1.– 4. September 2014
in Nürnberg

Herbstcampus

Wissenstransfer
par excellence

Data-Driven Documents

Infographiken für Jedermann in D3

Axel Feix

Itech Progress GmbH

D3 – Data Driven Documents
Infografiken für Jedermann mit D3

V1.1a

Axel Feix ITech Progress GmbH

Data-Driven Documents

Infografiken für Jedermann mit D3

- Haben Sie nicht auch schon ab und an einmal neidisch auf eine Infografik im „Spiegel“, der „New York Times“ oder in „Die Zeit“ geschaut und im Stillen bei sich gedacht: das würde ich auch gerne können – am besten sogar auf meiner Homepage? Die gute Nachricht ist: das geht jetzt. Mit D3 sind komplexe Infografiken ohne komplexe Programmierung möglich – wenn man weiss wie. Data Driven Documents, kurz D3, ist ein JavaScript-Framework für die einfache Erzeugung von komplexen Infografiken auf HTML-Seiten.
- Dieser Vortrag beleuchtet die Philosophie, die technischen Prinzipien und den Aufbau der API, und erläutert an mehreren aussagekräftigen Beispielen die Fähigkeiten dieses schönen und spannenden Frameworks.

Wer bin ich?

Axel Feix

Senior-Consultant der ITech Progress GmbH

Trainer und Berater mit langjähriger Projekterfahrung als Softwarearchitekt

ITech Progress GmbH

Kennzahlen	
Hauptsitz	ITech Progress GmbH Donnersbergweg 4 67059 Ludwigshafen
General Manager	Mahbouba Gharbi
Mitarbeiter/-innen	29
Kontakt	Tel.: +49 621 59 57 02 - 0 Fax: +49 621 59 57 02 – 99 contact@itech-progress.com
Homepage	www.itech-progress.com

General Manager
Mahbouba Gharbi

Hauptsitz
Ludwigshafen am Rhein

- Wir bieten umfassende Erfahrung, mit Kompetenz in den wichtigsten Industrien, von der Konzeption bis zur Umsetzung entlang der gesamten Wertschöpfungskette.
- Dank unseres herstellerunabhängigen Ansatzes bieten wir unseren Kunden technologisch optimierte Lösungen.

Übersicht

Eine kurze Einführung in Datenvisualisierung

Was ist D3?

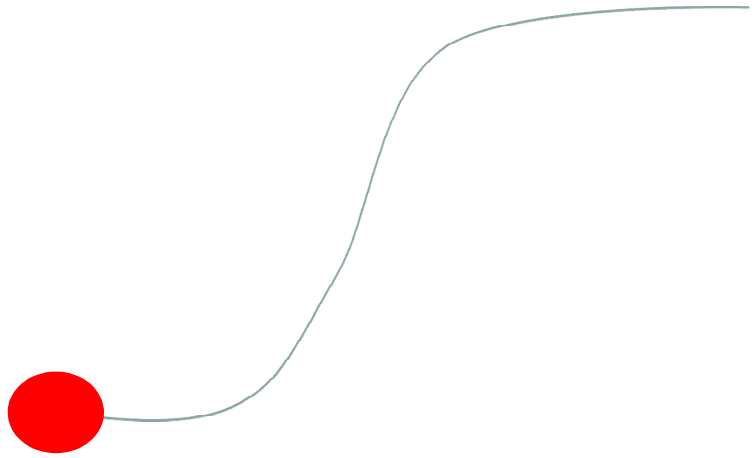
Erste Schritte in D3

Beispiel 1: Liniendiagramm

Beispiel 2: Baum

Beispiel 3: Geokomponenten

Ausblick: Wie geht es weiter?



ITECH PROGRESS

Haben sie eigentlich schon einmal was von

gehört ?

ITECH PROGRESS

Worum geht es bei D3?

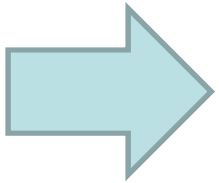
D^3 = Datenvisualisierung

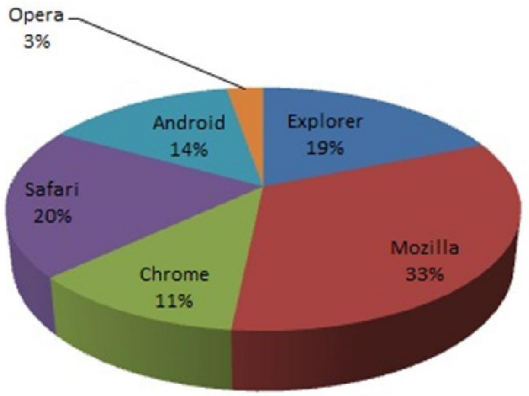
Fünf wichtige Fragen

- Warum Datenvisualisierung?
- Warum Computer zur Visualisierung nutzen?
- Warum interaktiv?
- Warum Web?
- Warum D3 nutzen?

Warum Informationsvisualisierung?

- Bilder sagen mehr als tausend Worte





Zahlentabellen vs. Diagramme

Warum Informationsvisualisierung?

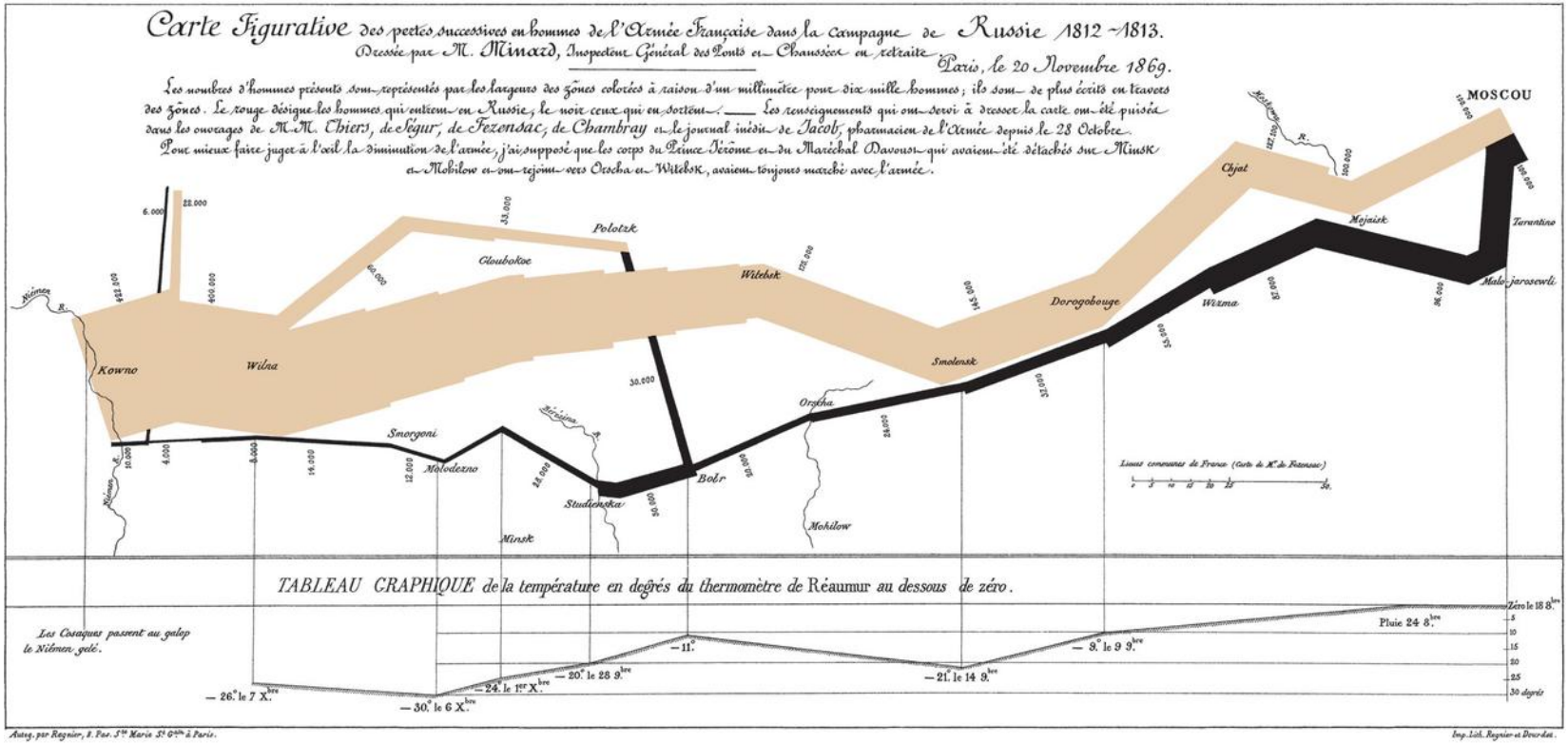
Ich sehe – und ich erkenne

Ich höre – und ich verstehe

Ich erfahre – und ich begreife

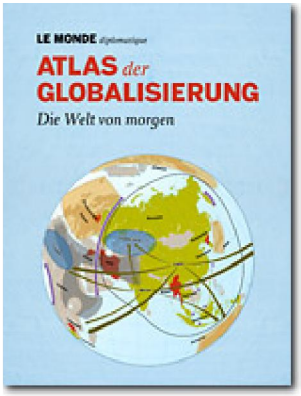
Ein Klassiker: Charles Joseph Minard

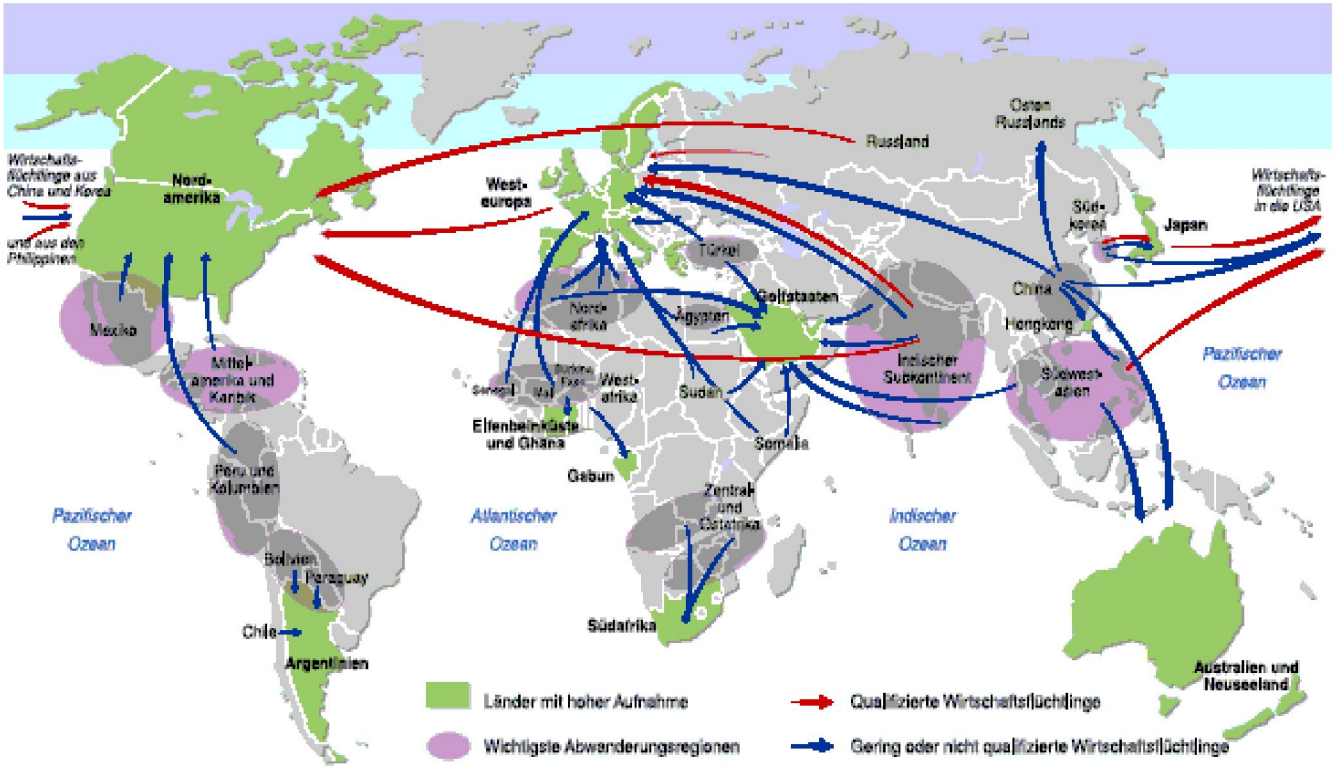
- Napoleons Russlandfeldzug 1812/1813



Warum Informationsvisualisierung?

- Infografik Globale Migration

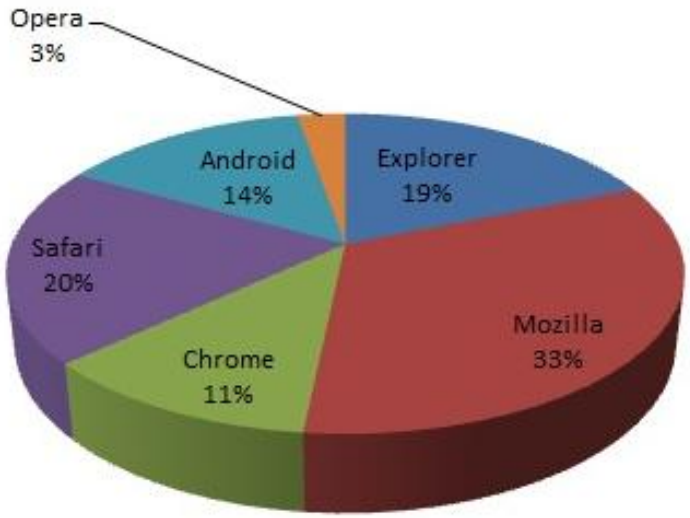




Quelle: Gildas Simon, Géodynamiques des migrations internationales dans le monde, Presses universitaires de France (PUF), Paris, 1995; Courrier de l'Unesco, Paris, November 1998; CNRS-université de Poitiers, Migrinter (Migrations internationales, espaces et sociétés)

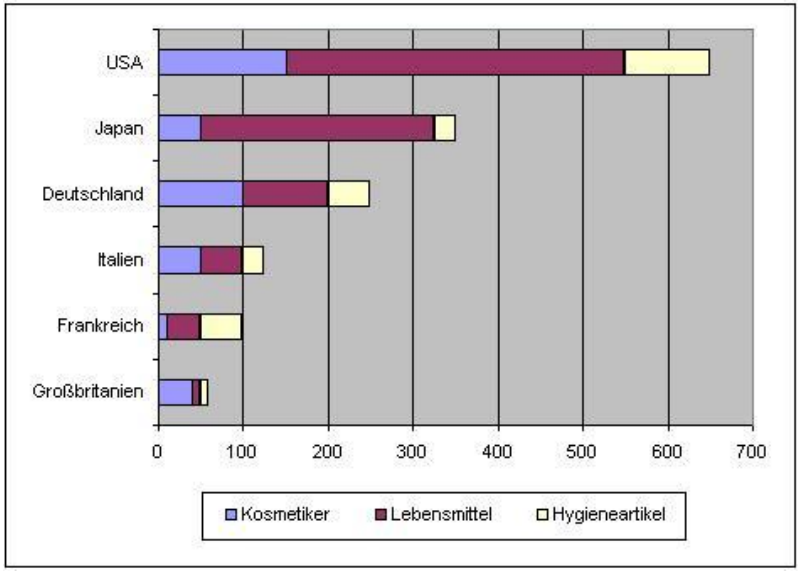
Warum Computer benutzen?

- Bisherige Infografiken



Gesamtumsatz in einem Land

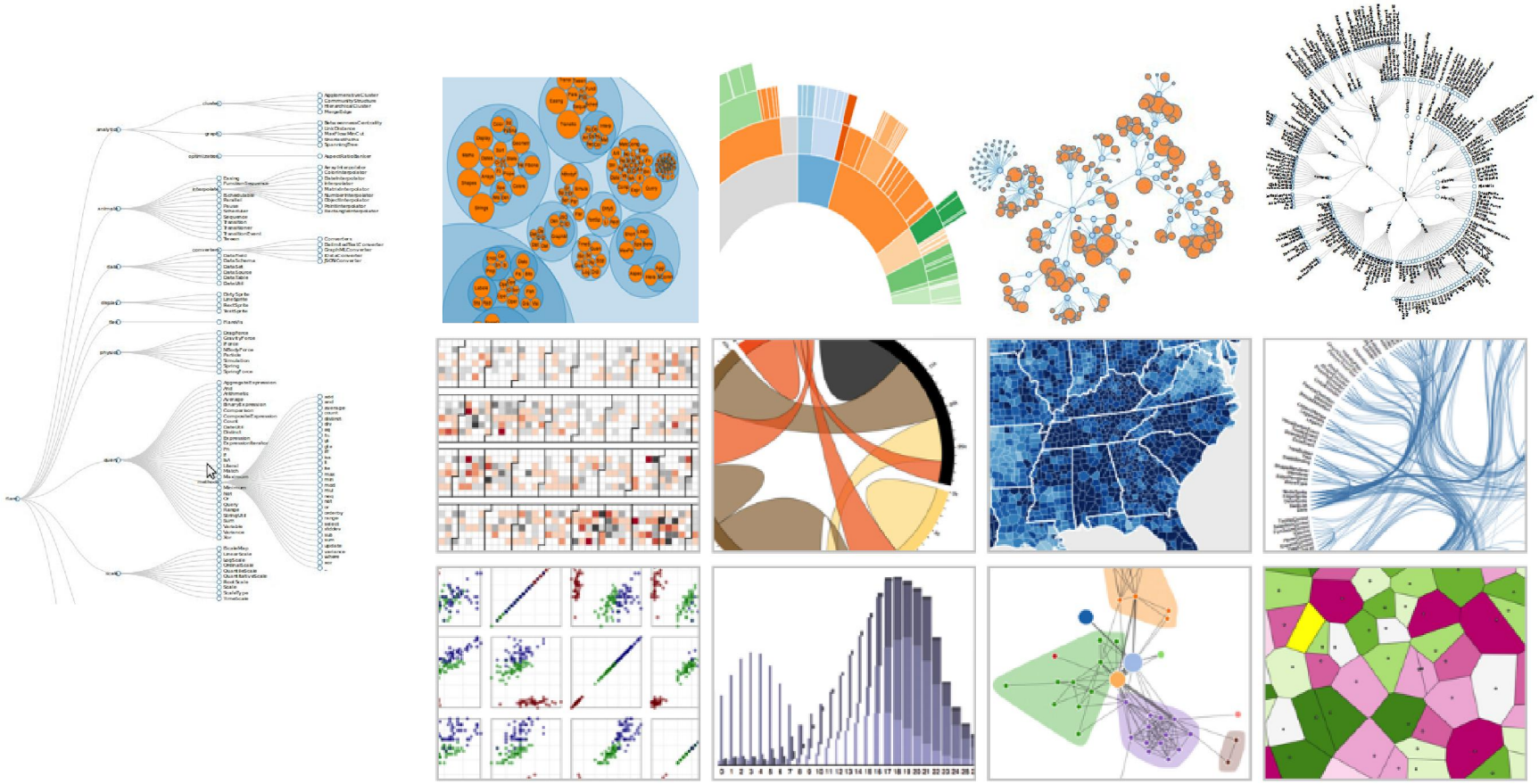
	Kosmetiker	Lebensmittel	Hygieneartikel
Großbritanien	40 T €	10 T €	10 T €
Frankreich	10 T €	40 T €	50 T €
Italien	50 T €	50 T €	25 T €
Deutschland	100 T €	100 T €	50 T €
Japan	50 T €	275 T €	25 T €
USA	150 T €	400 T €	100 T €



Warum interaktiv und Web?

- Beispiel von Spiegel Online (http://www.spiegel.de/politik/deutschland/bundeswehr-luftwaffe-ist-bei-transportflugzeugen-stark-eingeschraenkt-a-987712.html)

Warum D3 benutzen?



- Beispiel: http://vallandingham.me/vis/movie/

Übersicht

Eine kurze Einführung in Infovisualisierung

Was ist D3?

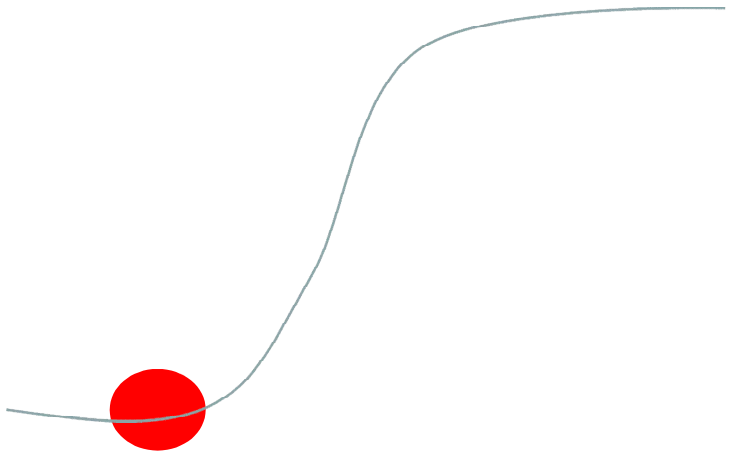
Erste Schritte in D3

Beispiel 1: Liniendiagramm

Beispiel 2: Baum

Beispiel 3: Geokomponenten

Ausblick: Wie geht es weiter?



Was ist D3?

- Ist eine Javascript Bibliothek (d3.js)
- Ist ein mächtiges Werkzeug
  - Zur Manipulation von Daten
  - Zur interaktiven Datenvisualisierung
- Hat mehr als 50 eingebaute Diagramm- und Layouttypen
- Ist BSD lizensiert (BSD-new)
- Ist minimiert gerade einmal 127KB groß
- Enthält hunderte Funktionen
- Die Entwicklung wird gesponsert von der New York Times

ITECH PROGRESS

Was ist D3 nicht?

- Ist weder einfach noch intuitiv
- Ist nicht in ein paar Minuten erlernbar
- Hat keine flache Lernkurve
- Erzeugt keine vorgefertigten Visualisierungen für Sie
- Unterstützt keine alten Browser
- Arbeitet nicht mit Bitmaps, sondern mit Vektorgrafiken
- D3 kann keine Daten verbergen

Der Weg zu D3

1996	Navigator	Netscape führt ersten Browser mit JS ein
|		
2005	Prefuse	Jeffrey Heer, Stuart Card, James Landay (Toolkit DV im WEB in Java, Berkley)
|		
2007	Flare	Jeff Heer (Datenvisualisierung mit Flash und Actionscript mit PlugIn, Berkley)
|		
2009	Protovis	Jeff Heer, Michael Bostock (JavaScript)
|		Stanfort
2011	D3	Mike Bostock, Vadim Ogievetsky, Jeff Heer (JavaScript), Stanfort

Mike Bostock
mbostock

Jeffrey Heer

Vadim Ogievetsky

Wie arbeitet D3?

- D3 ist ein elegantes Stück Software das die Erzeugung und Manipulation von Daten in Webdokumenten erlaubt
- Dies macht es mittels
  - Laden der Daten in den Speicher des Browsers
  - Verbinden von Daten und Elemente im Dokument und Erzeugung beliebig vieler Elemente
  - Transformation der Elemente durch Interpretation
    - Der Daten und Settings
    - seiner visuellen Eigenschaften
    - Transitionierung von Elementen zwischen Zuständen ausgelöst durch Benutzereingaben

Was ist bei D3 alles enthalten?

d3.js:	D3.js Kern, Selektionen, Transitionen, Skalierung, Array-/ String-Operationen Formatierungen, SVG Shape Generator, Drag & Drop, Zoom etc.
d3.chart.js:	Charts
d3.csv.js:	CSV Parsing, CSV Processing etc.
d3.geo.js:	Geoprojektionen, Pfadoperationen etc.
d3.geom.js:	Konturenpolygone für Vertices als Eingabe, Voronoi Diagramme, Delaunay Triangulation, konvexe Hüllen etc.
d3.layout.js:	Algorithmen, Layouts für Diagramme, Graphen, Dendrograme, Histogramme etc.
d3.time.js:	Zeitformatierung, Zeitintervalle etc.

Was verwendet D3?

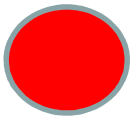
verwendete Web-Standards:

- HTML - Hypertext Markup Language
- CSS - Cascading Style Sheets
- JS - JavaScript
- SVG - Scalable Vector Graphics
- DOM - Document Object Model
- JSON - JavaScript Object Notation

SVG Elemente

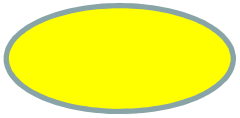
```
<rect
x="50"
y="20"
width="150"
height="100"/>
```

x: X-Koordinate der linken oberen Ecke
y: Y-Koordinate der linken oberen Ecke
width: Breite des Rechtecks
height: Höhe des Rechtecks



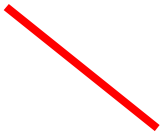
```
<circle
cx="100"
cy="50"
r="40"/>
```

cx: X-Koordinate des Mittelpunkts
cy: Y-Koordinate des Mittelpunkts
r: Radius



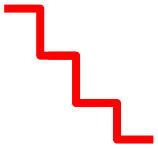
```
<ellipse
cx="300"
cy="80"
rx="100"/>
ry="50"/>
```

cx: X-Koordinate des Mittelpunkts
cy: Y-Koordinate des Mittelpunkts
rx: Radius in X-Richtung
ry: Radius in Y-Richtung



```
<line
x1="0"
y1="0"
x2="200"/>
y2="200"/>
```

x1: X-Koordinate Startpunkt
y1: Y-Koordinate Startpunkt
x2: X-Koordinate Endpunkt
y2: Y-Koordinate Endpunkt



```
<polyline
points="0,40,40,40,40,
80,80,80,80,120,120,1
20,120,160"/>
```

points: X- und Y-Koordinaten der Punkte des Linienzugs

```
<polygon
points="200,10 250,190
160,200"/>
```

points: X- und Y-Koordinaten der Eckpunkte des Polygons

```
<path
d="M 150 0 L75 200 L225
200 Z"/>
```

d: Beschreibung durch einfache Befehle
M = moveto (Cursor bewegen)
L = lineto (Linien Cursor -> Position)
H = horizontales Lineto
V = vertikales Lineto
C = curveto (Kurve)
S = smooth Curveto
Q = quadric Bezier curve
T = smooth quadric Bezier curve
A = elliptical arc
Z = closepath (Figur schliessen)

SVG Text

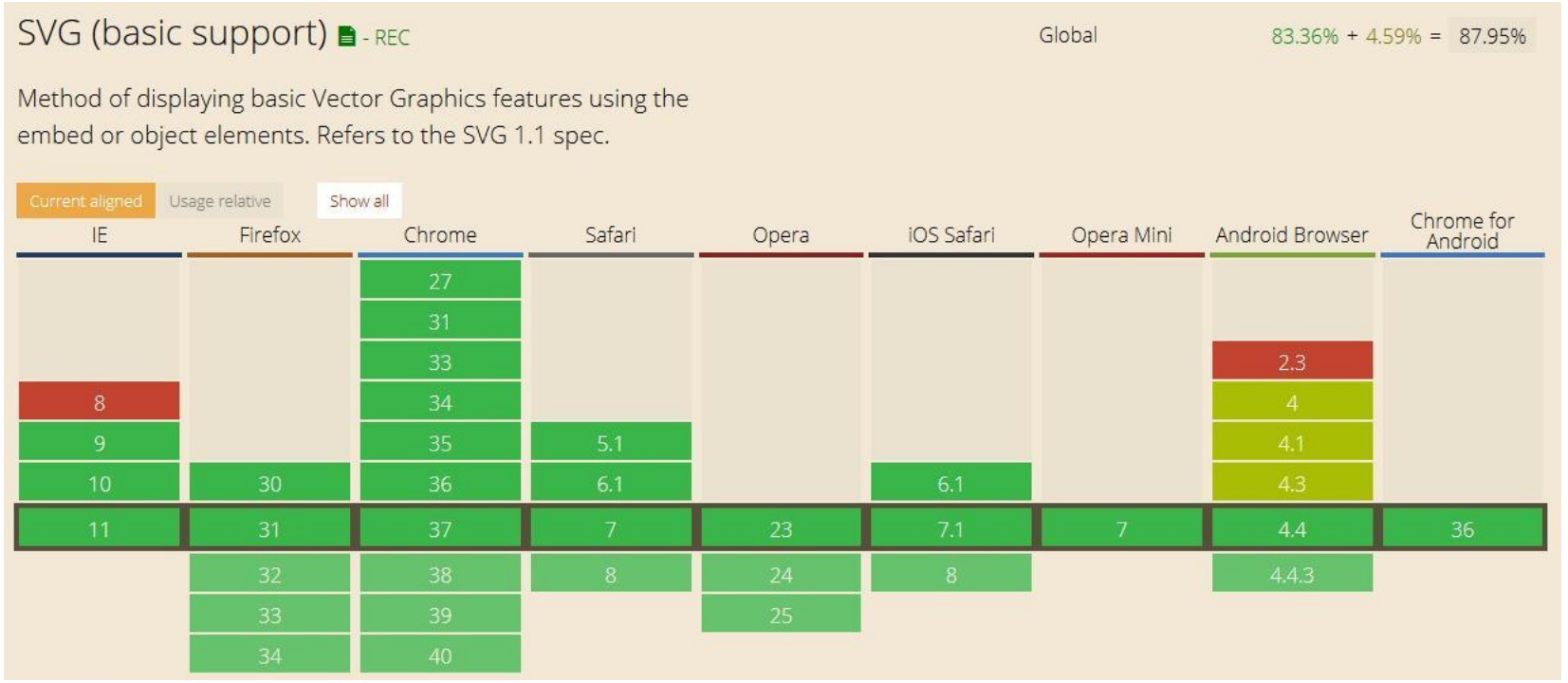
```
<text
x="0" y="15"
dy="1" dy ="5"
text-anchor = "start">
SVG Text
</text>
```

x: X-Koordinate Bezugspunkt (BP)
y: Y-Koordinate Bezugspunkt (BP)
dx: Abweichung von der X-Koordinate
dy: Abweichung von der Y-Koordinate
text-anchor: Verhältnis des Texts zum BP

```
<g id="gruppe1" fill="red"
>
<rect ... /> <rect ... />
</g>
<g id="gruppe2" fill="blue"
>
<rect ... /> <rect ... />
</g>
```

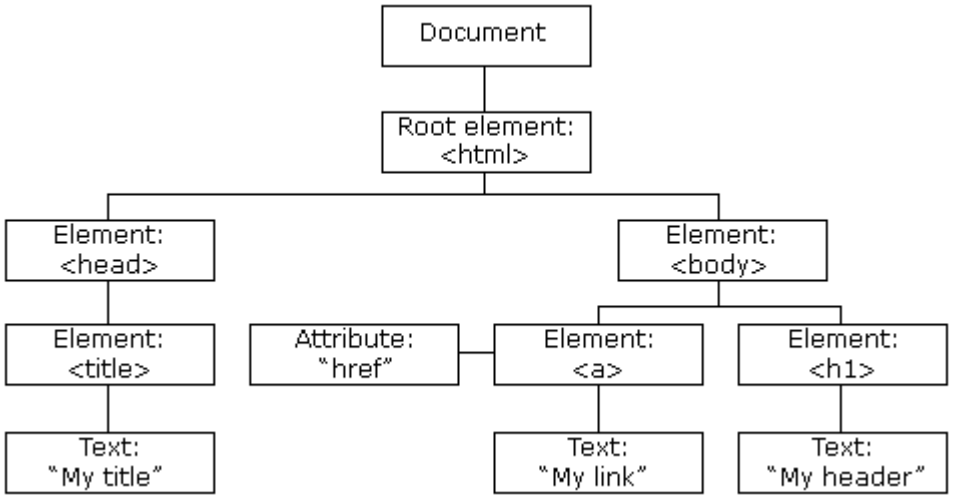
g: Gruppieren von Elementen, die gleiche Attribute haben sollen.

SVG Verträglichkeit: welcher Browser kann was?



D3 manipuliert den DOM-Tree

- DOM – Document Object Model
- Wird als Szenengraph genutzt

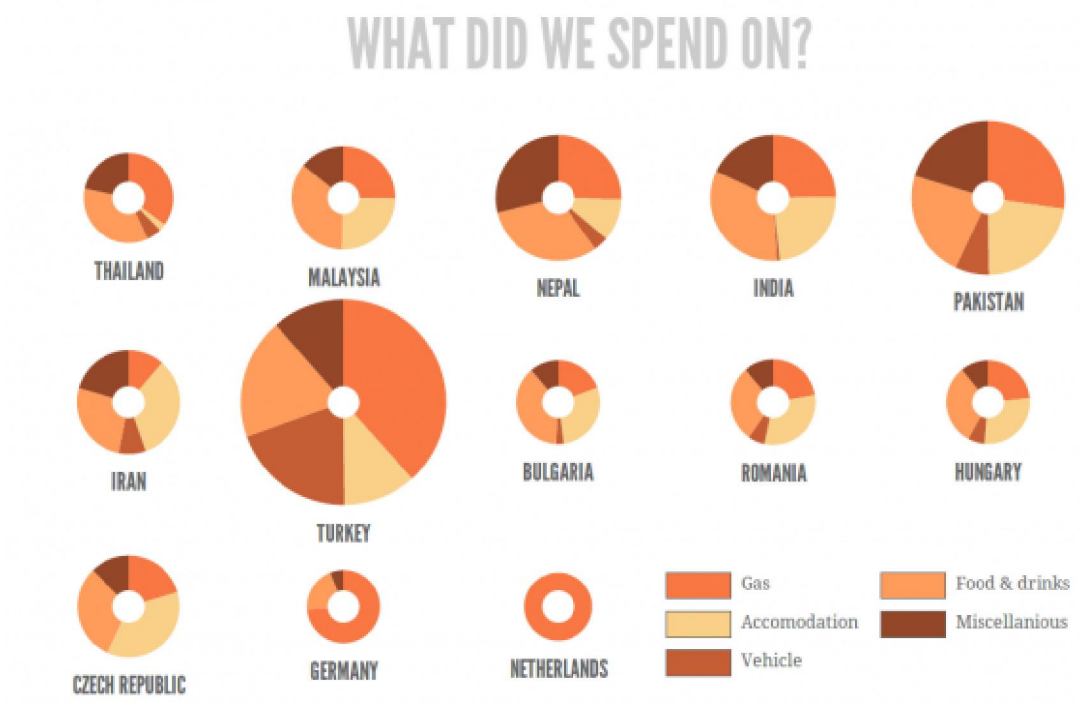


D3 enthält hunderte Funktionen, die gruppiert werden können in:

- Transitions,
- Arrays,
- Math,
- Color,
- Scales,
- SVG,
- Time,
- Layouts
- Geography,
- Geometry,
- Behaviors

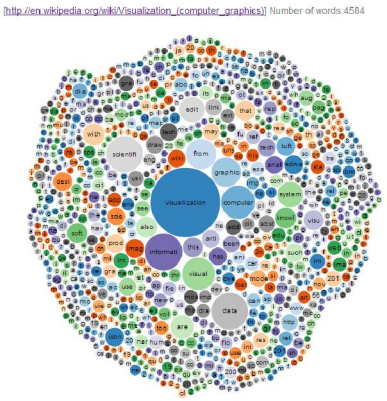
Ein kleines Beispiel: Minitruck Eurasia

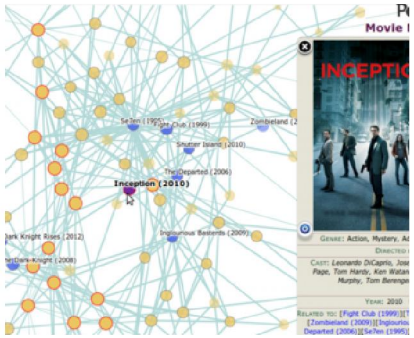
- http://joostkiens.com/project/driving-from-thailand-to-the-netherlands-datavis-with-d3-is/

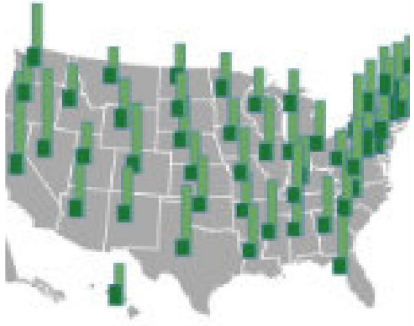


http://demo.joostkiens.com/mtea-data/

Willkommen auf der Spielwiese







- Bubblediagramm (http://www.infocaptor.com/bubble-my-page)
- Movie Network (http://bl.ocks.org/paulovn/9686202)
- Wordcloud (http://www.jasondavies.com/wordcloud/#%2F%2Fwww.jasondavies.com%2Fwordtree%2Fcat-in-the-hat.txt)
- Karte und Balkendiagramm kombiniert (https://vida.io/discussion/s5qo5Gwrct5HNxAD2)

Willkommen auf der Spielwiese

http:///d3js.org/
https://github.com/mbostock/d3/wiki/Gallery

Übersicht

Eine kurze Einführung in Datenvisualisierung

Was ist D3?

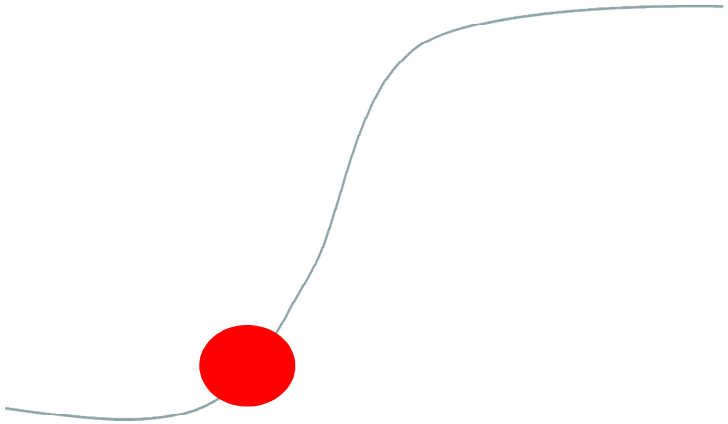
Erste Schritte in D3

Beispiel 1: Liniendiagramm

Beispiel 2: Baum

Beispiel 3: Geokomponenten

Ausblick: Wie geht es weiter?



ITECH PROGRESS

Was braucht man für D3?

- Einen Texteditor (mit JS Syntaxhighlighting)
- Die D3 Bibliothek
- einen Webbrowser
- einen Webserver

Webserver und Webbrowser

- Webserver
  - d3 kann man meist auch lokal im Webbrowser ausgeführen
  - Webserver um zur Laufzeit Daten zu lesen (XMLHttpRequest)
  - Es gibt viele Möglichkeiten
    - EasyPHP (windows), Mac OS X Server, MAMP (Mac OS X)
    - Python
- Webbrowser
  - Webbrowser mit Console
    - Internet Explorer F12
    - Mozilla Zusatz -> Webentwickler -> Console
    - Chrome

Template für D3

```
<!DOCTYPE html>
<html lang="en">
<head>
	<meta charset="utf-8">
	<title>Hello World in D3</title>
	<script type="text/javascript"
src=„d3/d3.v3.js"></script>
</head>
<body>
	<script type="text/javascript">
		// hier steht ihr wunderbarer D3 Code
	</script>
</body>
</html>
```

Hallo Welt gesamt

http://localhost:8888/G01_HelloWorld.html

```
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="Content-Type" content="text/html;charset=utf-8">
      <title>Mein Projekt</title>
      <script type="text/javascript" src=„d3.v3.js"></script>
      <style> <!--- hier ggf. CSS →</style>
</head>
<body>
     <div id=„grafik"></div>
     <script type="text/javascript">
              var w=960,h=500;
              var svg=d3.select("#grafik")
                 .append("svg")
                 .attr("width",w).attr("height",h);
              svg
                 .append("text")
                 .text("hello world!")
                 .attr("x",100)   .attr("y",100);
     </script>
</body>
</html>
```

Was braucht man für D3?

- Vorbereitungsschritte für D3

1. Python installieren
2. D3.js Bibliothek in UV D3 installieren
3: Texteditor mit Syntax-Highlighting
4: C:\d3 erstellen
5: unter C:\d3\lib\ die Files d3.js und d3....js ablegen

Was braucht man für D3?

Vorgehensweise zur Übersetzung und Deployment

Quelltexte unter C:\d3\ ablegen

1. mit [win]+[D] auf den Desktop wechseln
2. mit [win]+[R] Ausführungsfenster öffnen
3. cmd eintippen und eine Console starten
4. mit cd \d3 in das Verzeichnis wechseln
5. mit **python -m http.server 8888** einen einfachen Webserver mit Python3 starten
6. in Webbrowser http://localserver:8888

Erste Schritte in D3: selectAll()

Einführung selectAll (deklarativer Ansatz)

- - erlaubt es alle Elemente mittels einer Bedingung zu selektierten um alle auf einmal zu ändern

```
d3.selectAll("p").style("font-weight","bold");
```

- Anonyme Funktionen;

```
var fs= ["10px","20px","30px"];
d3.selectAll("p")
        .data(fs)
        .style("font-size",function(d) {return d;})
```

- der Name der anonymen Funktion ist aus Konventionsgründen "d"

Erste Schritte in D3: selectAll() Beispiel 1

http://localhost:8888/G02b_AuswahlMitD3.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="d3/d3.v3.js"></script>
    <title>Selektieren via d3.js</title>
  </head>
  <body>
    <h3>Bevorzugte Automarken:</h3>
    <ul>
      <li>Audi</li>
      <li>Volvo</li>
      <li>Mercedes</li>
    </ul>
    <script type="text/javascript">
      d3.selectAll("li").style("color", "red");
    </script>
  </body>
</html>
```

Erste Schritte in D3: selectAll() Beispiel 2

http://localhost:8888/G05_SelektionMitEinfaerben.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="d3/d3.v3.js"></script>
    <title>Selektieren via d3.js (dynamische Properties)</title>
  </head>
  <body>
    <h3>Bevorzugte Automarken:</h3>
    <ul>
      <li>Audi</li>
      <li>Volvo</li>
      <li>Mercedes</li>
     </ul>
    <script type="text/javascript">
      d3.selectAll("li").style("color", function(d, i) {
        return i % 2 ? „blue" : "red";
      });
    </script>
  </body>
</html>
```

Erste Schritte in D3: selectAll() Beispiel 3

http://localhost:8888/G06_SelektierenViaD3.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="d3/d3.v3.js"></script>
    <title>Selektieren via d3.js (dynamische Properties)</title>
  </head>
  <body>
    <h3>Bevorzugte Automarken:</h3>
    <ul>
      <li>Audi</li>
      <li>Volvo</li>
      <li>Mercedes</li>
</ul>
    <script type="text/javascript">
      d3.selectAll("li")
        .data([32, 12, 24])
        .style("font-size", function(d) { return d + "px"; });
    </script>
  </body>
</html>
```

Erste Schritte in D3: enter()

Einführung enter

```
d3.select("body").selectAll("p").data(data).enter()
```

erzeugt "pods" (Hüllen) für zukünftige Elemente

```
....selectAll("p").data(data).enter().append("p")
```

- Die Funktion append erzeugt so viele Elemente wie benötigt werden im Gegensatz zu nur einem Element
- ein d3.selectAll("p") funktioniert nicht wenn die Elemente noch nicht existieren
- stattdessen nimmt man einen Container zur Anlage der neuen Elemente

```
d3.select("body").selectAll("p").data(…).enter()…
```

Erste Schritte in D3: enter()

http://localhost:8888/G07_AddDOMNodesMitEnter.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="d3/d3.v3.js"></script>
    <title>Hinzufügen neuer DOM Knoten mittels enter()</title>
  </head>
  <body>
    <h3>Bevorzugte Automarken:</h3>
    <ul>
      <li>…. </li>
    </ul>
    <script type="text/javascript">
      d3.select("ul").selectAll("li")
          .data(["Audi", "Volvo", "Mercedes", "Volkswagen", "Ford", "Toyota"])
          .enter().append("li")
          .text(function(d) {return d});
    </script>
  </body>
</html>
```

Erste Schritte in D3: remove() und exit()

- auf jede Selection kann auch ein remove() angewendet werden, das dann alle Elemente löscht

```
d3.selectAll("p").remove()

d3.selectAll("p").data(["hello world"]).html(String);
```

- exit Methode

```
d3.selectAll("p").data(["hello world"]).exit()
// do stuff to those, often .remove()
```

Erste Schritte in D3: Transformation/Transitionen

http://localhost:8888/G09_Transitionen.html

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8">
    <script type="text/javascript" src="d3/d3.v3.js"></script>
    <title>D3 und SVG</title>
  </head>
  <body>
    <script type="text/javascript">
      var canvas = d3.select("body").append("svg:svg")
        .attr("width", 640).attr("height", 480);
      canvas.append("svg:circle")
        .attr("cx", 100)
        .attr("cy", 100)
        .attr("r", 50);
      .transition()
          .duration(1000)
          .attr("r", 1e-6);
    </script>
  </body>
</html>
```

Übersicht

Eine kurze Einführung in Infovisualisierung

Was ist D3?

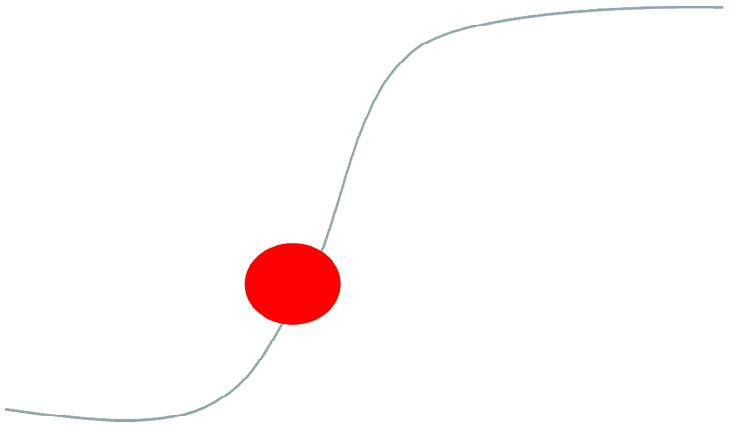
Erste Schritte in D3

Beispiel 1: Liniendiagramm

Beispiel 2: Baum

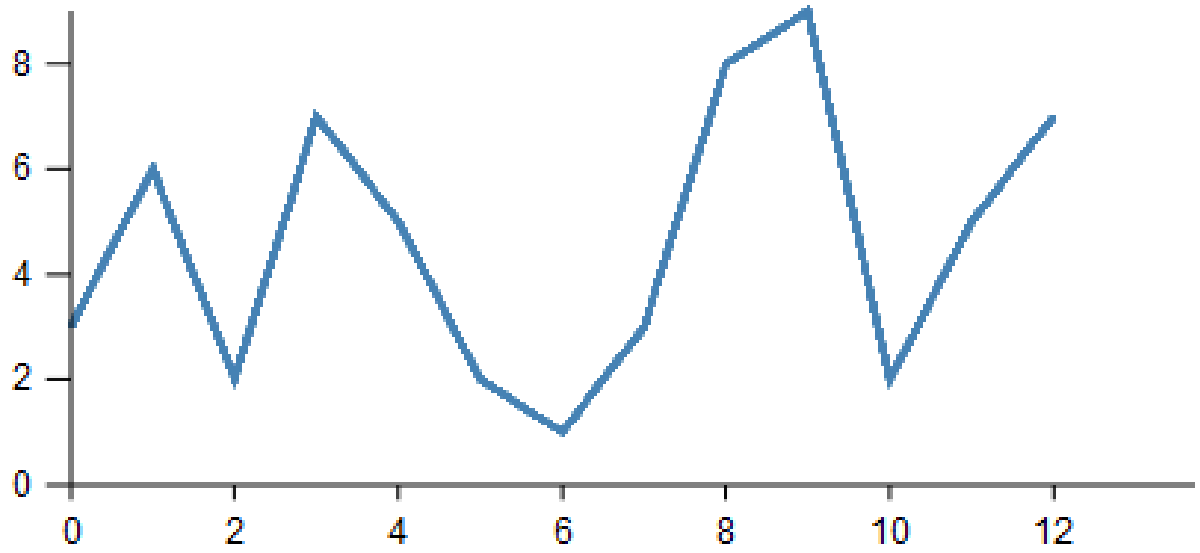
Beispiel 3: Geokomponenten

Ausblick: Wie geht es weiter?



Liniendiagramm in D3

- Das Endergebnis:



In diesem Tutorial erzeugen wir ein Liniendiagramm mit x- und y-Achse, Markierungen und Labels

Liniendiagramm in D3

- Schritt 0: Festlegen der zentralen Steuerung für das Diagramm

```
path {
       stroke: steelblue;
       stroke-width: 2;
       fill: none;
}
line {
       stroke: black;
}
text {
       font-family: Arial;
       font-size: 9pt;
}
```

Liniendiagramm in D3

- Zuerst beschreiben wir ein paar Variablen

```
var data = [3, 6, 2, 7, 5, 2, 1, 3, 8, 9, 2, 5, 7],
w = 400,
h = 200,
margin = 20,
y = d3.scale.linear().domain([0, d3.max(data)]).range([0 +
margin, h - margin]),
x = d3.scale.linear().domain([0, data.length]).range([0 +
margin, w - margin])
```

Liniendiagramm in D3

- Schritt 2: SvG Element in Dokument einfügen und Ursprung nach unten transmieren

```
var vis = d3.select("body")
            .append("svg:svg")
            .attr("width", w)
            .attr("height", h)
var g = vis.append("svg:g")
       .attr("transform", "translate(0,
200)");
```

Liniendiagramm in D3

- Schritt 3: Daten für Linie laden

```
var line = d3.svg.line()
.x(function(d,i) { return x(i); })
.y(function(d) { return -1 * y(d);
})
```

- The i parameter is the index of the current item, and the d is the current item itself. Note the usage of the scale functions (x() and y() to convert from i and d to x- and y-coordinates (note the -1. This is needed because right now the coordinate have a positive y-axis down, and since we have translated our g down 200 pixels, we need to use the negative values of the y-axis now).

Liniendiagramm in D3

- Schritt 4: Path Element und Achsen hinzufügen

```
g.append("svg:path").attr("d", line(data));

g.append("svg:line")
        .attr("x1", x(0))
        .attr("y1", -1 * y(0))
        .attr("x2", x(w))
        .attr("y2", -1 * y(0))
g.append("svg:line")
        .attr("x1", x(0))
        .attr("y1", -1 * y(0))
        .attr("x2", x(0))
        .attr("y2", -1 * y(d3.max(data)))
```

Liniendiagramm in D3

- Schritt 5: Skalenbeschriftung hinzufügen

```
g.selectAll(".xLabel")
        .data(x.ticks(5))
        .enter().append("svg:text")
        .attr("class", "xLabel")
        .text(String)
        .attr("x", function(d) { return x(d) })
        .attr("y", 0)
        .attr("text-anchor", "middle")
g.selectAll(".yLabel")
        .data(y.ticks(4))
        .enter().append("svg:text")
        .attr("class", "yLabel")
        .text(String)
        .attr("x", 0)
        .attr("y", function(d) { return -1 * y(d) })
        .attr("text-anchor", "right")
        .attr("dy", 4)
```

Liniendiagramm in D3

http://janwillemtulp.com/d3linechart/

- Schritt 6: Strichlein hinzufügen

```
g.selectAll(".xTicks")
        .data(x.ticks(5))
        .enter().append("svg:line")
        .attr("class", "xTicks")
        .attr("x1", function(d) { return x(d); })
        .attr("y1", -1 * y(0))
        .attr("x2", function(d) { return x(d); })
        .attr("y2", -1 * y(-0.3))
g.selectAll(".yTicks")
        .data(y.ticks(4))
        .enter().append("svg:line")
        .attr("class", "yTicks")
        .attr("y1", function(d) { return -1 * y(d); })
        .attr("x1", x(-0.3))
        .attr("y2", function(d) { return -1 * y(d); })
        .attr("x2", x(0))
```

Übersicht

Eine kurze Einführung in Infovisualisierung

Was ist D3?

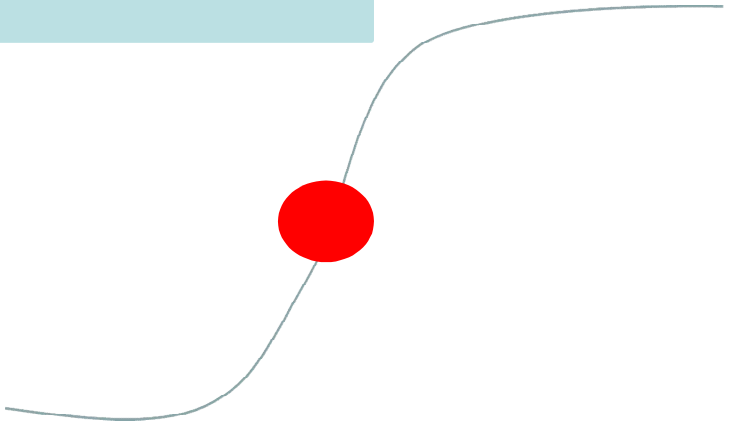
Erste Schritte in D3

Beispiel 1: Liniendiagramm

Beispiel 2: Baum

Beispiel 3: Geokomponenten

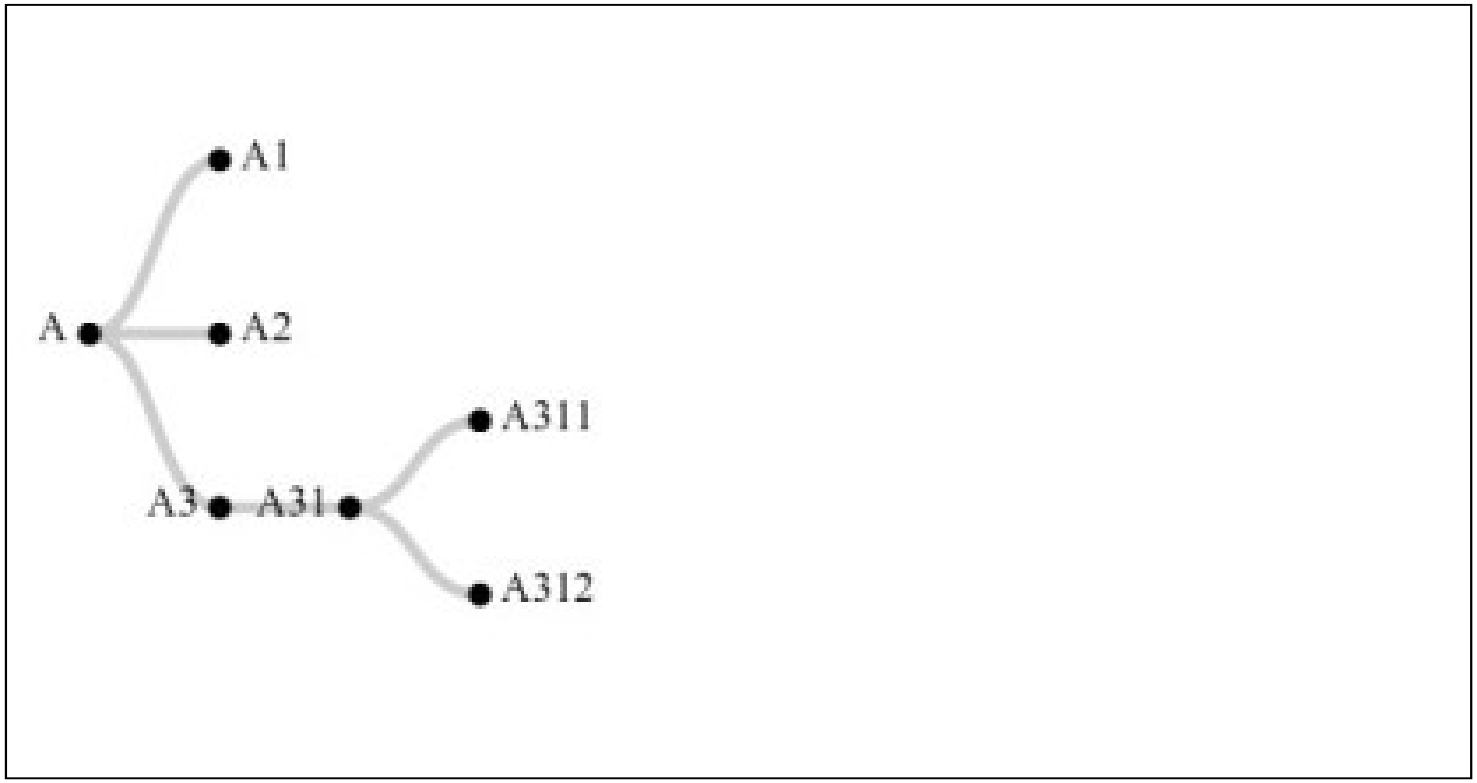
Ausblick: Wie geht es weiter?



Layouts in D3

- Layouts
  - Bundle
  - Chord
  - Cluster
  - Force
  - Hierarchy
  - Histogram
  - Pack
  - Partition
  - Pie
  - Stack
  - Tree
  - Treemap
- wichtig: D3 Layouts zeichnen nicht den Graphen für Dich. Sie sorgen nur für das Laden der Daten.

Beispiel 2: Erstellung eines Baums



Beispiel 2.1: Layout und Styple

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ein einfacher Baum</title>
    <script type="text/javascript" src="lib/d3.js"></script>
    <script type="text/javascript"
src="lib/d3.layout.js"></script>

     <style>
        .link {
            fill: none;
            stroke: #ccc;
            stroke-width: 4.5px;
         }
      </style>
</head>
```

Beispiel 2.2: Baum : Die Daten als JSON-Objekt

```
<body>

    <div id="viz"></div>

    <script type="text/javascript">

      //JSON object with the data
      var treeData = {"name" : "A", "info" : "tst",
"children" : [
            {"name" : "A1" },
            {"name" : "A2" },
            {"name" : "A3", "children": [
                  {"name" : "A31", "children" :[
            {"name" : "A311" },
            {"name" : "A312" }
    ]}] }
      ]};
```

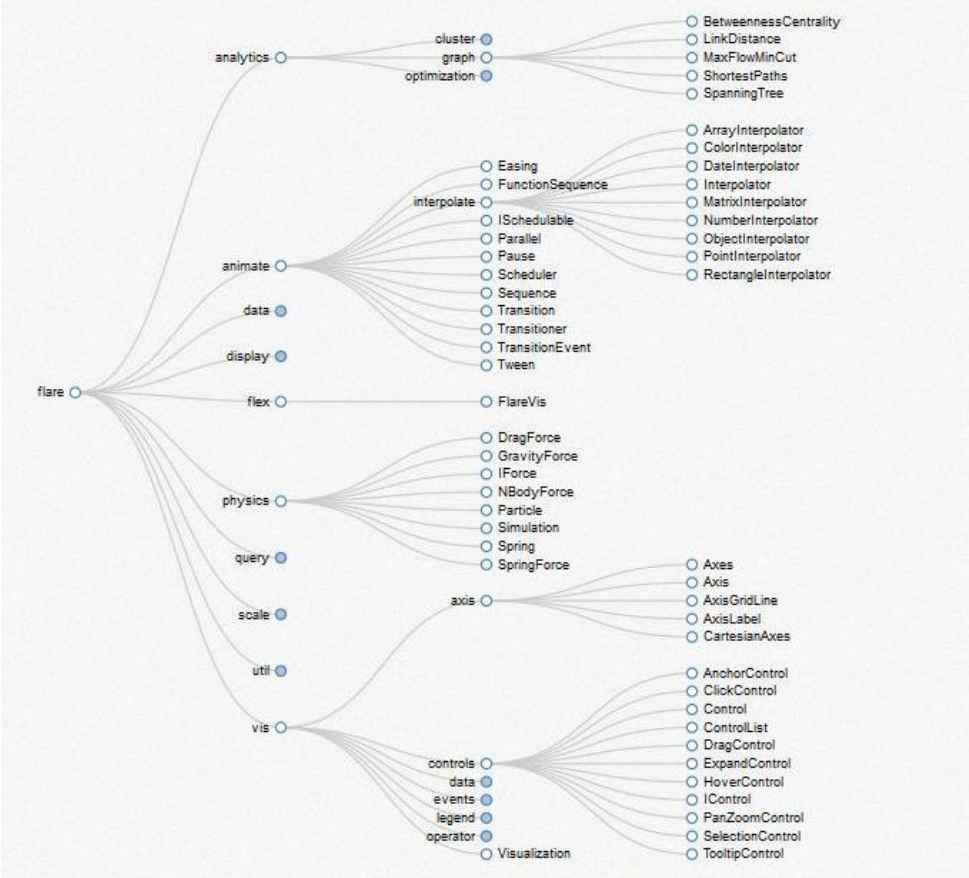
Beispiel 2c: Baum: Canvas, Layout, Daten

```
// Create a svg canvas
     var vis = d3.select("#viz").append("svg:svg")
         .attr("width", 400)
         .attr("height", 300)
         .append("svg:g")
         .attr("transform", "translate(40, 0)"); // shift everything to the right

     // Create a tree "canvas"
     var tree = d3.layout.tree()
         .size([300,150]);

     var diagonal = d3.svg.diagonal()
     // change x and y (for the left to right tree)
         .projection(function(d) { return [d.y, d.x]; });

     // Preparing the data for the tree layout, convert data into an array of nodes
     var nodes = tree.nodes(treeData);
     // Create an array with all the links
     var links = tree.links(nodes);
```

Beispiel 2d: Baum: Knoten und Text

```
var link = vis.selectAll("pathlink")
      .data(links)
      .enter().append("svg:path")
      .attr("class", "link")
      .attr("d", diagonal)

      var node = vis.selectAll("g.node")
      .data(nodes)
      .enter().append("svg:g")
      .attr("transform", function(d) { return "translate(" + d.y + "," + d.x + ")"; })
      // Add the dot at every node
      node.append("svg:circle")
      .attr("r", 3.5);

      // place the name atribute left or right depending if children
      node.append("svg:text")
      .attr("dx", function(d) { return d.children ? -8 : 8; })
      .attr("dy", 3)
      .attr("text-anchor", function(d) { return d.children ? "end" : "start"; })
      .text(function(d) { return d.name; })

    </script>
  </body>
</html>
```

Beispiel 2: Erweiterung ausklappbarer Baum



- Beispiel http://localhost:8888/17AusklappBaum.html

Übersicht

Eine kurze Einführung in Infovisualisierung

Was ist D3?

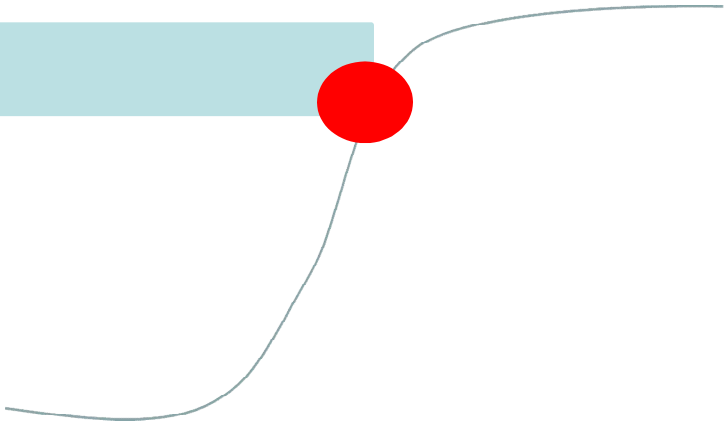
Erste Schritte in D3

Beispiel 1: Liniendiagramm

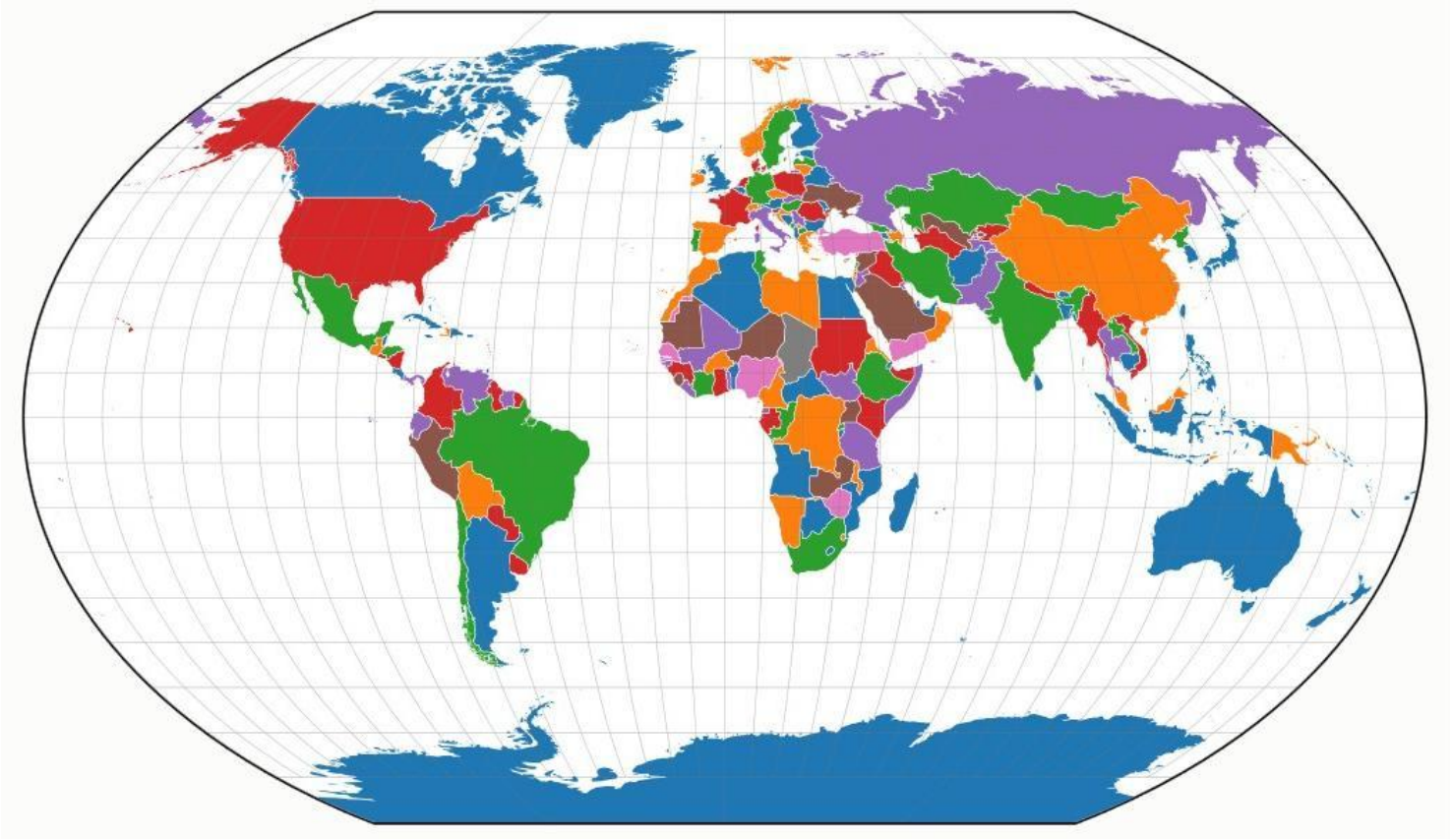
Beispiel 2: Baum

Beispiel 3: Geokomponenten

Ausblick: Wie geht es weiter?



Beispiel 3: Arbeiten mit Karten



- Beispiel World Map: (http://bl.ocks.org/mbostock/4180634)

Beispiel 3.1: Weltkarte: Einstellungen

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>
    body { background: #fcfcfa; }
        .stroke { fill: none; stroke: #000; stroke-width: 3px; }
        .fill { fill: #fff; }
        .graticule { fill: none; stroke: #777;
                stroke-width: .5px; stroke-opacity: .5; }
        .land { fill: #222; }
        .boundary { fill: none; stroke: #fff; stroke-width: .5px; }
 </style>
```

Beispiel 3.2: Projektion & Tranformationen

```
<body>
<script src="http://d3js.org/d3.v3.min.js"></script>
<script src="http://d3js.org/d3.geo.projection.v0.min.js"></script>
<script src="http://d3js.org/topojson.v1.min.js"></script>
<script>

var width = 960,
    height = 580;

var color = d3.scale.category10();

var projection = d3.geo.kavrayskiy7()
    .scale(170)
    .translate([width / 2, height / 2])
    .precision(.1);

var path = d3.geo.path()
    .projection(projection);

var graticule = d3.geo.graticule();
```

Beispiel 3.3: Weltkarte: Linienzüge + Links

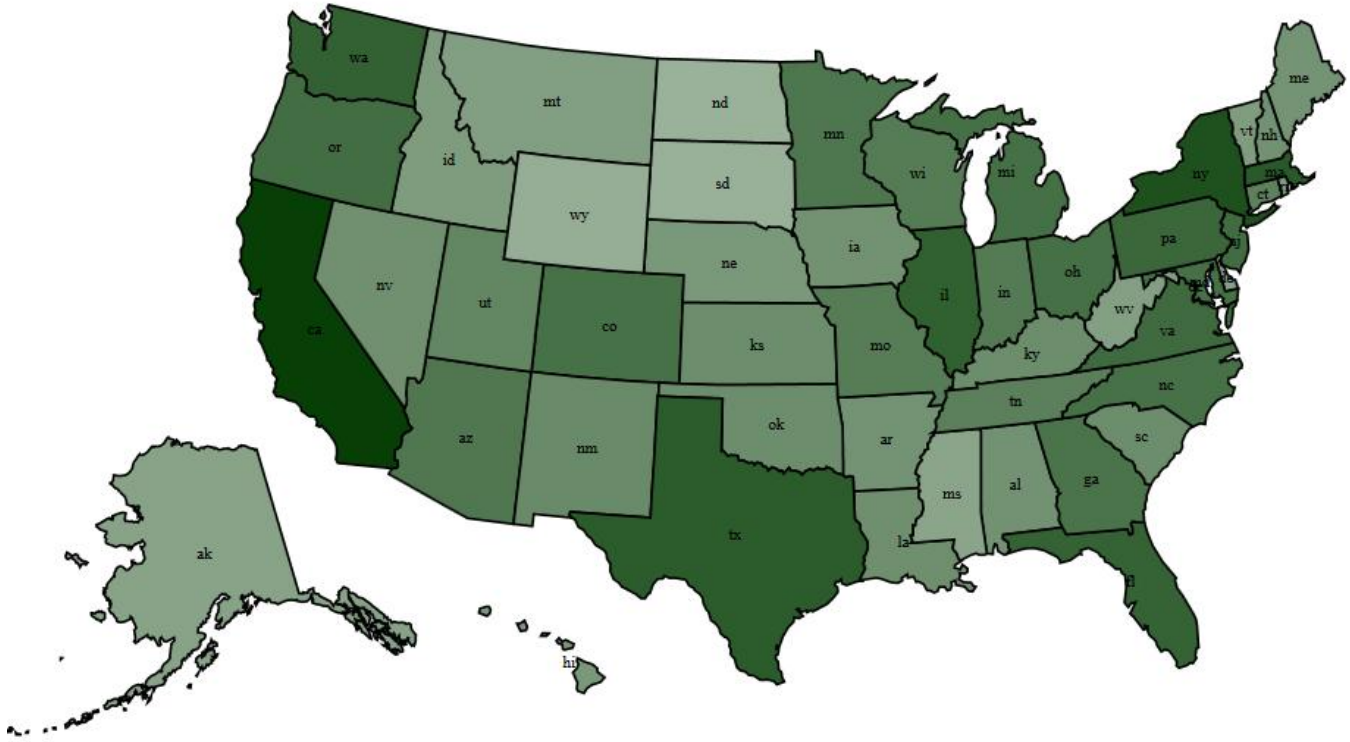
```
var svg = d3.select("body").append("svg")
    .attr("width", width)
    .attr("height", height);

svg.append("defs").append("path")
    .datum({type: "Sphere"})
    .attr("id", "sphere")
    .attr("d", path);

svg.append("use")
    .attr("class", "stroke")
    .attr("xlink:href", "#sphere");

svg.append("use")
    .attr("class", "fill")
    .attr("xlink:href", "#sphere");

svg.append("path")
    .datum(graticule)
    .attr("class", "graticule")
    .attr("d", path);
```

Beispiel 3.4: Weltkarte: Einstellungen

http://bl.ocks.org/mbostock/4180634

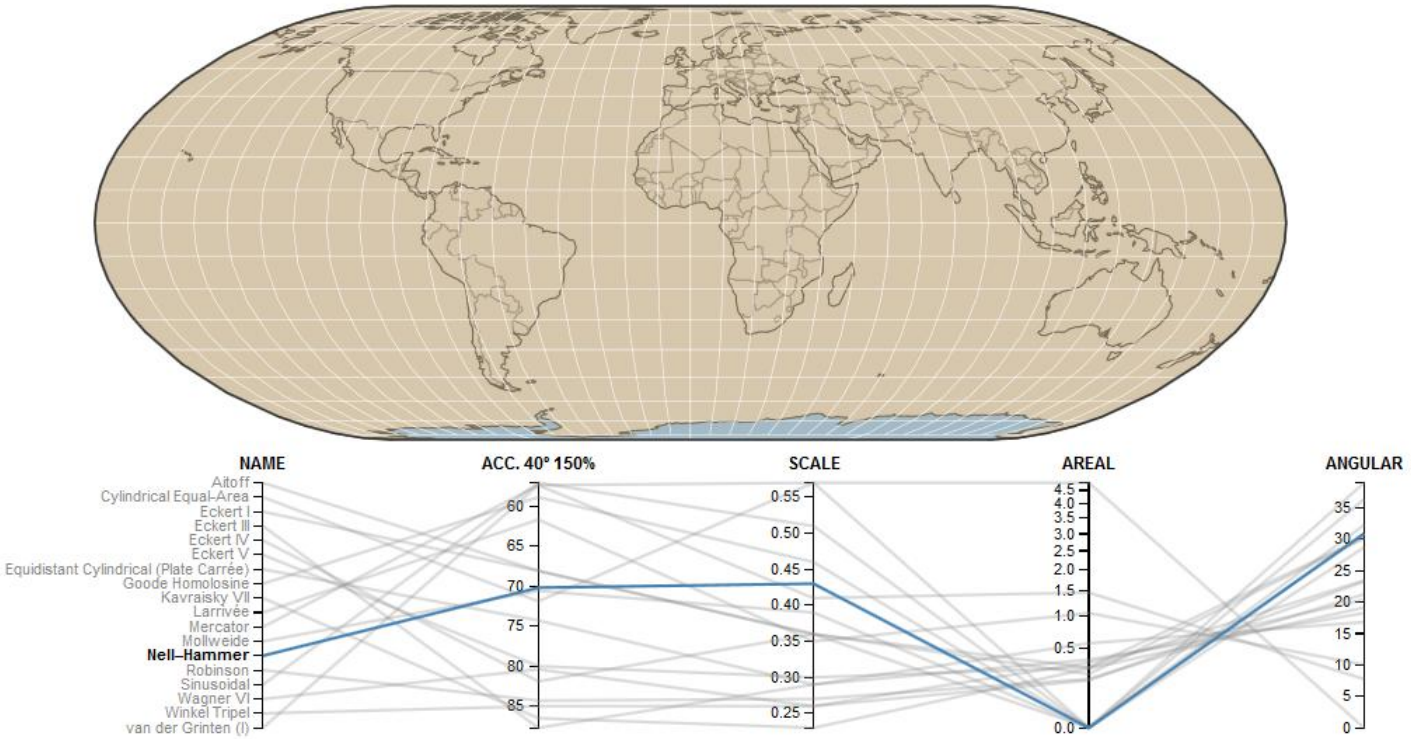
```
d3.json("/mbostock/raw/4090846/world-50m.json", function(error, world) {
  var countries = topojson.feature(world, world.objects.countries).features,
      neighbors = topojson.neighbors(world.objects.countries.geometries);

  svg.selectAll(".country")
      .data(countries)
    .enter().insert("path", ".graticule")
      .attr("class", "country")
      .attr("d", path)
      .style("fill", function(d, i) { return color(d.color = d3.max(neighbors[i],
          function(n) { return countries[n].color; }) + 1 | 0); });

  svg.insert("path", ".graticule")
      .datum(topojson.mesh(world, world.objects.countries,
          function(a, b) { return a !== b; }))
      .attr("class", "boundary")
      .attr("d", path);
});

d3.select(self.frameElement).style("height", height + "px");

</script>
```

Beispiel 3: Erweiterung Karten: Zooming



- http://bl.ocks.org/thedod/raw/4548858/

Beispiel 3: Erweiterung Karten: Projektionsarten



- http://bl.ocks.org/syntagmatic/raw/3711245/

Beispiel 3: Erweiterung Karten: Worldtour

- http://bl.ocks.org/mbostock/raw/4183330/

Übersicht

Eine kurze Einführung in Infovisualisierung

Was ist D3?

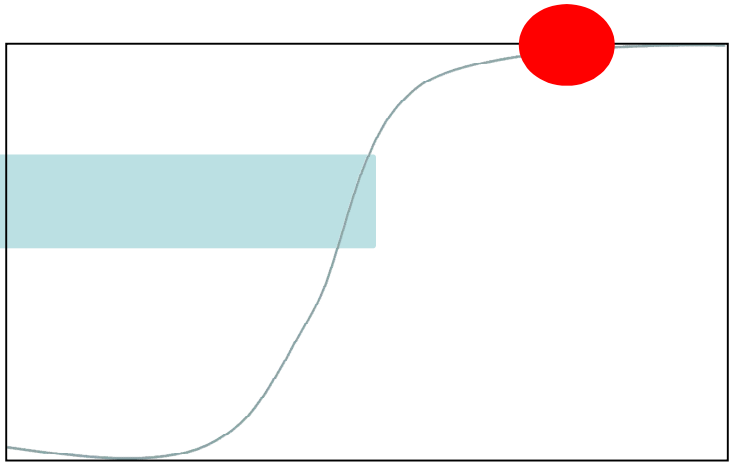
Erste Schritte in D3

Beispiel 1: Liniendiagramm

Beispiel 2: Baum

Beispiel 3: Geokomponenten

Ausblick: Wie geht es weiter?



Alternativen zu D3

Easy Charts
- Flot
- gRaphael
- Highcharts JS
- JavaScript Info Vis Toolkit
- jgPlot
- Peity
- Timeline.js

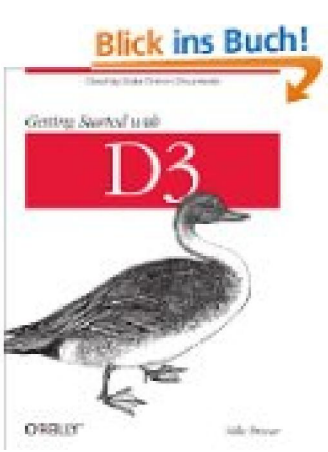
Graph Visualisierung
- Arbor.js
- Sigma.js

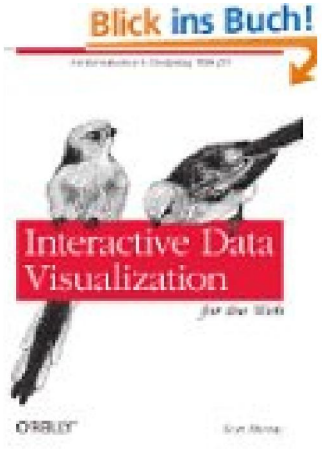
Geomapping
- Kartograph
- Leaflet
- Modest Maps
- Polymaps

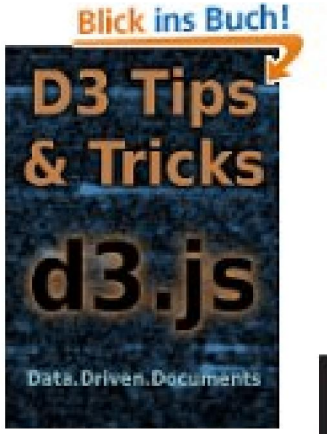
Almost from Scratch (D3 vglbar)
- Processing.js
- Paper.js
- Raphael

3D-Tools
- PhiloGL
- Three.js

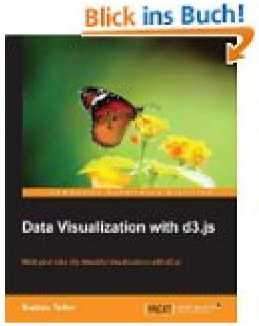
Literatur zu D3

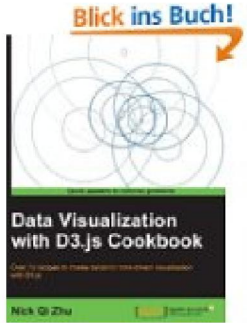


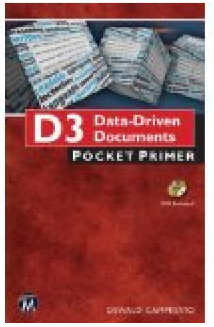


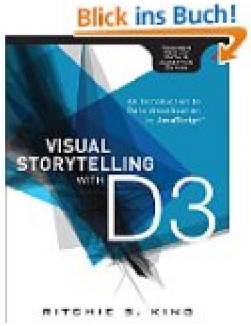


- Getting Started with D3
- Interactive Data Visualization for the Web
- D3 Tips & Tricks
- Data Visualization with D3.js Cookbook
- Data Visualization with D3.js
- D3_ Pocket Primer
- Visual Storytelling with D3









Links

- http://d3js.org/
- https://github.com/mbostock/d3/wiki/Gallery
- http://wheresmydiskspace.com/
- http://tulpinteractive.com/
- http://www.janwillemtulp.com/category/d3/
- http://bl.ocks.org/jasondavies/4091835
- http://bl.ocks.org/ilyabo/2209220
- http://bl.ocks.org/mbostock/4180634
- http://world.time.com/2013/12/05/nelson-mandelas-extraordinary-life-an-interactive-timeline/

Gibt es Fragen?


Herzlichen Dank und auf ein baldiges Wiedersehen