Responsive Web Design:
Write once, run everywhere

16. April 2016

Oscar Meier

Universität Bern, Institut für Wirtschaftsinformatik
Abteilung Informationsmanagement
Forschungsstelle Digitale Nachhaltigkeit

UNIVERSITÄT BERN

Programming Coaching

Oscar Meier
Hilfsassistent IWI
Forschungsstelle Digitale Nachhaltigkeit

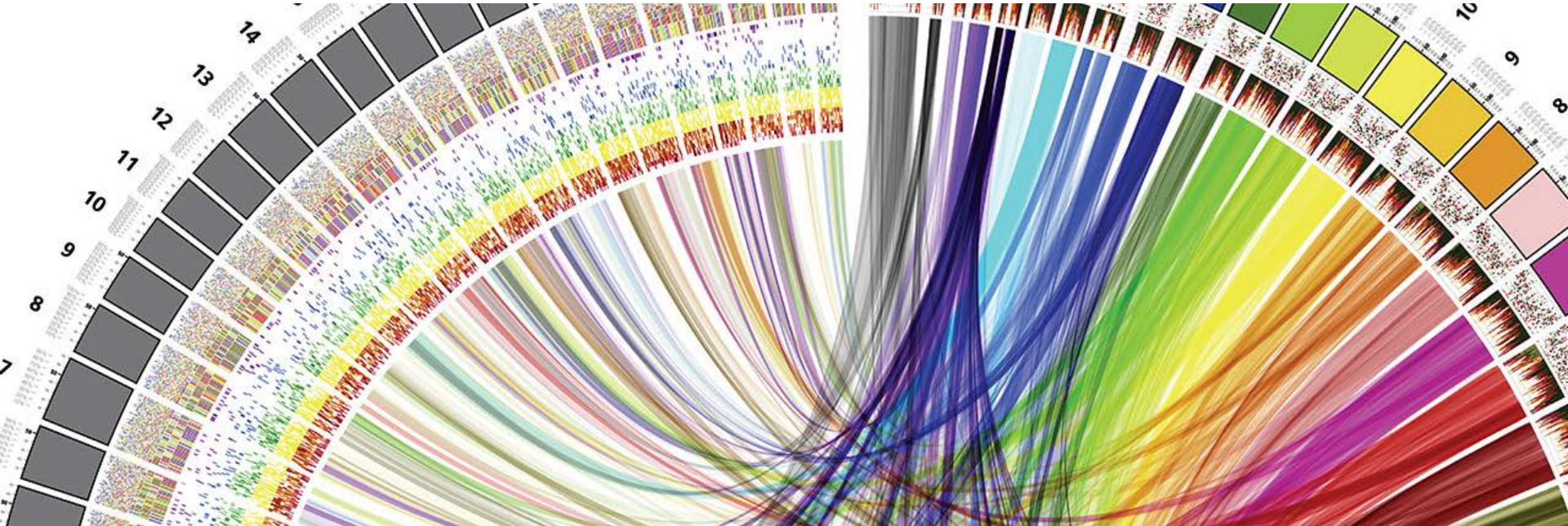
Janik Endtner
Hilfsassistent IWI
Forschungsstelle Digitale Nachhaltigkeit

Kontaktadresse für Fragen betreffend der Applikation:
opendata@iwi.unibe.ch

UNIVERSITÄT BERN

Agenda

1. **Einführung Responsive Web Design**
2. Umsetzung
3. Bootstrap
4. D3.js und Responsive Web Design
5. Fragen und Antworten



Was bedeutet Responsive Web Design?

- Write once, run everywhere – in jedem Browser

Designed by Freepik

Responsive Web Design

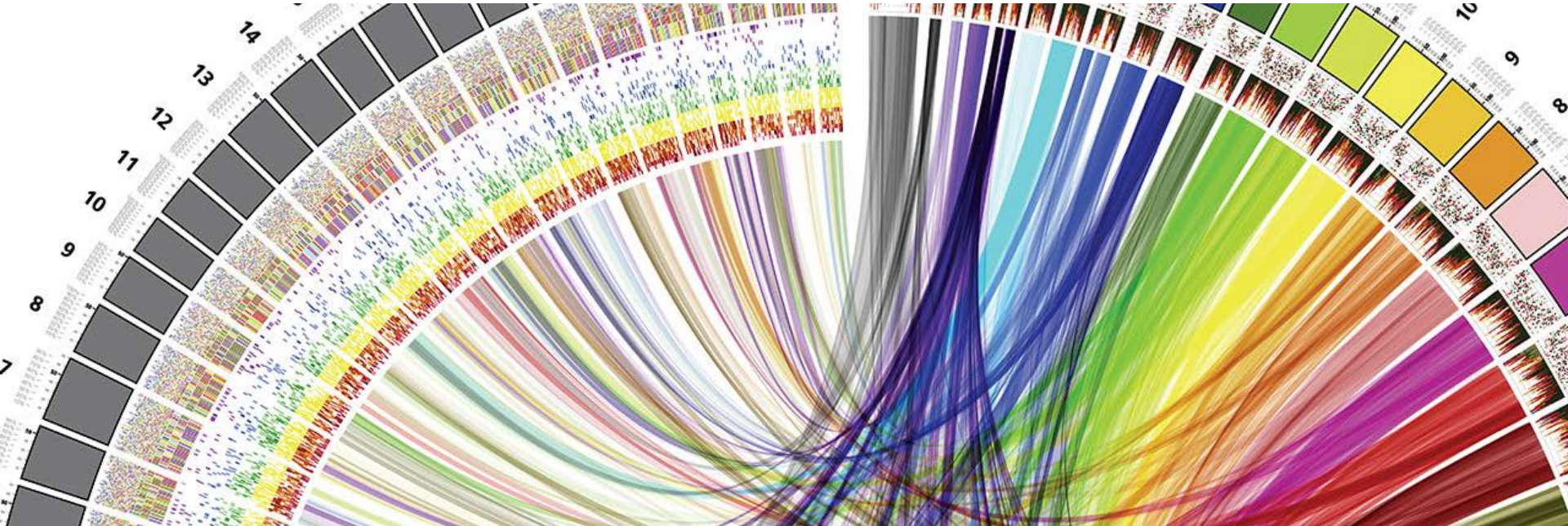
- Benutzerfreundliche(s) Bedienung/Layout auf mehreren Geräten mit verschiedenen Bildschirmgrössen
- Weniger Aufwand/Code
- Effiziente Alternative zu native Android/Apple Applikationen
- Hauptsächlich CSS-basierte Programmierung => Braucht kaum Javascript Kenntnisse

Responsive Web Design

- Beispiel mit schlechtem Responsive Web Design
  - http://arngren.net/

Agenda

1. Einführung Responsive Web Design
2. **Umsetzung**
3. Bootstrap
4. D3.js und Responsive Web Design
5. Fragen und Antworten



Umsetzung – Erste Ansätze mit Pixelwerten

Ansatz Pixel

Container .pixel-container

```
.pixel-container {
    width: 1060px;
    padding: 10px;
    box-sizing: border-box;
    text-align: center;
}
```

Div Eins .pixel-div-one

```
.pixel-div-one {
    width: 490px;
    height: 200px;
    box-sizing: border-box;
    float: left;
}
```

Div Zwei .pixel-div-two

```
.pixel-div-two {
    width: 490px;
    height: 200px;
    box-sizing: border-box;
    float: left;
}
```

Umsetzung – Erste Ansätze mit Pixelwerten

Umsetzung – Ohne Responsive Design

- ResponsiveDesignLectureCode\index-responsive-design-pixel.html
- https://bl.ocks.org/mbostock/3885304

- Problem:
  - Auf kleineren Bildschirmen wird der Inhalt **abgeschnitten**

Umsetzung – Erste Ansätze mit Prozentwerten

Ansatz Prozent

Container .percent-container

```
.percent-container {
    width: 100%;
    box-sizing: border-box;
    text-align: center;
    padding: 1%;
}
```

Div Eins .percent-div-one

```
.percent-div-one {
    width: 50%;
    box-sizing: border-box;
    float: left;
}
```

Div Zwei .percent-div-two

```
.percent-div-two {
    width: 50%;
    box-sizing: border-box;
    float: left;
}
```

Umsetzung – Erste Ansätze mit Prozentwerten

Ansatz Prozent

Container .percent-container

```
.percent-container {
    width: 100%;
    box-sizing: border-box;
    text-align: center;
    padding: 1%;
}
```

Div Eins .percent-div-one

```
.percent-div-one {
    width: 50%;
    box-sizing: border-box;
    float: left;
}
```

Div Zwei .percent-div-two

```
.percent-div-two {
    width: 50%;
    box-sizing: border-box;
    float: left;
}
```

Umsetzung – Erste Ansätze mit Prozentwerten

Ansatz Prozent

Container .percent-container

```
1  .percent-container {
2      width: 100%;
3      box-sizing: border-box;
4      text-align: center;
5      padding: 1%;
6  }
```

Div Eins .percent-div-one

```
1  .percent-div-one
2      width: 50%;
3      box-sizing: b
4      float: left;
5  }
```

Div Zwei .percent-div-two

```
1  .percent-div-two
2      width: 50%;
3      box-sizing: b
4      float: left;
5  }
```

Umsetzung – Erste Ansätze mit Prozentwerten

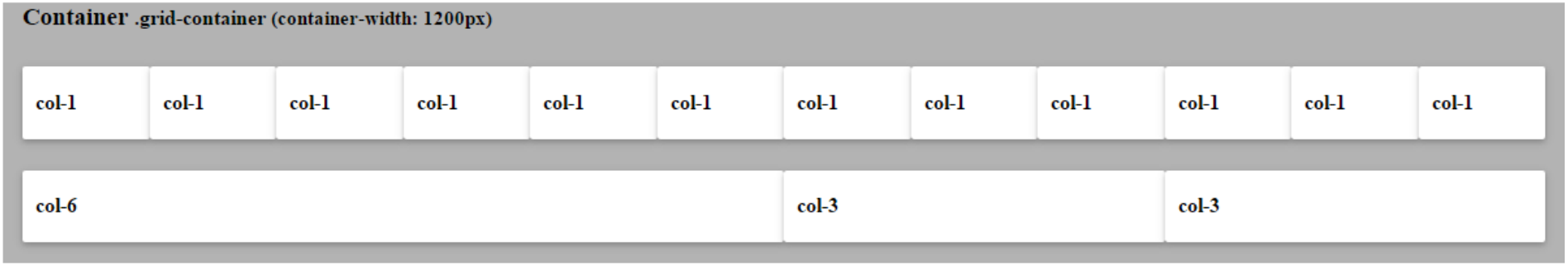
- ResponsiveDesignLectureCode\index-responsive-design-percent.html

- Vorteile im Vergleich zum Pixelansatz:
  - Gesamter HTML-Inhalt unabhängig von der Bildschirmgrösse **immer abgebildet ohne Scrollbar**
- Probleme:
  - Auf grösseren Bildschirmen sind die Inhalte im Gegensatz zu dem Pixelansatz **verzogen**
  - Auf kleineren Bildschirmen sind die Inhalte **nicht mehr übersichtlich**

Umsetzung – Grid Ansatz

- Problem 1:
  - Auf grösseren Bildschirmen sind die Inhalte im Gegensatz zu dem Pixelansatz **verzogen**
- Lösung: Mischung aus Pixel- und Prozentansatz
  - Das Grid besteht aus mehreren Reihen, die jeweils aus **12 Kolonnen** zusammengesetzt sind
  - Die Dimensionen der Kolonnen in Prozent
  - Die Dimensionen des Containers in Pixels und somit eingeschränkt

Ansatz Grid



Umsetzung – Grid Ansatz

- @media Css-Attribut
  - Ermöglicht verschiedene CSS-Definitionen für verschiedene Bildschirmgrössen

```
@media (min-width: 768px) {
    .grid-container {
        width: 750px;
    }
    .container-title:after {
        font-size: 15px;
        content: "(container-width: 750px)";
    }
}
```

```
@media (min-width: 992px) {
    .grid-container {
        width: 970px;
    }
    .container-title:after {
        font-size: 15px;
        content: "(container-width: 992px)";
    }
}
```

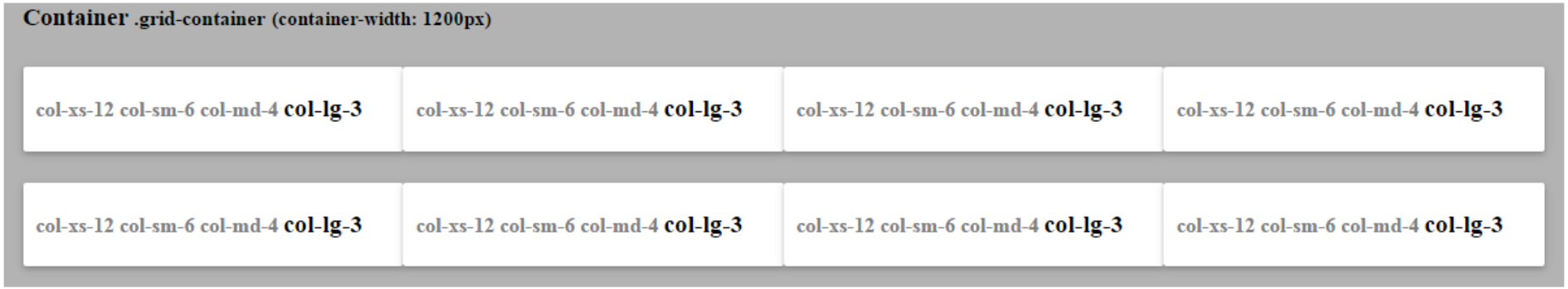
  - ResponsiveDesignLectureCode\index-responsive-design-grid-container.html

Umsetzung – Grid Ansatz

- Problem 2:
  - Auf kleineren Bildschirmen sind die Inhalte **nicht mehr übersichtlich abgebildet**
- Lösung: Kolonnendefinitionen bezüglich Bildschirmgrösse
  - Kolonnen-CSS-Klassen für 4 verschiedene Grössen
    - col-xs, col-sm, col-md, col-lg
    - Werden mithilfe des @media-Attributes jeweils nur für die entsprechenden Bildschirmgrössen definiert

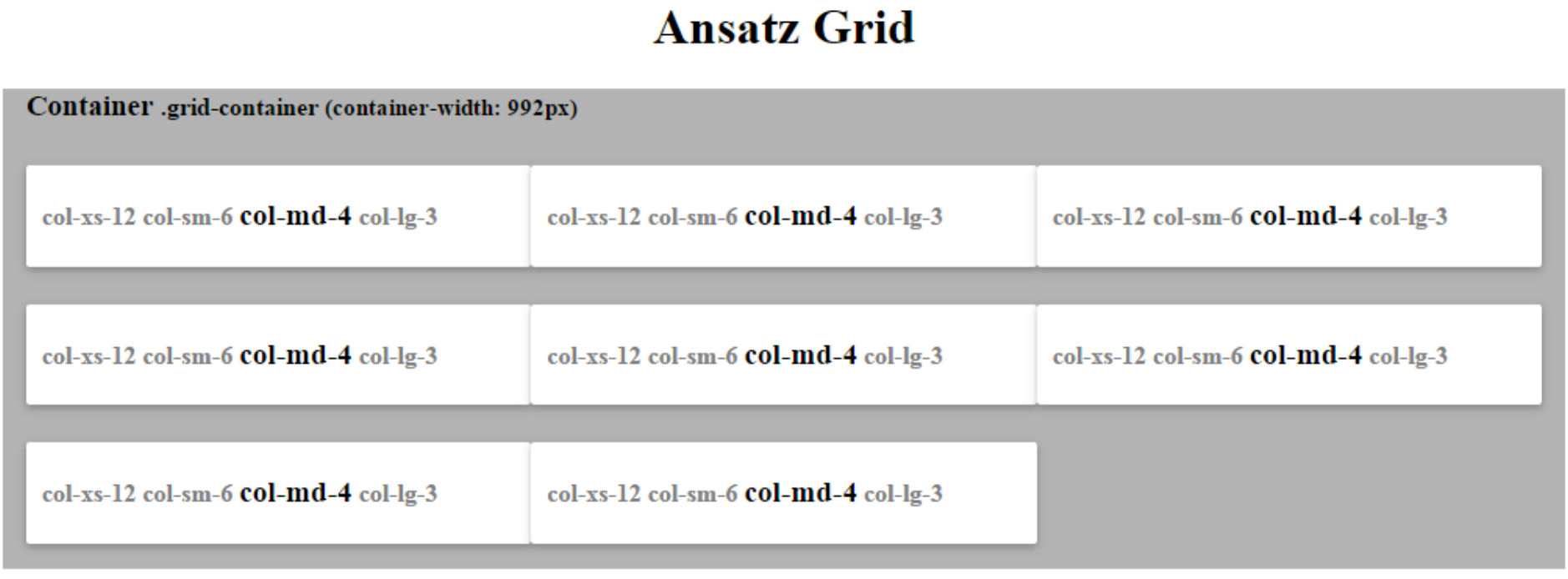
Umsetzung – Grid Ansatz

- **col-lg Klasse** ist aktiv, falls der Bildschirm grösser ist als 1200px
  - col-lg Definitionen in @media(min-width: 1200px)



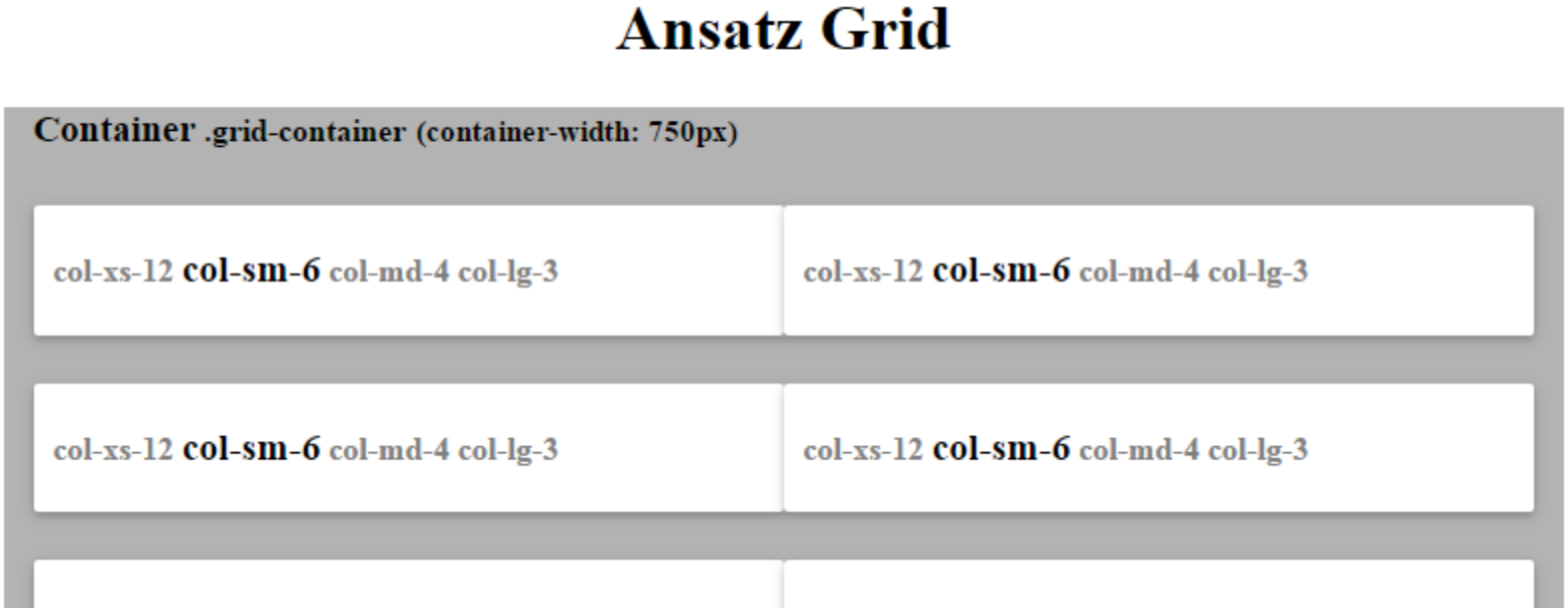
Umsetzung – Grid Ansatz

- **col-md Klasse** ist aktiv, falls der Bildschirm grösser ist als 992px
  - col-md Definitionen in @media(min-width: 992px)



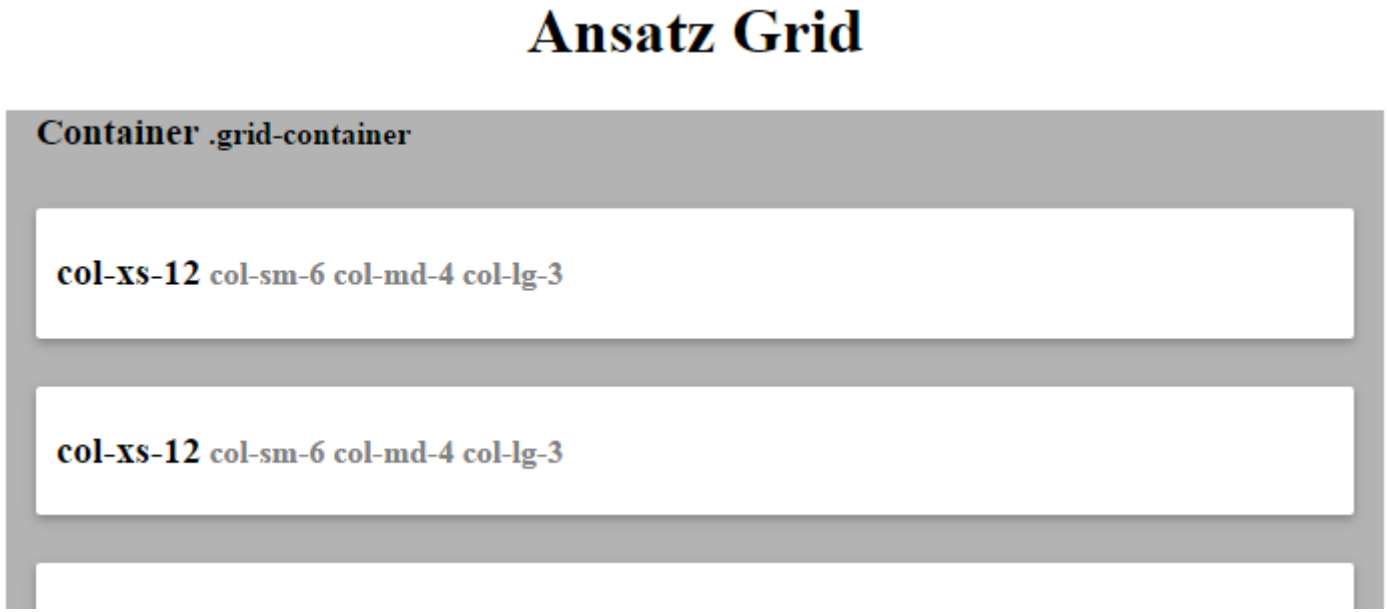
Umsetzung – Grid Ansatz

- **col-sm Klasse** ist aktiv, falls der Bildschirm grösser ist als 768px
  - col-sm Definitionen in @media (min-width: 768px)



Umsetzung – Grid Ansatz

- **col-xs Klasse** ist aktiv, falls der Bildschirm kleiner ist als 768px
  - col-xs Definitionen ohne @media



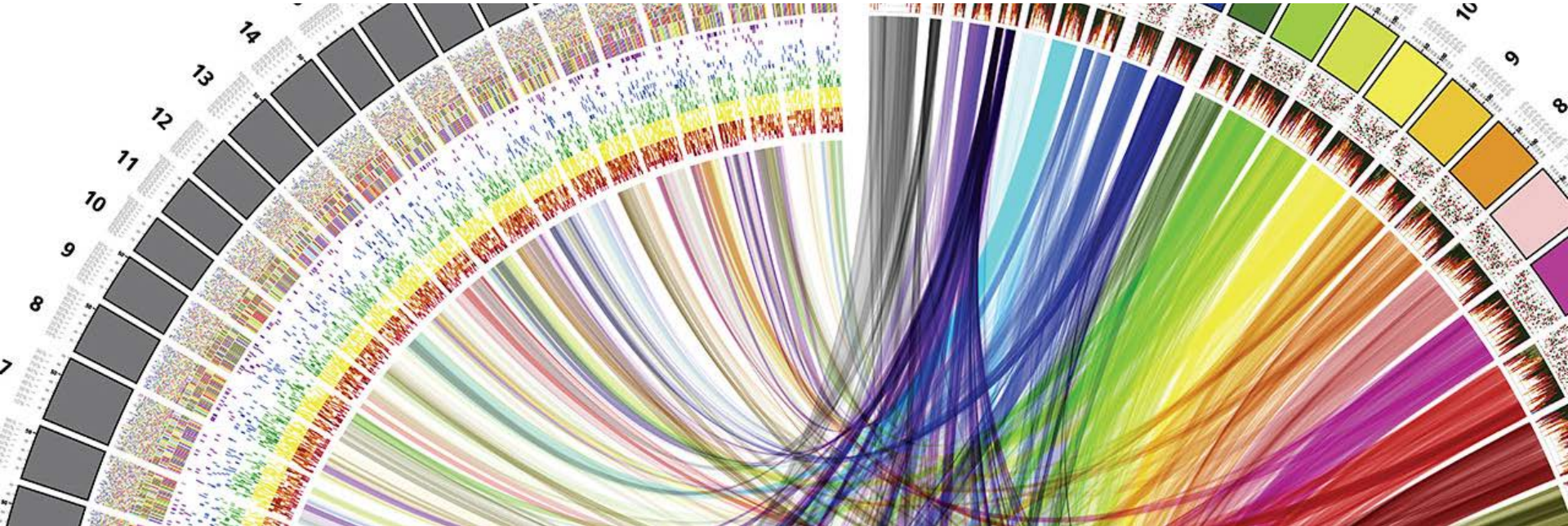
Umsetzung – Grid Ansatz

- ResponsiveDesignLectureCode\index-responsive-design-grid.html

- Grid als Responsive Design Lösung:
  - **Auf grösseren Bildschirmen** wird der Container eingeschränkt, so dass dieser eine definierte Maximalbreite nicht überschreitet
  - **Auf kleineren Bildschirmen** können die Kolonnen so definiert werden, dass sie untereinander abgebildet werden

Agenda

1. Einführung Responsive Web Design
2. Umsetzung
3. **Bootstrap**
4. D3.js und Responsive Web Design
5. Fragen und Antworten



Bootstrap

- HTML, CSS und JS Framework
- Vereinfacht das Web Design extrem. Es beinhaltet:
  - Das vorgestellte Gridlayout
  - Vordefinierte CSS Styles, Fonts (Icons), Javascript-Scripts und HTML-Komponenten

Bild von http://getbootstrap.com/

Bootstrap in Projekt einbinden

- Bootstrap unter http://getbootstrap.com/ downloaden
- jQuery unter https://jquery.com/download/ downloaden
  - Compressed jQuery 2.2.3
- Die CSS-, Font- und JS-Ordner von Bootstrap und das jquery-2.2.3.min.js in das Projekt kopieren
- Die Skripte durch folgende Zeilen im **index.html** einbinden:

```
<link rel="stylesheet" href="css/bootstrap.css">
<link rel="stylesheet" href="css/bootstrap-theme.css">
<script src="js/jquery-2.2.3.min.js"></script>
<script src="js/bootstrap.js"></script>
```

- Copy&Paste von der Bootstrap Webseite

jQuery

- Javascript Library
- Optimierte Funktionen für DOM(Document Object Model)-Selektionen, -Manipulationen und Event Handling
- http://callmenick.com/post/jquery-functions-javascript-equivalents

```
function showAlertMessage(type, title, message) {
    $('#alert').html('<b>' + title + '</b> ' + message);
    $('#alert').attr('class', 'alert alert-' + type);
    $('#alertWrapper').fadeIn().delay(2000).fadeOut(1000);
}
```

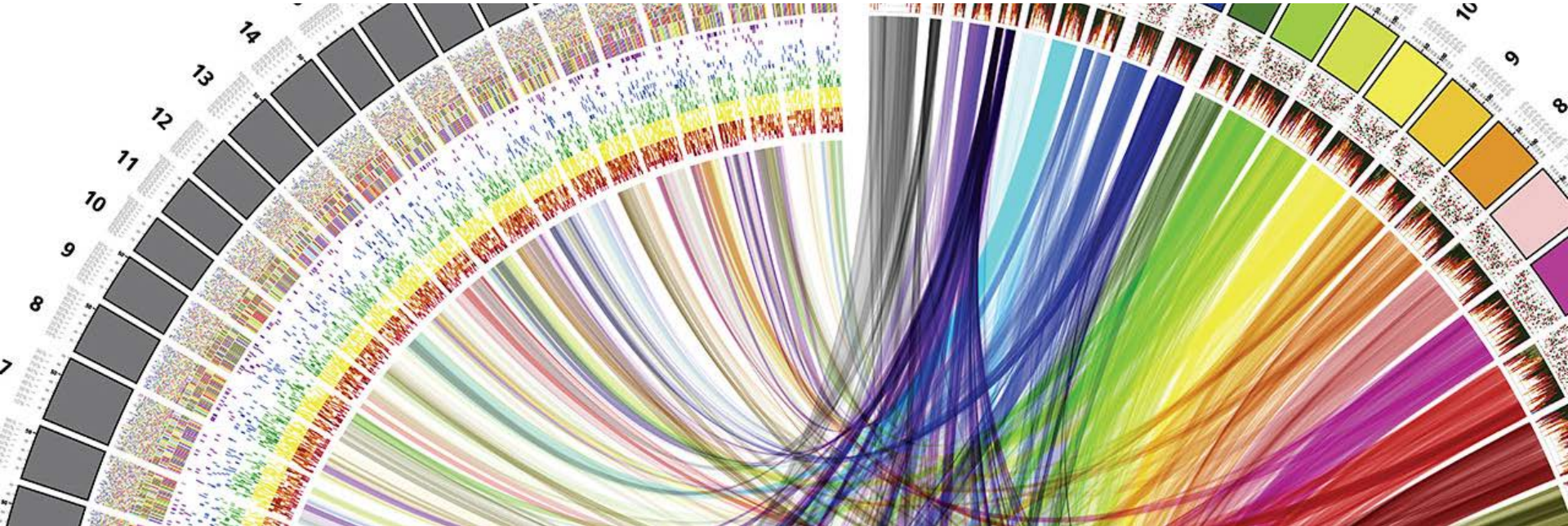
Bootstrap Beispiel

> ResponsiveDesignLectureCode\index-bootstrap-example.html

UNIVERSITÄT BERN

Agenda

1. Einführung Responsive Web Design
2. Umsetzung
3. Bootstrap
4. **D3.js und Responsive Web Design**
5. Fragen und Antworten



D3.js und Responsive Web Design

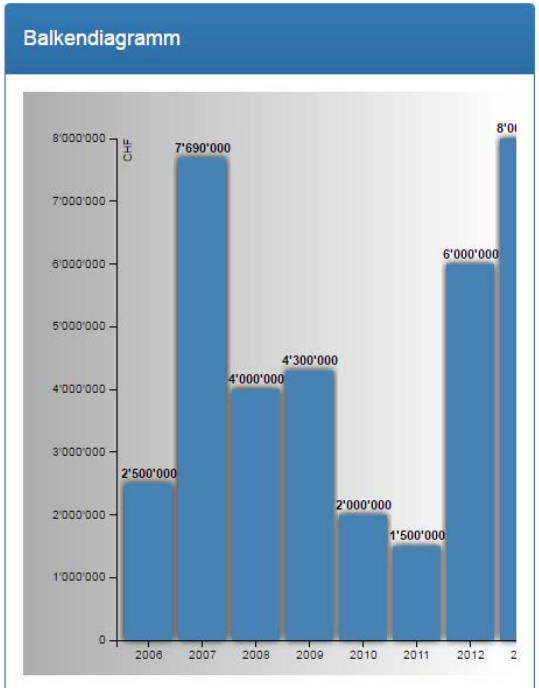
- ResponsiveDesignLectureCode\index-responsive-design-chart.html
- Die Breite und die Höhe der Charts müssen in **px** berechnet werden, da keine Prozentwerte erlaubt sind:

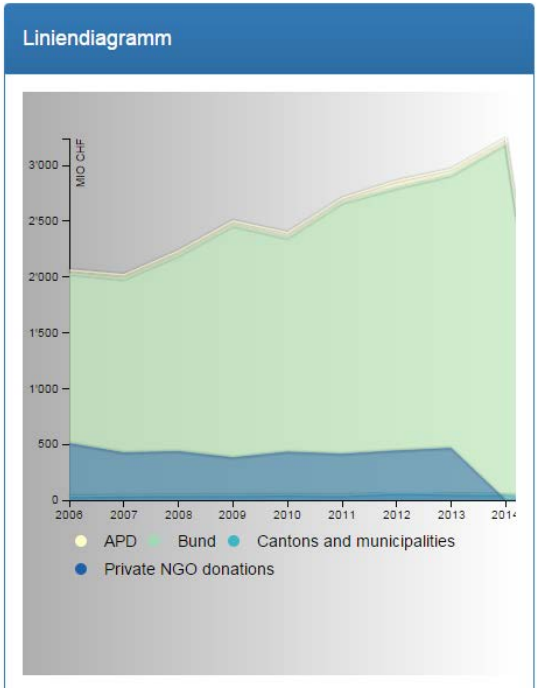
```
var svgWidth = $('#barchartDiv').width();
var svgHeight = $('#barchartDiv').height();
```

- Es ist wichtig, das Script am Ende des HTML-Files durchzuführen, da sonst die Dimensionen des **'#barchartDiv'** noch nicht bekannt sind

D3.js und Responsive Web Design

- Problem:
  - Bei Fenstervergrösserungen/-verkleinerungen werden die Diagramme nicht angepasst
  - Die Seite muss jedes mal aktualisiert werden





D3.js und Responsive Web Design

- Lösung:
  - Dynamische Anpassung der Diagramme durch **Javascript**
    - Implementierung eines **window resize listeners**

```
var resizeTimeout;
//Handle on window resize events for the charts
window.addEventListener('resize', function () {
    clearTimeout(resizeTimeout);
    resizeTimeout = setTimeout(function () {
        var svgWidth = $('#barchartDiv').width();
        var svgHeight = $('#barchartDiv').height();
        //Update chart with new svg width/height
        ...
    }, 500);
});
```

D3.js und Responsive Web Design - Resultat

- ResponsiveDesignLectureCode\index-responsive-design-chart-js-resize.html
- https://visual-eza.fdn.iwi.unibe.ch

Agenda

1. Einführung Responsive Web Design
2. Umsetzung
3. Bootstrap
4. D3.js und Responsive Web Design
5. **Fragen und Antworten**