

# Was machen Informatiker, und was ist Informatik?

## Great Principles and Practices of Computing

Es gibt viele Antworten auf diese Fragen. Eine mögliche Antwort ist die Sichtweise der Great Principles and Practices of Computing: Welche Prinzipien gibt es in der Informatik, und welche Kerntätigkeiten üben Informatiker aus. Wir starten mit einer kleinen und unvollständigen Tour d'Horizon.

» [Übersicht Great Principles and Practices of Computing \[SwissEduc\]](#)

## Practice: Programming

Programmieren ist eine der zentralen Aktivitäten von Informatikern, und wohl die bekannteste. Google Code Playground bietet die Möglichkeit, vorgefertigte JavaScript-Programme zu verwenden. Mit dem Playground zeigt Google Entwicklern anschaulich, wie sie die diversen Schnittstellen zu Google nutzen können. Die Programme reichen von einfachen Visualisierungen über Google Maps-Einbindung bis hin zur Konfiguration einer eigenen Google-Suche auf ausgewählten Websites. Die Programme können problemlos in eigenen Webseiten eingebunden werden.

» [Google Code Playground](#)

» [Google Code inkl. Dokumentation](#)

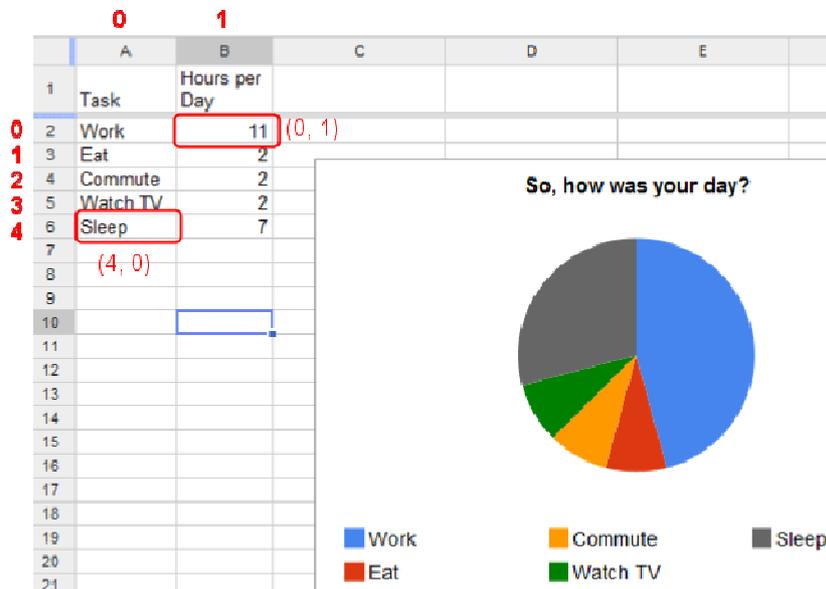
Als erstes Beispiel betrachten wir ein Program, mit dem wir auf einer Webseite ein Kuchendiagramm zeichnen können. Der grösste Teil des Programmtextes braucht uns im Moment nicht zu interessieren: Das ist einfach das "Drumherum", damit wir ein Diagramm erstellen können. Wir konzentrieren uns darauf, in dem Programm die Daten für das Diagramm anzugeben:

» [Kuchendiagramm mit Google Code Playground](#)

```
// zuerst beschreiben wir unsere Tabelle ("data") mit den Daten für das Kuchendiagramm
data.addColumn('string', 'Task'); // fügt Spalte hinzu
data.addColumn('number', 'Hours per Day'); // fügt Spalte hinzu
data.addRows(5); // fügt fünf leere Zeilen hinzu
```

```
// jetzt füllen wir Zellen in unserer Tabelle:
data.setValue(0, 0, 'Work'); // nullte Zeile, nullte Spalte
data.setValue(0, 1, 11); // nullte Zeile, erste Spalte
data.setValue(1, 0, 'Eat'); // erste Zeile, nullte Spalte
data.setValue(1, 1, 2);
data.setValue(2, 0, 'Commute');
data.setValue(2, 1, 2);
data.setValue(3, 0, 'Watch TV');
data.setValue(3, 1, 2);
data.setValue(4, 0, 'Sleep');
data.setValue(4, 1, 7);
```

Was dabei auffällt: Informatiker fangen bei Null an zu zählen. Die folgende Grafik zeigt zum Vergleich, wie die einzelnen Datenzellen in einer Tabellenkalkulation nummeriert sind:



### Aufgaben zu Kuchendiagramm

1 - Ändern Sie im [Kuchendiagramm](#) von Google Code Playground die Werte der Tabellenzellen (also zum Beispiel "Eat" zu "Drink", 11 zu 1 etc.), und aktualisieren Sie die Ansicht.

2 - Was passiert, wenn Sie ungültige Angaben für die Zellen eingeben, also zum Beispiel `data.setValue(0,2,"Work")`? Was sind andere mögliche ungültige Eingaben?

3 - Erstellen Sie ein eigenes Diagramm Ihrer Wahl.

### Was steckt dahinter?

#### Programmiersprache

Das Programm zeigt eine Anwendung der Programmiersprache JavaScript. Es gibt eine riesige Anzahl von Programmiersprachen; einige davon sind weit verbreitet im Einsatz (Java, C/C++/C#), die meisten sind nicht weit verbreitet, und einige wie zum Beispiel Brainfuck werden als "esoterisch" bezeichnet.

- » [Kleine Übersicht über Programmiersprachen \[SwissEduc\]](#)
- » [Brainfuck - Beispiel einer esoterischen Programmiersprache \[SwissEduc\]](#)

#### Datenstruktur Feld

Das Programm zeigt, wie in einem Programm Daten in Tabellen-Struktur verwaltet werden; der Fachbegriff für diese Tabellen ist Felder oder in Englisch Array. Arrays sind eine der grundlegendsten Arten, Daten in Programmen zu organisieren. Sie werden verwendet, um mehrere Daten gleicher Art zusammenzufügen, zum Beispiel eine Tabelle von Zahlen oder eine Tabelle von Zeichenketten.

- » [Datenstruktur Feld \[Wikipedia\]](#)

## Aufgaben zu Organigramm

Google Code bietet auch die Möglichkeit, Organigramme zu zeichnen. Auch Organigramme werden in Google Code Playground tabellarisch beschrieben.

» [Organigramme mit Google Code Playground](#)

Betrachten wir das Beispiel vom Playground:

```
data.addColumn('string', 'Name');
data.addColumn('string', 'Manager');
data.addRows(5);
data.setCell(0, 0, 'Mike');
data.setCell(1, 0, 'Jim');
data.setCell(1, 1, 'Mike');
data.setCell(2, 0, 'Alice');
data.setCell(2, 1, 'Mike');
data.setCell(3, 0, 'Bob');
data.setCell(3, 1, 'Jim');
data.setCell(4, 0, 'Carol');
data.setCell(4, 1, 'Bob');
```

Der Programmtext beschreibt folgende Tabelle:

Mike	
Jim	Mike
Alice	Mike
Bob	Jim
Carol	Bob

### Aufgaben

- 1 - Erweitern Sie die Tabelle, so Alice zwei Untergebene hat: Sarah und David.
- 2 - Fügen Sie ein zweites Organigramm in die gleiche Tabelle ein, mit Dumbledore an der Spitze, Snape und McGonagall ihm unterstellt (Pardon an alle Harry Potter-Fans).
- 3 - Erstellen Sie ein Organigramm Ihrer Wahl.

## Was steckt dahinter?

### Verweise

In tabellarischer Form beschreiben wir eine hierarchische Datenstruktur. Jede Zeile enthält einen Eintrag für eine Person (Spalte 0) und einem Verweis auf eine andere Person (Spalte 1). Diese Verweise erlauben es, Personen miteinander zu verknüpfen. Allgemein werden mit Verweisen (auch Referenzen, engl. Pointers oder References) Objekte miteinander verknüpft.

### Datenstruktur Baum

Mit Hilfe der Verweise beschreiben wir eine Datenstruktur, die Baum genannt wird. Die Wurzel ist dabei oben, und nach unten gehen die Äste weg, wobei jeder Ast weitere Unteräste haben kann. Bäume sind eine zentrale und sehr mächtige Datenstruktur. Sie erlauben es zum Beispiel, Daten sortiert zu halten oder in sortierten Daten sehr schnell zu suchen.

### Weitere Aufgaben

Erstellen Sie selbständig eine weitere Visualisation. Hier eine Auswahl möglicher Visualisationen:

» <http://code.google.com/apis/ajax/playground/#gauge> - Darstellung eines numerischen Wertes in Form eines Druckmessers

» <http://code.google.com/apis/ajax/playground/#sparkline> - einfache Liniendarstellung numerischer Werte

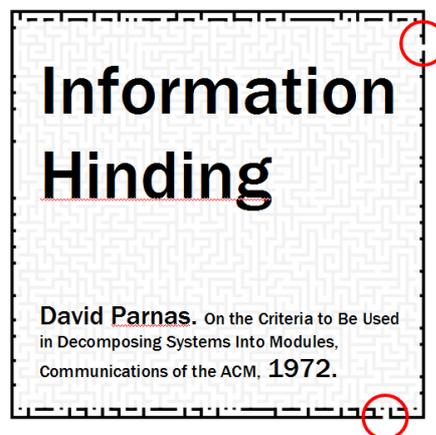
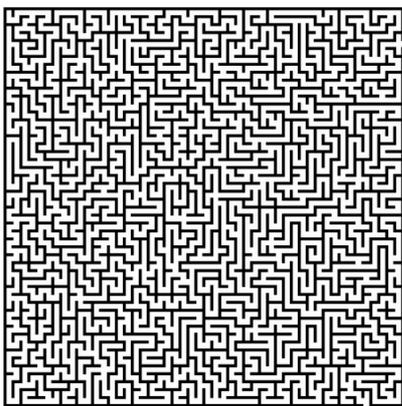
» [http://code.google.com/apis/ajax/playground/#geo\\_map](http://code.google.com/apis/ajax/playground/#geo_map) - Weltkarte mit farblicher Darstellung eines numerischen Wertes pro Land

» [http://code.google.com/apis/ajax/playground/#scatter\\_chart](http://code.google.com/apis/ajax/playground/#scatter_chart) - Visualisation mathematischer Funktionen

## Practice: Engineering Systems

Die meisten der obigen Beispiele (mit Ausnahme der Geo Map) verwenden zwar Google Code, sind aber in sich geschlossen: Die Programme beschreiben die darzustellenden Daten, der Google Code übernimmt die Darstellung. Das ist schon mal eine klare Aufgabentrennung und ermöglicht es uns, Grafiken auf Webseiten darzustellen, ohne uns darum kümmern zu müssen, wie das eigentlich im Detail bewerkstelligt wird.

Betrachten wir eine fiktive Post: Wie diese Post in dem Bild unten Pakete vom Eingang zum Ausgang transportieren würde, interessiert Kunden eigentlich nicht. Sie wollen nur Pakete verschicken. Welche Wege die Pakete nehmen, über welche Paketverteilungszentren die Pakete gehen, ist ihnen egal. Diese Details verbirgt die Post vor den Kunden. Das nennt sich in der Informatik *Information Hiding*, ein Konzept, das in den 1970er Jahren von David Parnas explizit gemacht wurde.



Die meisten Systeme entstehen in der Praxis durch das Zusammenspiel von verschiedenen beliebig komplexen Systemen. Damit das Zusammenspiel funktioniert, braucht es klare Spielregeln. Ein Beispiel aus der Küche: Verschiedenste Arten von Dosenöffner können verschiedenste Arten von Dosen öffnen, weil die Konservendose quasi standardisiert ist [[Wikipedia zu Konservendosen](#)]. Das funktioniert, weil die "Schnittstelle" klar definiert ist: Dosenhersteller wissen, wie der Rand einer Dose beschaffen sein müssen, damit Dosenöffner den Rand aufschneiden können, und Hersteller von Dosenöffnern wissen, wie stark die Schneider sein müssen.



In der Systementwicklung regeln ebenfalls sogenannte Schnittstellen das Zusammenspiel von Systemen. Betrachten wir als ein Beispiel, wie wir Google Maps in unsere eigenen Webseiten einbinden können. Wieder braucht es nur ein paar wenige Zeilen JavaScript, und schon können wir eine Google Map darstellen:

```
map.setCenter(new GLatLng(47.343947, 8.700026), 19);  
// Koordinaten von Kartenmitte; 19: Zoomlevel  
map.addControl(new GLargeMapControl()); // fuer unsere Zwecke nicht so wichtig  
var mapControl = new GMapTypeControl(); // fuer unsere Zwecke nicht so wichtig  
map.addControl(mapControl); // fuer unsere Zwecke nicht so wichtig  
var point = new GLatLng(47.343947, 8.700026); // Koordinate fuer Marker  
map.addOverlay(new GMarker(point)); // Marker hinzufuegen
```

Der Output des Programms ist eine Karte mit einer Markierung:



» <http://code.google.com/apis/ajax/playground/#markers> - Google Maps in Webseite einbinden mit eigenen Markern

» Koordinaten bestimmen: auf [Google Maps](#) zum Zielort navigieren, rechte Maustaste, im Menü "Was ist hier?" wählen, dann werden Koordinaten in Eingabefeld angezeigt

Das ist eigentlich schon beeindruckend: Wir müssen nur ein paar Angaben machen - Koordinaten, Zoomlevel - et voilà. Diese Schnittstelle ist sehr einfach. Wir sehen sozusagen nur die Spitze des Eisberges. Es braucht uns nicht zu kümmern, wie das Bild zustande kommt. Und da steckt ein grosser Eisberg dahinter: Google braucht massenweise Satellitenbilder, Flugzeugaufnahmen; alle Bilder müssen genauestens den Koordinaten auf der Weltkarte zugeordnet sein; gigantische Datenbanken, um diese Bilddaten zu verwalten; schnelle Server, damit die Benutzer nicht lange auf die Kartenbilder warten müssen; und so weiter.

## Aufgaben zu Google Maps

1 - Erstellen Sie ein Map mit Marker für eine Koordinate Ihrer Wahl. Verwenden Sie dazu bei der Visualisierung „Markers“ folgendes Programm zum Starten:

```
function initialize() {  
  if (GBrowserIsCompatible()) {  
    var map = new GMap2(document.getElementById("map_canvas"));  
    map.setCenter(new GLatLng(47.343947, 8.700026), 19);  
    // Koordinaten von Kartenmitte; 19: Zoomlevel  
    map.addControl(new GLargeMapControl()); // fuer unsere Zwecke nicht so wichtig  
    var mapControl = new GMapTypeControl(); // fuer unsere Zwecke nicht so wichtig
```

```
map.addControl(mapControl); // fuer unsere Zwecke nicht so wichtig
var point = new GLatLng(47.343947, 8.700026); // Koordinate fuer Marker
map.addOverlay(new GMarker(point)); // Marker hinzufuegen
}
}
```

2 - Fügen Sie einen zweiten Marker im gleichen Kartenausschnitt hinzu

3 - Erstellen Sie mit [http://code.google.com/apis/ajax/playground/#info\\_window](http://code.google.com/apis/ajax/playground/#info_window) eine Karte mit einem Informationstext

## Principle: Recollection

Eine der ersten Anwendungen von Computern war die Datenverarbeitung - daher auch eine der ursprünglichen deutschsprachigen Bezeichnungen, EDV für Elektronische Datenverarbeitung. Dazu gehört wesentlich das Verwalten von Daten: Ablage, Organisation und Wiederauffinden von Daten. Die Anwendungen dieses Prinzips erstrecken sich von der Organisation einer Festplatte durch das Betriebssystem über eine vom Benutzer angelegte Verzeichnisstruktur bis hin zu Datenbanken und dem Internet als gigantische Datenquelle.

Betrachten wir ein Beispiel. Wir suchen nach Informationen zum WM Gruppenspiel Schweiz-Spanien. Wir können dazu einfach mit Google suchen (linkes Bild). Wir haben Glück, dass Google Spass an der WM hat und speziell auf diese Anfrage reagiert. Die anderen Resultate zu der Anfrage "Schweiz Spanien" sind genauso unspezifisch wie die Anfrage selbst. Mit einer benutzerdefinierten Suche, die nur auf den [Sport-Seiten von Tages-Anzeiger, NZZ und 20 Minuten](#) sucht, finden wir hingegen zielführende Informationen (rechtes Bild).

» [Google Suche auf Sport-Seiten von Tages-Anzeiger, NZZ und 20 Minuten](#)



schweiz spanien

Search

About 25,900,000 results (0.18 seconds)

[Advanced search](#)

**Everything**  
[More](#)

**The web**  
[Pages from Switzerland](#)

**Any time**  
[Past 2 days](#)  
[More search tools](#)

Tip: [Search for English results only](#). You can specify your search language in [Preferences](#)

2010 FIFA World Cup™: Switzerland vs. Spain - [Add to iGoogle](#)

Spain **0 : 1** Switzerland - [Highlights](#)

Group H	MP	W	D	L	GF	GA	+-	Pts
Chile	2	2	0	0	2	0	+2	6
Spain	2	1	0	1	2	1	+1	3
Switzerland	2	1	0	1	1	1	0	3
Honduras	2	0	0	2	0	3	-3	0

[FIFA.com](#) - [Schedule](#) - [Standings](#) - [Switzerland Overview](#) - [Spain Overview](#)

[Schweiz - Spanien](#) ☆ - [ [Translate this page](#) ]

**Schweiz - Spanien**. Reiseroute. Sie fahren auf der Autobahn via Lausanne Richtung ...

Tanken Sie in der Schweiz vor Genf voll auf. In Spanien ist es dann ...

[www.feriencostabrava.ch/plan1.htm](#) - [Cached](#) - [Similar](#)

[Olé: Schweiz trifft auf Spanien, Chile und Honduras - WM 2010 ...](#) ☆ - [ [Translate this page](#) ]

5. Dez. 2009 ... Die **Schweiz** bekommt es an der WM 2010 in Südafrika mit drei lateinischen Gegnern zu tun. Unsere Nati wird der Gruppe H zugelost.

[www.blick.ch/sport/wm2010/die-auslosung-der-wm-gruppen-135035](#)

[Schweizerische Botschaft Madrid](#) ☆ - [ [Translate this page](#) ]

Schweizerschulen in **Spanien** · **Schweizer** Vereinigungen in **Spanien** · Sozialversicherungen und Krankenkasse · Lotteriebetrug · Electronic System for Travel ...

[www.eda.admin.ch > Vertretungen > Europa](#) - [Cached](#) - [Similar](#)

[YouTube - WM 2010: Schweiz - Spanien @ Gare de Lion, Wil](#) ☆

16 Jun 2010 ... public viewing im gare de lion! das spiel **schweiz - spanien** ist zu ende und die schweiz gewinnt 1:0!

[www.youtube.com/watch?v=H4O7qWewgPs&feature=related](#) - [Cached](#)

[Reisebericht Schweiz - Spanien](#) ☆ - [ [Translate this page](#) ]

Dann kommen wir nach **Spanien** rein. An der Grenze von Andorra nach **Spanien** gilt's erst mal anzustehen. D'Spaniögel wünds schinbar wüsse. ...

[www.e-prom.ch/Afrikapage/Web/europa.html](#) - [Cached](#) - [Similar](#)

[Tennisurlaub Österreich, Schweiz, Spanien - Tennis Academy in ...](#) ☆

Tennisurlaub in fünf einzigartigen Tennisschulen. Die European Tennis Academy Tennishotels in Österreich, der **Schweiz** und **Spanien** informieren über ...

[www.tennis-academy.com/en-partner\\_werden.shtml](#) - [Austria](#) - [Cached](#) - [Similar](#)

Google erlaubt es, sehr einfach benutzerdefinierte Suchen zu erstellen. Die Konfiguration für die obige Sport-Suche sieht wie folgt aus:

#### Steuerungsfeld - Websites: Sport Suchmaschine

**Eingeschlossene Websites** Anzeigen von 1 - 3 von 3

URL enthält:

- [nzz.ch/nachrichten/sport/\\*](#)
- [tagi.ch/sport/\\*](#)
- [20min.ch/sport/\\*](#)

---

**Ausgeschlossene Websites**

*Sie haben keine Websites ausgeschlossen. – [Websites ausschließen](#)*

Was erreichen wir mit einer eigenen benutzerdefinierten Suche? Wir schränken den riesigen Heuhaufen, den das Internet für Suchanfragen darstellt, ein auf den Bereich, der für uns relevant ist:



Das Einschränken einer Suche auf relevante Teilbereiche ist eines der grundlegenden Prinzipien der effizienten Informationsbeschaffung. Die meisten Suchanfragen sind zu unspezifisch und suchen in einem zu grossen Heuhaufen.

#### Aufgaben zu Googles benutzerdefinierten Suchmaschinen

1 - Erstellen Sie eine Sport-Suche, die auf den Sport-Seiten Ihrer Wahl sucht. Verwenden Sie dazu Ihr eigenes Google-Account oder [erstellen Sie ein neues Google Account](#) oder verwenden Sie das Account, das für Ihre Klasse zur Verfügung steht: aksa.efi.2010.rr AT gmail.com. Unter <http://www.google.com/cse/manage/all> können Sie die Suchmaschinen verwalten.

2 - Hat die Reihenfolge der "Eingeschlossenen Websites" einen Einfluss auf die Reihenfolge der Suchresultate?

3 - Erstellen Sie eine Suchmaschine, die alle Inhalte zum Beispiel von 20 Minuten berücksichtigt ausser den People- und Lifestyle-Beiträgen.

## Principle: Recollection – strukturierte Daten gezielt abfragen

In den ersten Beispielen waren die Daten jeweils direkt Teil des Programmes: Wir haben die darzustellenden Daten via Programm beschrieben. Das ist unschön, denn es vermischt Programm und Daten. Und es wird unpraktisch, wenn wir mit grösseren Datenmengen arbeiten. Sauberes Software Engineering heisst in diesem Fall, Daten und Programm zu trennen (*Separation of Concerns*).

Schauen wir uns als Beispiel die Google Visualization „Using the Query Language“ an. Für diese Visualisierung werden die Daten von einer Tabellenkalkulation in Google Documents geladen. Zudem kann genau spezifiziert werden, welche Daten geladen werden sollen und wie sie zum Beispiel sortiert werden sollen. Die Daten im Beispiel des Code Playgrounds sehen wie folgt aus:

	A	B	C	D
1		Country code	Population	Population Density
2	China	CN	1322970000	137
3	India	IN	1130130000	336
4	United States	US	303605941	31
5	Indonesia	ID	231627000	117
6	Brazil	BR	186315468	22
7	Pakistan	PK	162652500	198
8	Bangladesh	BD	158665000	1045
9	Nigeria	NG	148093000	142
10	Russia	RU	141933955	8.4
11	Japan	JP	127790000	339

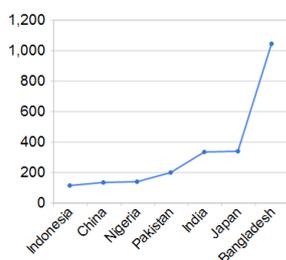
Im JavaScript-Code wird zunächst angegeben, von wo die Daten geladen werden sollen:

```
var query = new google.visualization.Query(  
  'http://spreadsheets.google.com/tq?key=pCQbetd-CptGXxxQIG7VFIQ&pub=1');
```

Anschliessend wird eine Abfrage spezifiziert:

```
query.setQuery('SELECT A,D WHERE D > 100 ORDER BY D');
```

Das bedeutet: Wähle die Spalten A (Land) und D (Bevölkerungsdichte) aus, und wähle dabei nur diejenigen Zeilen aus, bei denen der Wert in Spalte D > 100 ist. Sortiere anschliessend nach Spalte D aufsteigend. Basierend auf den Daten und der Abfrage erstellt Google eine neue Tabelle, welche die Daten wie spezifiziert enthält. Und diese Tabelle wird dann dargestellt, in diesem Fall als Liniendiagramm:



Auf diese Art erhalten wir eine unglaubliche Flexibilität: Die Originaldaten bleiben immer gleich. Aber mit den Abfragen können wir beliebig viele unterschiedliche Visualisierungen erstellen.

## Aufgaben zur Query Language

Die Website <http://www.gapminder.org/> liefert eine beeindruckende Fülle von Daten und Visualisierungen. Übrigens, ähnliche Grafiken können mit der Google Visualization „Motion Chart Time Formats“ auch selbst erstellt werden. Für die folgende Übung verwenden wir aber lediglich Daten von Gapminder. Wir wollen zum Beispiel die „Total number of billionaires“ visualisieren (siehe Link auf der Seite <http://www.gapminder.org/data/>). Die zugehörigen Daten sehen wie folgt aus:

	A	B	C	D	E
1		Jahr 2004	Jahr 2005	Jahr 2006	Jahr 2007
2	Abkhazia	0	0	0	0
3	Afghanistan	0	0	0	0
4	Akrotiri and Dhekelia	0	0	0	0
5	Albania	0	0	0	0

Sie finden die Daten unter:

<https://docs.google.com/spreadsheet/pub?key=0AjfyEQvnd95dGNRXzVtUDIWU2d5NkJNaWxIT1owdEE&output=csv>

1 – Erstellen Sie eine Visualisierung aller Länder und der Anzahl Milliardäre des Jahres 2007, sortiert nach Ländernamen. Falls die Ansicht zu klein ist, verwenden Sie den Button „Edit HTML“ (oben rechts). Ganz unten im HTML können Sie die width angeben (vordefiniert sind 400px, probieren Sie es mit 1000px).

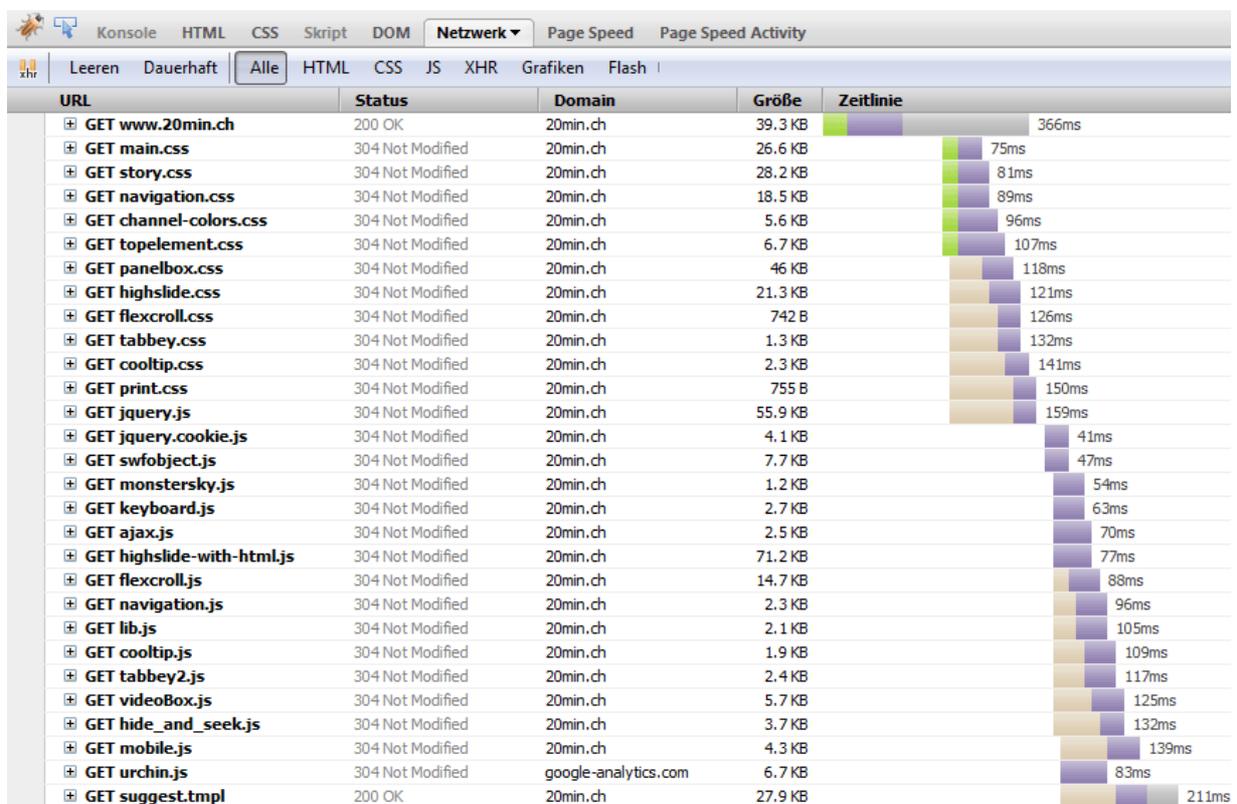
2 – Zeigen Sie die 50 Länder mit den meisten Milliardären des Jahres 2007 an. Dazu müssen Sie absteigend nach Spalte E sortieren (ORDER BY E DESC, wobei DESC für descending, also absteigend, steht). Und Sie müssen die Resultattabelle auf 50 Länder beschränken. Ergänzen Sie die Abfrage dazu mit LIMIT 50.

3 – Erstellen Sie ein Balkendiagramm statt eines Liniendiagramms: BarChart statt LineChart. Beeindruckend ist auch ein Kuchendiagramm (PieChart), wo man auf einen Blick die relative Anzahl Milliardäre erkennt.

## Principle Evaluation

Wie kann sichergestellt werden, dass ein Telefonnetz fast immer genügend Kapazität hat? Wie kann Google seine Resultate so schnell anzeigen? Wie viel Bandbreite braucht eine Firma für ihre Internet-Anbindung? Warum dauert zum Beispiel das Laden von [www.20min.ch](http://www.20min.ch) so lange?

Als konkretes Beispiel betrachten wir die letztere Frage genauer. Für Website-Betreiber ist die Frage, warum das Laden ihrer Webseiten wie lange dauert, von grossem Wert. Das Firefox-Plugin [Firebug](#) stellt auf einer Zeitachse die Ladezeiten für alle Elemente einer Webseite (Webseite selbst, Bilder, weitere verwendete Dateien) dar. Die folgende Grafik zeigt die Auswertung von [www.20min.ch](http://www.20min.ch):



Die Grafik zeigt nur einen kleinen Ausschnitt der Ladezeiten aller Bestandteile einer 20min-Webseite.