



Open Data: Datenmanagement und Visualisierung Daten darstellen und positionieren, interaktive Diagramme

Termin 10, 1. Mai 2014

Dr. Matthias Stürmer und Prof. Dr. Thomas Myrach
Universität Bern, Institut für Wirtschaftsinformatik
Abteilung Informationsmanagement
Forschungsstelle Digitale Nachhaltigkeit

10: Daten darstellen und positionieren, interaktive Diagramme

Donnerstag, 1. Mai 2014

- > Buch „Interactive Data Visualization for the Web“
Kapitel: 6. Drawing with Data
- > Variablen und Funktionen, Daten verketteten und anbinden, Balken und Kreise zeichnen, Diagramme aufbauen

Gastreferat im zweiten Teil der Vorlesung:

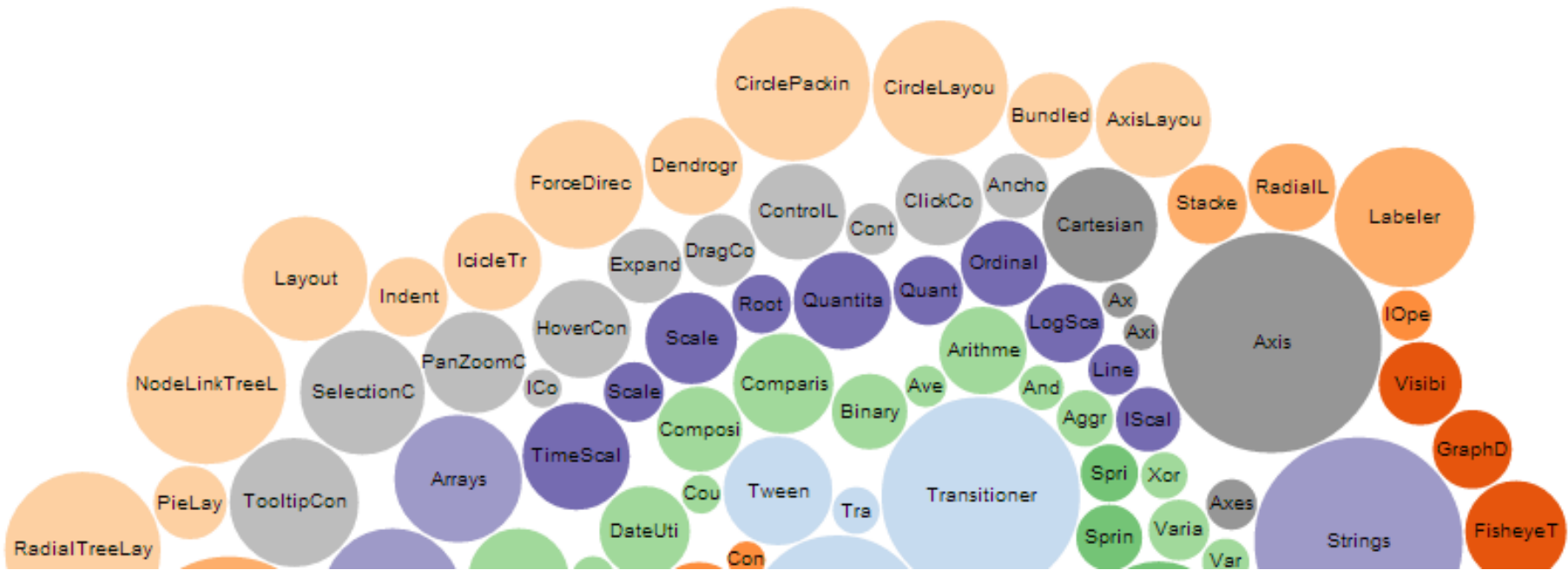
- > **Thomas Preusse, Open Data App Entwickler**
- > Entwicklung der Open Budget App mit D3.js:
 - Datenaufbereitung, Datenimport
 - Bedienelemente der grafischen Benutzeroberfläche
 - Technischer Aufbau der Open Budget App
 - Learnings und Verbesserungsmöglichkeiten





Agenda

1. Infos zur Open Data App und Prüfung
2. Daten als Kreise darstellen und positionieren



Infos zur Open Data App

- > Kurzpräsentation in der Vorlesung vom **Donnerstag, 22. Mai 2014** ist **Voraussetzung für Anrechnung und Note. Mindestens eine Person muss anwesend sein.** App zählt 50% der Note, Prüfung zählt 50% der Note.
- > **Minimalanforderungen an die App (für Note "genügend"):**
 1. Neue Daten aufbereiten und visualisieren (Data Coach, Datenportal, eigene Daten)
 2. Mindestens eine kreative Visualisierung, nicht bloss Balken oder Kuchendiagramm
 3. Mindestens eine interaktive Funktion (Hover, Mouse Click, Scroll Wheel, Button etc.)
 4. Aufwand von rund 40 bis 50 Stunden pro Person muss erkennbar sein
- > **Bewertungskriterien:**
 1. **Komplexität:** Wie anspruchsvoll sind die visualisierten Daten und der behandelte Themenkomplex als ganzes?
 2. **Kreativität:** Wie neuartig und attraktiv sind die Visualisierung der Daten und technische Implementierung der Open Data App?
 3. **Umsetzung:** Wie benutzerfreundlich, verständlich und gut dokumentiert ist die Open Data App?
 4. **Impact:** Wie hoch ist die Bedeutung und die Aussagekraft der Datenvisualisierung und der Open Data App als gesamtes?

Vorgehen Präsentationen Open Data App

- > **Folien** zu den 5 Punkten:
 1. Team (wer, welchen Hintergrund)
 2. Ziel, Motivation (Aufgabestellung)
 3. Resultat (mit öffentlich zugänglichem Link)
 4. Datenquellen (Ursprungsformat, Endformat)
 5. Vorgehen (notwendige Schritte)

- > Kurze **Live App-Demo**

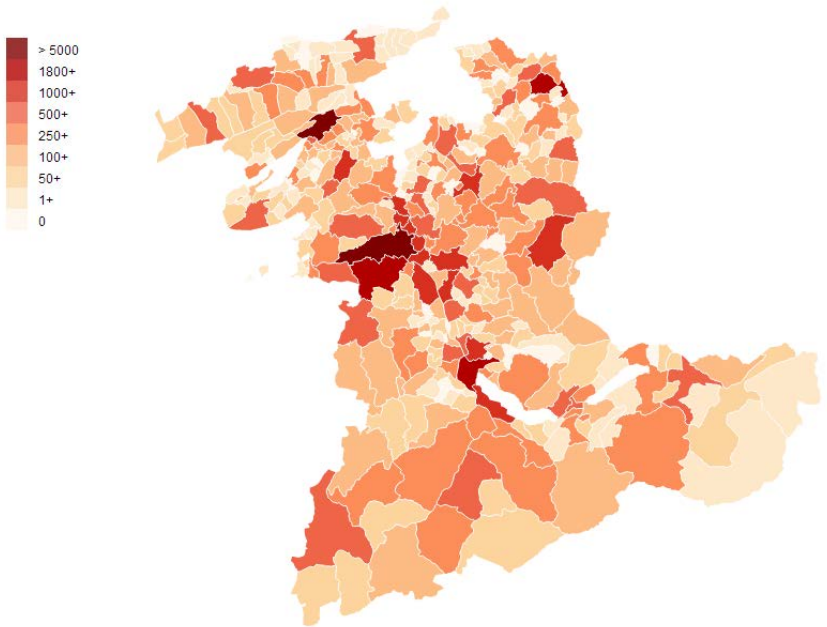
- > Zeitvorgabe: maximal 4 Minuten! (je nach Anzahl Apps mehr)

- > Anmeldung für Kurzpräsentation:
Mail an Rahel Winkelmann rahel.winkelmann@iwi.unibe.ch
bis 20. Mai 2014 mit den Folien als PowerPoint oder PDF

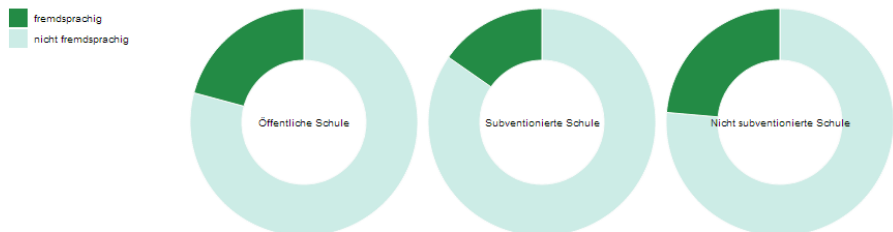
Open Data App

> Beispiel einer hervorragenden Studierenden-App:
<http://st4ple.github.io/schulstat/>

Anzahl Schüler pro Gemeinde

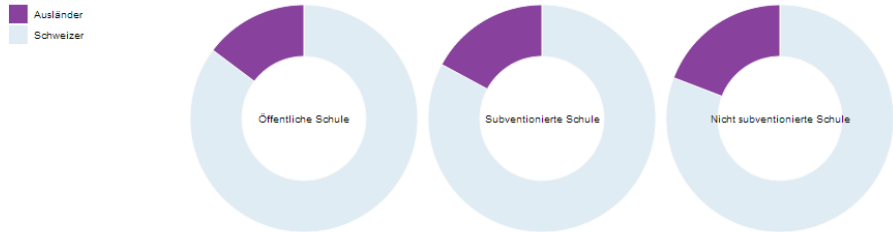


Relativer Anteil fremdsprachiger Schüler pro Kostenträger



An subventionierten Privatschulen sind anteilmässig etwas weniger fremdsprachige Schüler als an öffentlichen Schulen anzutreffen, an nicht subventionierten Privatschulen sind hingegen etwas mehr. Es ist nicht wirklich ein Trend zu erkennen.

Relativer Anteil ausländischer Schüler pro Kostenträger



Prüfung

- > **1. Prüfungstermin FS2014**
Datum: Donnerstag, 05.06.2014
Zeit: 14:45 - 16:15
Ort: 201 HG

- > **2. Prüfungstermin FS2014:**
Datum: Donnerstag, 04.09.2014
Zeit: 14:15 - 15:45
Ort: 201 HG

Beispielaufgaben der Prüfung

Beispielaufgaben:

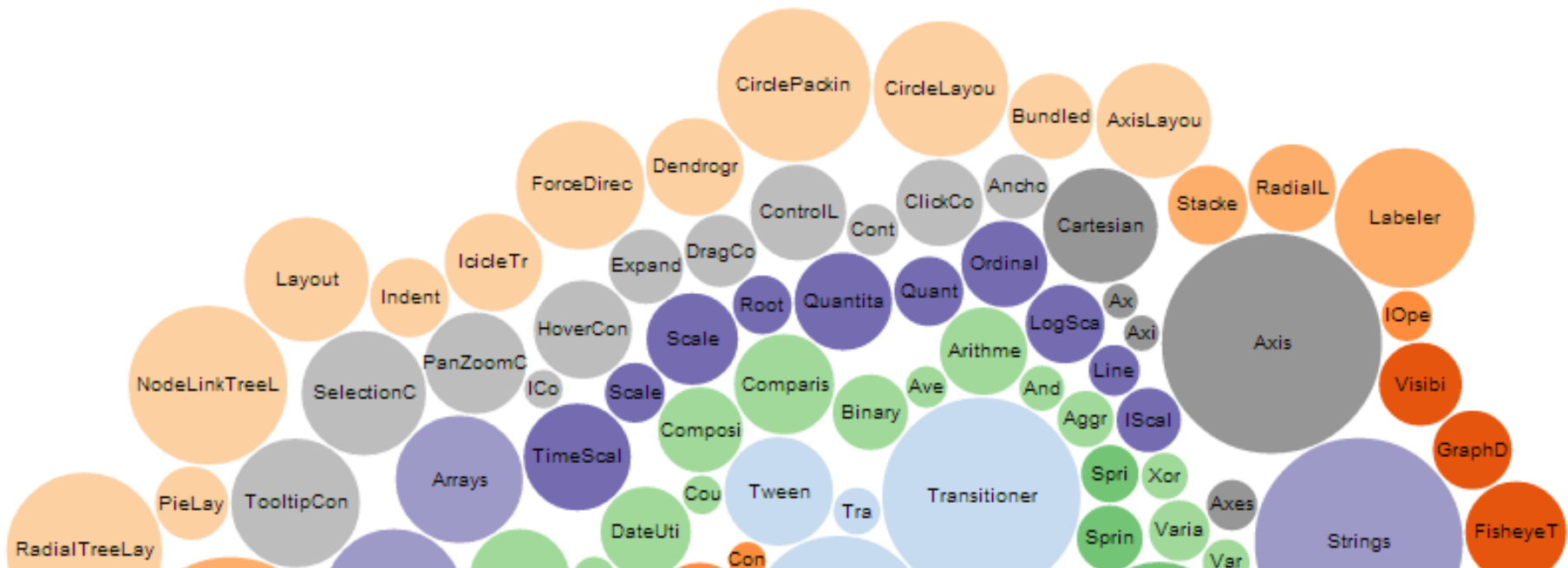
- > **Inhalte der Vorlesung:**
"Was misst der Open Budget Survey?"
Lösung: siehe Folien zu Open Finance Data

- > **Inhalte der Gastreferate:**
"Zähle 4 kritische Punkte auf, die Haftungsfragen bei der Informationsqualität hervorrufen können."
Lösung: siehe Folien Gastreferat Dr. Christian Laux

- > **Grundwissen Programmierung:**
"Was für eine Variable ergibt der Befehl
`var animal = {color: "brown", skin: "fur", carnivore:true} ?`"
Lösung: siehe Folien zu Einführung D3 App Entwicklung

Agenda

1. Infos zur Open Data App und Prüfung
2. **Daten als Kreise darstellen und positionieren**



D3.js Circles Example

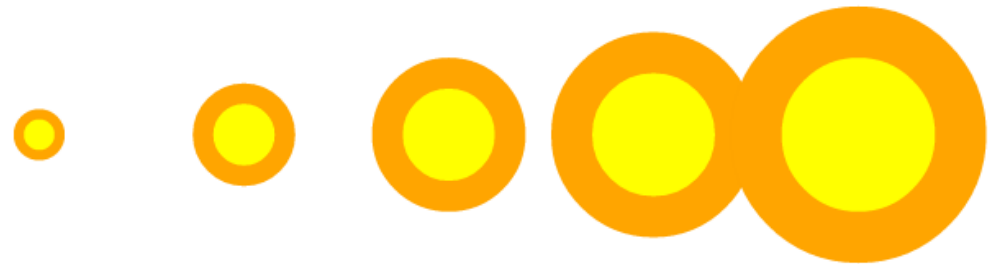
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Chapter 6 - Circles Example</title>
    <script type="text/javascript" src="/js/vendor/d3.min.js"></script>
  </head>
  <body>
    <script type="text/javascript">
//Width and height
var w = 500;
var h = 100;

//Data
var dataset = [ 5, 10, 15, 20, 25 ];

//Create SVG element
var svg = d3.select("body")
  .append("svg")
  .attr("width", w)
  .attr("height", h);

var circles = svg.selectAll("circle")
  .data(dataset)
  .enter()
  .append("circle");

circles.attr("cx", function(d, i) {
  return (i * 50) + 25;
})
  .attr("cy", h/2)
  .attr("r", function(d) {
    return d;
  })
  .attr("fill", "yellow")
  .attr("stroke", "orange")
  .attr("stroke-width", function(d) {
    return d/2;
  });
</script>
</body>
</html>
```



SVG Circle

9.3 The 'circle' element

The '[circle](#)' element defines a circle based on a center point and a radius.

Categories:

[Basic shape element](#), [graphics element](#), [shape element](#)

'circle'

Content model:

Any number of the following elements, in any order:

[animation elements](#) [show »](#)

[descriptive elements](#) [show »](#)

Attributes:

[conditional processing attributes](#) [show »](#)

[core attributes](#) [show »](#)

[graphical event attributes](#) [show »](#)

[presentation attributes](#) [show »](#)

'class'

SVG Circle

w3schools.com

HOME HTML CSS JAVASCRIPT SQL PHP JQUERY XML ASP.NET MORE... REFERENCES | EXAMPLES

SVG Basic
 SVG HOME
 SVG Intro
 SVG in HTML5

SVG Shapes
 SVG Rectangle
SVG Circle
 SVG Ellipse
 SVG Line
 SVG Polygon
 SVG Polyline
 SVG Path
 SVG Text
 SVG Stroking

SVG Filters
 SVG Filters Intro
 SVG Blur Effects
 SVG Drop Shadows

SVG Gradients
 SVG Linear
 SVG Radial

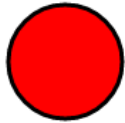
SVG Examples

SVG <circle>

[« Previous](#) [Next Chapter »](#)

SVG Circle - <circle>

The <circle> element is used to create a circle:



Here is the SVG code:

```
Example
<svg height="100" width="100">
  <circle cx="50" cy="50" r="40" stroke="black" stroke-width="3" fill="red" />
</svg>
```

[Try it yourself »](#)

Code explanation:

- The cx and cy attributes define the x and y coordinates of the center of the circle. If cx and cy are omitted, the

Link: http://www.w3schools.com/svg/svg_circle.asp

D3.js Circles Example

Grösse der SVG Fläche festlegen, Daten festlegen, SVG Fläche generieren:

```
//Width and height
var w = 500;
var h = 100;

//Data
var dataset = [ 5, 10, 15, 20, 25 ];

//Create SVG element
var svg = d3.select("body")
    .append("svg")
    .attr("width", w)
    .attr("height", h);
```

D3.js Circles Example

Kreise der SVG Fläche hinzufügen:

```
var circles = svg.selectAll("circle")
    .data(dataset)
    .enter()
    .append("circle");
```

Die Kreise werden generiert, aber noch ohne Attribute:

```
▼ <svg width="500" height="100">
  <circle></circle>
  <circle></circle>
  <circle></circle>
  <circle></circle>
  <circle></circle>
  </svg>
```

D3.js Circles Example

Kreise immer 50 Pixel nach rechts verschoben mit `cx` positionieren:

```
circles.attr("cx", function(d, i) {  
    return (i * 50) + 25;  
})
```

`d` sind die Daten (5, 10, 15, 20, 25)

`i` ist der Index des Datenpunkts (1, 2, 3, 4, 5)

```
▼ <svg width="500" height="100">  
  <circle cx="25"></circle>  
  <circle cx="75"></circle>  
  <circle cx="125"></circle>  
  <circle cx="175"></circle>  
  <circle cx="225"></circle>  
</svg>
```

D3.js Circles Example

Kreise vertikal in der Mitte der SVG-Fläche $h/2$ positionieren und den Radius so gross wie den Datenwert d festlegen:

```
circles.attr("cx", function(d, i) {  
    return (i * 50) + 25;  
})  
    .attr("cy", h/2)  
    .attr("r", function(d) {  
        return d;  
    });
```


D3.js Circles Example

The screenshot shows a web browser window with the address bar containing '127.0.0.1:64995/chapter6.html'. The main content area displays five black circles of increasing size, centered horizontally. Below the browser window is a developer console showing the following HTML code:

```

<!DOCTYPE html>
<html lang="de">
  <head>...</head>
  <body>
    <script type="text/javascript">...</script>
    <svg width="500" height="100">
      <circle cx="25" cy="50" r="5"></circle>
      <circle cx="75" cy="50" r="10"></circle>
      <circle cx="125" cy="50" r="15"></circle>
      <circle cx="175" cy="50" r="20"></circle>
      <circle cx="225" cy="50" r="25"></circle>
    </svg>
  </body>
</html>

```

D3.js Circles Example

Füllfarbe gelb, Randfarbe orange und Randdicke proportional zum Kreis:

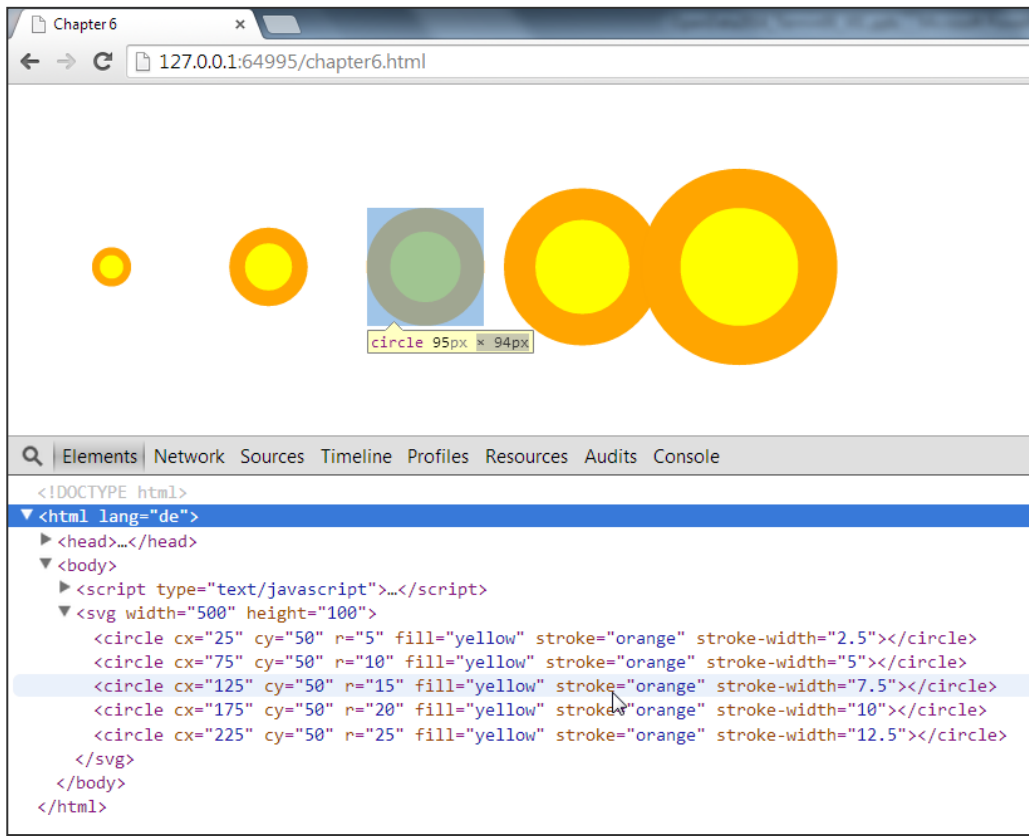
```
circles.attr("cx", function(d, i) {  
    return (i * 50) + 25;  
})  
.attr("cy", h/2)  
.attr("r", function(d) {  
    return d;  
})  
.attr("fill", "yellow")  
.attr("stroke", "orange")  
.attr("stroke-width", function(d) {  
    return d/2;  
});
```

D3.js Circles Example

```

1 | <!DOCTYPE html>
2 | <html lang="de">
3 |   <head>
4 |     <meta charset="utf-8">
5 |     <title>Chapter 6 - Circles Example</title>
6 |     <script type="text/javascript" src="/js/vendor/d3.min.js"></script>
7 |   </head>
8 |   <body>
9 |     <script type="text/javascript">
10 |
11 |       //Width and height
12 |       var w = 500;
13 |       var h = 100;
14 |
15 |       //Data
16 |       var dataset = [ 5, 10, 15, 20, 25 ];
17 |
18 |       //Create SVG element
19 |       var svg = d3.select("body")
20 |         .append("svg")
21 |         .attr("width", w)
22 |         .attr("height", h);
23 |
24 |       var circles = svg.selectAll("circle")
25 |         .data(dataset)
26 |         .enter()
27 |         .append("circle");
28 |
29 |       circles.attr("cx", function(d, i) {
30 |         return (i * 50) + 25;
31 |       })
32 |         .attr("cy", h/2)
33 |         .attr("r", function(d) {
34 |           return d;
35 |         })
36 |         .attr("fill", "yellow")
37 |         .attr("stroke", "orange")
38 |         .attr("stroke-width", function(d) {
39 |           return d/2;
40 |         });
41 |
42 |     </script>
43 |   </body>
44 | </html>

```



D3.js Circles Example

3 Aufgaben zum Üben:

1. Kreise horizontal proportional zu ihrer Grösse verteilen
2. Zufällige Grösse der Kreise erstellen
3. Datenwert pro Kreis anzeigen

Another D3.js Circles Example

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Chapter 6 - Another Circles Example</title>
    <script type="text/javascript" src="/js/vendor/d3.min.js"></script>
  </head>
  <body>
    <script type="text/javascript">

      var w = 500;
      var h = 100;
      var dataset = [
        [5, 20], [480, 90], [250, 50], [100, 33], [330, 95],
        [410, 12], [475, 44], [25, 67], [85, 21], [220, 88]
      ];

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

      svg.selectAll("circle")
        .data(dataset)
        .enter()
        .append("circle")
        .attr("cx", function(d) {
          return d[0];
        })
        .attr("cy", function(d) {
          return d[1];
        })
        .attr("r", function(d) {
          return Math.sqrt(h - d[1]);
        });

      svg.selectAll("text")
        .data(dataset)
        .enter()
        .append("text")
        .text(function(d) {
          return d[0] + ", " + d[1];
        })
        .attr("x", function(d) {
          return d[0];
        })
        .attr("y", function(d) {
          return d[1];
        })
        .attr("font-family", "sans-serif")
        .attr("font-size", "11px")
        .attr("fill", "red");

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

```

The screenshot shows a web browser displaying a visualization of 10 data points. Each point is represented by a black circle with a red text label indicating its coordinates (x, y). The labels are: 5,20; 480,90; 250,50; 100,33; 330,95; 410,12; 475,44; 25,67; 85,21; and 220,88. The browser's developer tools are open, showing the HTML source code. The code includes a D3.js script that uses the dataset to create circles and text elements. The SVG elements are defined with attributes for cx, cy, r, x, y, font-family, font-size, and fill.