). Dieser Gedankengang ist sehr wichtig für die weiteren Kapitel, daher hier nochmals in anderen Worten: Die Gesamterlöse entsprechen dem, was ein Unternehmen an Gegenwert für die verkauften Waren erhält, d.h. die verkaufte Menge multipliziert mit dem Verkaufspreis. Anhand der obigen Grafik sehen wir, dass im Profitmaximum (Gewinnmaximum) die Steigung der Erträge gleich der Steigung der Kostenkurve sein muss. Da die Steigung der Geraden dem Preis entspricht, muss also auch die Steigung der Kosten im Gewinnmaximum dem Preis entsprechen.

Wir sehen, Ziel eines jeden Unternehmens muss die Menge Q sein, welche sich beim Gewinnmaximum, d.h. in Punkt B befindet. Wir werden die Gewinnkurve nun abschnittsweise genauer betrachten.

X-Achse Abb. 12: 0 bis A

Die Kosten sind grösser als der Erlös, so dass ein Unternehmen einen Verlust macht. Der Grenzerlös, d.h. der Erlös, wenn eine zusätzliche Einheit produziert wird, ist höher als die zusätzlichen Kosten (Grenzkosten). Dies bedeutet, dass sich eine Steigerung der Produktion positiv auf den Gewinn auswirkt. Dies sehen wir auch anhand der Gewinnkurve, welche ansteigt.

X-Achse Abb. 12: A bis B

Mittlerweile ist der Erlös grösser als die Kosten, es wird also ein Gewinn erwirtschaftet. Noch immer ist der Grenzerlös grösser als die Grenzkosten, daher lohnt es sich, die Produktion auszuweiten. Ob der Grenzerlös grösser oder kleiner als die Grenzkosten ist, sehen wir übrigens anhand der Steigungen. Von A nach B ist die Steigung der Erlöskurve (TR) immer grösser als die Steigung der Kostenkurve (TC). Somit steigt der Gewinn stetig an.

X-Achse Abb. 12: B bis C

Von B nach C ist die Steigung der Kosten höher als die Steigung der Erträge, der Gewinn sinkt, es lohnt sich also nicht, die Produktion auszuweiten. In diesem Fall sollte das Unternehmen die Produktion reduzieren.

Definition:

Der Gewinn kann auch mathematisch hergeleitet werden.

Der Gewinn π entspricht dem Erlös minus den Totalkosten.

$$\pi(Q) = TR(Q) - TC(Q)$$

Das Q in Klammer bedeutet, dass sich die Variablen mit der Menge verändern. Durch Ableiten nach Q und dem Nullsetzen, erhält man die notwendige Bedingung, für die Gewinnmaximierung.

$$\max \pi(Q) = TR(Q) - TC(Q)$$

$$d\pi/dQ = dTR/dQ - dTC/dQ \quad \text{Anmerkung: } dTR/dQ = MR \text{ und}$$

$$dTC/dQ = MC$$

$$d\pi/dQ = MR - MC$$

$$0 = MR - MC$$

$$MR = MC$$

Anmerkung: die TR entsprechen dem Preis mal der Menge, also $P * Q$.

Nach Q abgeleitet ergibt dies $dTR/dQ = P$. Somit können wir auch anstelle von $MR = MC$ auch $P = MC$ schreiben (Ch. Ewerhart, 2007, S.280).

A.4.4 Zusammenfassung

In dieser Lerneinheit haben Sie einen Einblick in die Kosten- und Produktionsfunktion erhalten.

Folgende Begriffe haben Sie in der Lerneinheit kennen gelernt:

Gewinn: Erlös minus Totalkosten

Gewinnmaximierender Preis entspricht den Grenzkosten.

A.5 Zusammenfassung

Folgende Themen wurden in diesem Modul behandelt:

Angebot und Nachfrage: Marktgleichgewicht, komparative Statik, Konsumenrentenrente, Nachfrage- und Angebotsüberschuss

Produktions- und Kostentheorie: marginale Kosten, Durchschnittskosten, Gewinn

A.6 Literatur

1. Beck, B. (2008). Volkswirtschaft verstehen 5. Aufl. Zürich: vdf Hochschulverlag AG an der ETH Zürich.
2. Ewerhart, Ch. (2007). Skript zur Vorlesung Mikroökonomie I an der Universität Zürich.
3. Hildmann, G. (2005). Mikroökonomie - Intensivtraining 5. Aufl. Wiesbaden: Betriebswirtschaftlicher Verlag Dr.Th.Gabler/GWV Fachverlag GmbH.
4. Löchel, H. (2003). Mikroökonomie - Haushalte, Unternehmen, Märkte 1. Aufl. Wiesbaden: Gabler.