



openHPI Responsive Webdesign

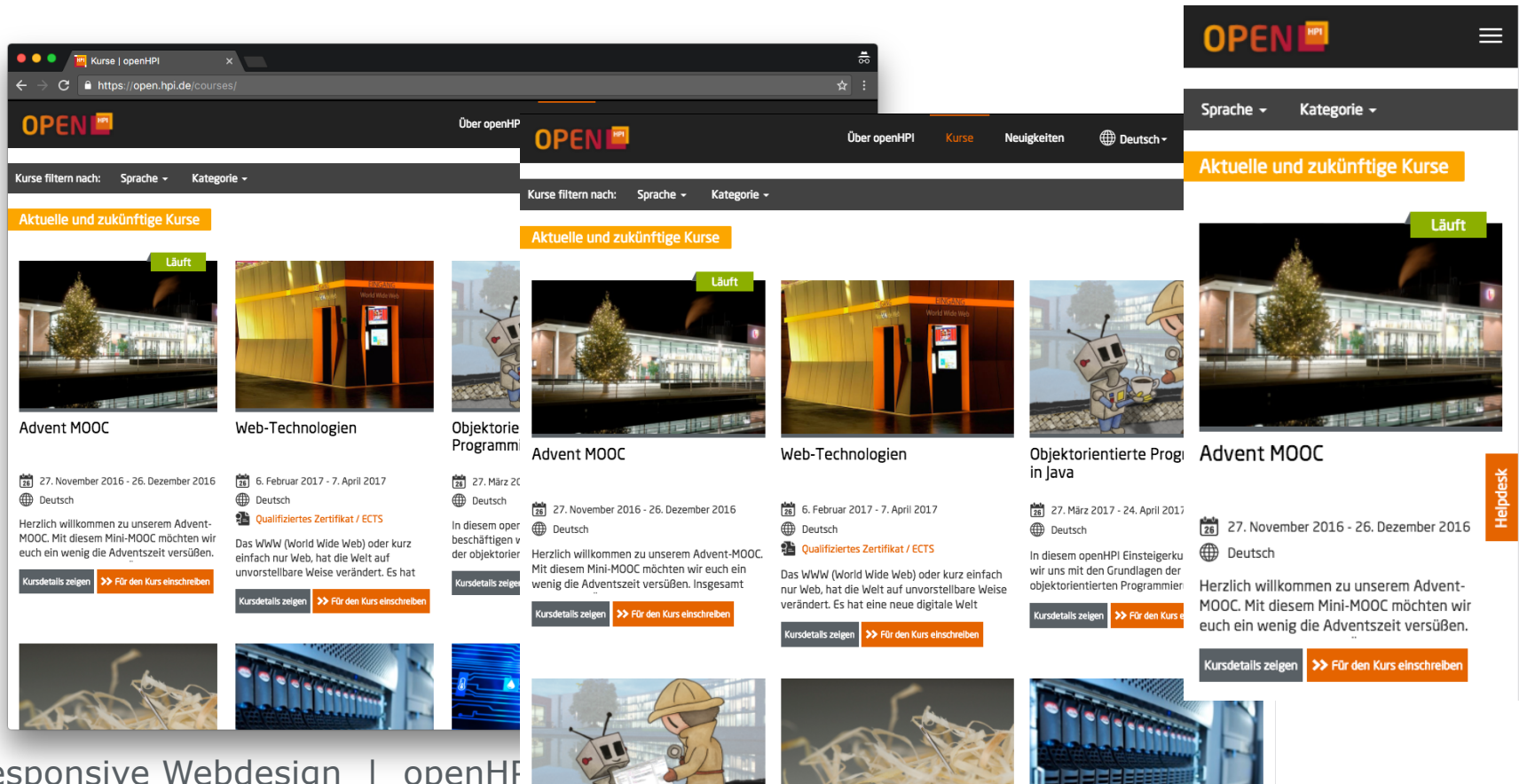
Prof. Dr. Christoph Meinel

Hasso-Plattner-Institut

Universität Potsdam

Responsive Webdesign (1/2)

Responsives („reagierendes“) **Webdesign** (RWD) ist **Paradigma** bei Erstellung von Webseiten → Webdesign soll auf Anforderungen verschiedener Endgeräte (Desktop, Tablet, Smartphone) reagieren können



Responsive Webdesign (2/2)

- **Ziel beim Responsive Design:** Webseite erstellen, die für alle Nutzer auf allen Geräten gut aussieht und gut zu bedienen ist
- RWD sollte technisch nur mit HTML und CSS realisiert werden – JavaScript ist in der Regel nicht nötig
- RWD sollte nicht darauf beruhen, auf kleineren Bildschirmen einfach Inhalte wegzulassen, sondern diese anders anzuordnen
- **Beispiel:** openHPI-Kursliste
 - Tablets: Bilder werden zunächst schmaler, bei kleineren Bildschirmen werden 3 statt 4 Kursen pro Zeile angezeigt
 - Smartphones:
 - Navigation hinter „Hamburger-Icon“ verborgen 
 - Kurse untereinander angeordnet
 - Links in Fußleiste werden untereinander angeordnet

Viewport (1/2)

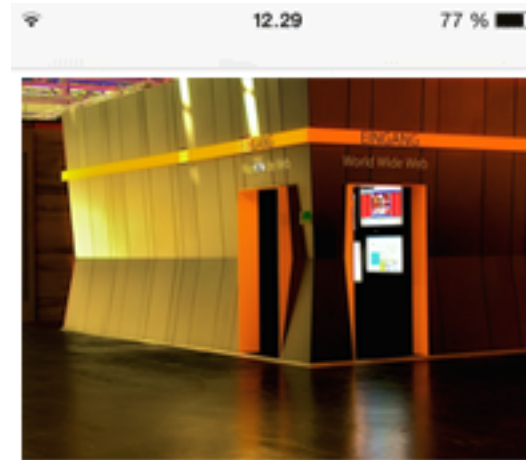
- **Früher:** Webseiten wurden für Desktop-Browser mit fester Breite gestaltet („Diese Seite ist optimiert für 1024x768 Pixel“)
 - Seiten waren für mobile Endgeräte zu breit
 - Mobile Browser verkleinern gesamte Seite → unlesbare Schrift
- **Einfache Maßnahme:** Angabe des **Viewports**
 - Viewport ist der für Nutzer sichtbare Bereich einer Webseite
 - variiert mit Breite des Bildschirms bzw. Fensters
- HTML 5 erlaubt Autoren, den Viewport selbst festzulegen:
`<meta name="viewport" content="width=device-width">`
 - **device-width** ist die Breite des genutzten Gerätes, Elemente werden in der Breite auf Bildschirmbreite begrenzt
- Zusätzlich beachten: Keine Elemente mit großer fester Breite verwenden (also z.B. `width="1000px"`), diese ragen immer über Viewport hinaus

Viewport (2/2)

Ohne Viewport-Angabe



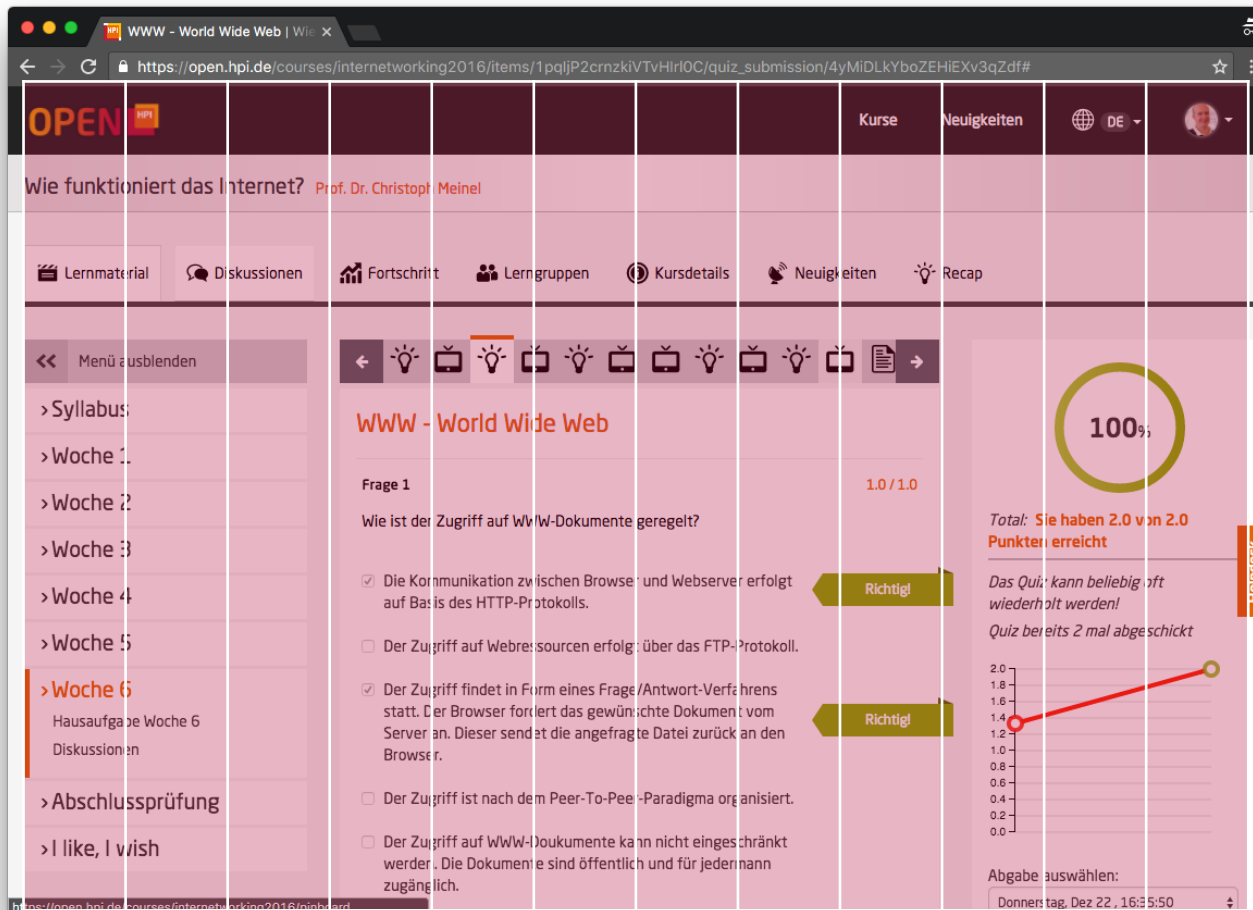
Mit Viewport-Angabe



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duiis dolore te feugait nulla facilisi. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet domine...

CSS Raster Layout (1/2)

- Webseiten werden häufig anhand eines horizontalen Rasters (Grid) gestaltet, die für verschiedene Bildschirmbreiten skaliert werden



12-spaltiges
Raster bei
openHPI

CSS Raster Layout (2/2)

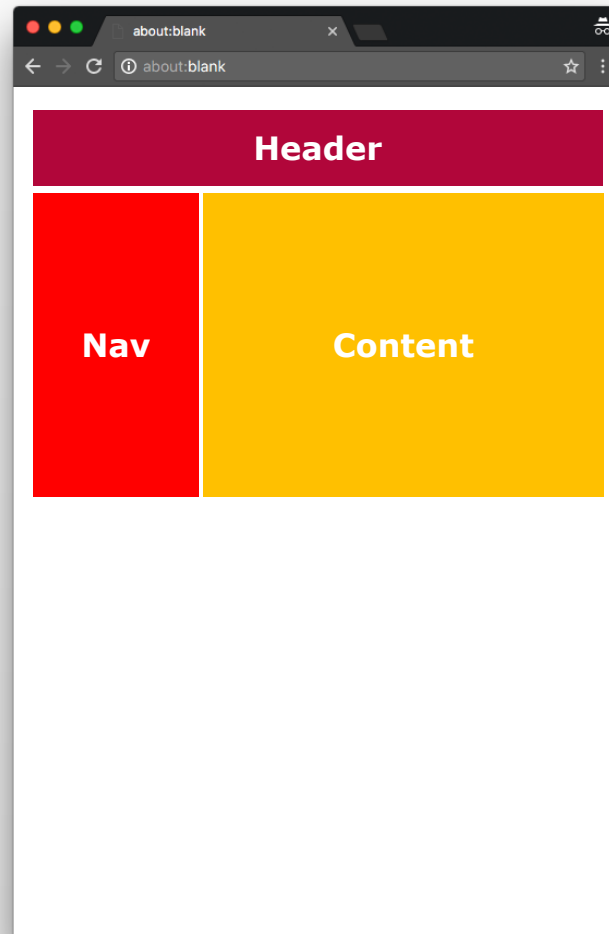
Code-Beispiel für CSS Raster

```
.col-1 {width:
.col-2 {width:
.col-3 {width:
.col-4 {width:
.col-5 {width:
.col-6 {width:
.col-7 {width:
.col-8 {width:
.col-9 {width:
.col-10 {width:
.col-11 {width:
.col-12 {width:

[class*="col-"]
    float: left
}

.row::after {
    content: ""
    clear: both
}
```

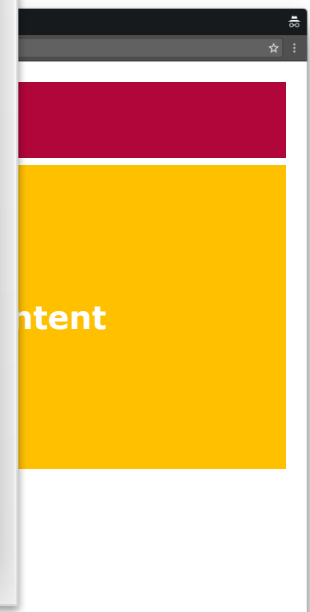
Raster bewirkt, dass
alle Elemente der
Webseite auf
schmalen Geräten
in der Breite
gleichmäßig
skaliert werden



Header</div>

Navigation</div>

Content</div>



CSS Media Queries (1/3)

- CSS 3 führt erstmals **Media Queries** ein
- **@media**-Blöcke erlauben Definition von CSS-Regeln, die nur gelten, wenn bestimmte Bedingung gelten
- **Beispiele:**
 - nur für Bildschirme bis maximale Smartphone-Breite

```
@media only screen and (max-width: 768px) {  
    img { width: 100% }  
}
```
 - nur für Bildschirme im Querformat

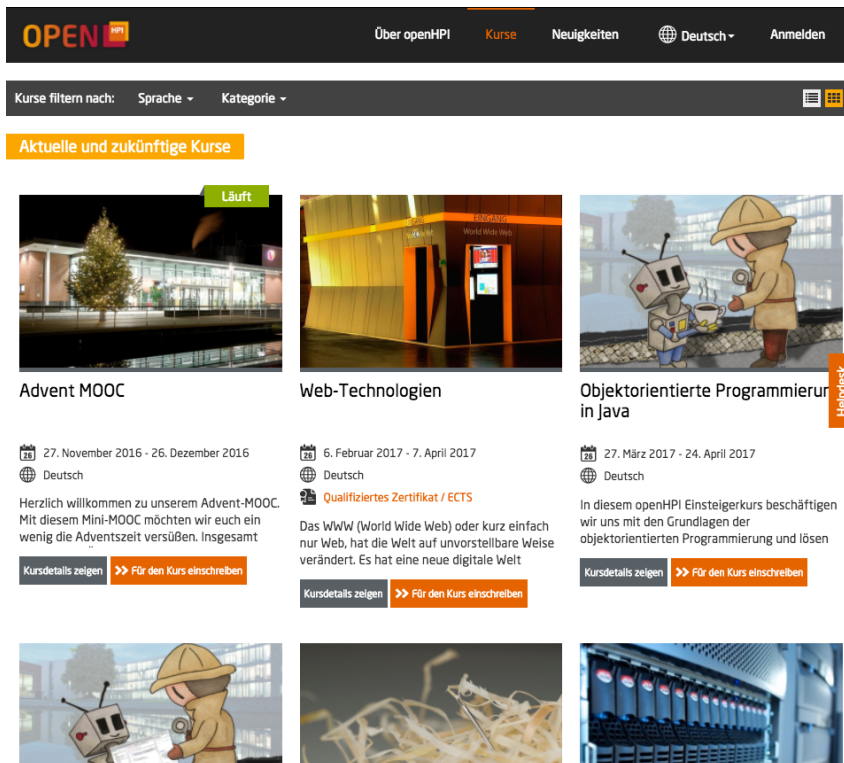
```
@media only screen and (orientation: landscape) { ... }
```
 - nur für Druck

```
@media only print { ... }
```
- Media Queries eignen sich auch für Einbinden externer CSS-Dateien:
`<link rel="stylesheet" href="druck.css" media="print">`

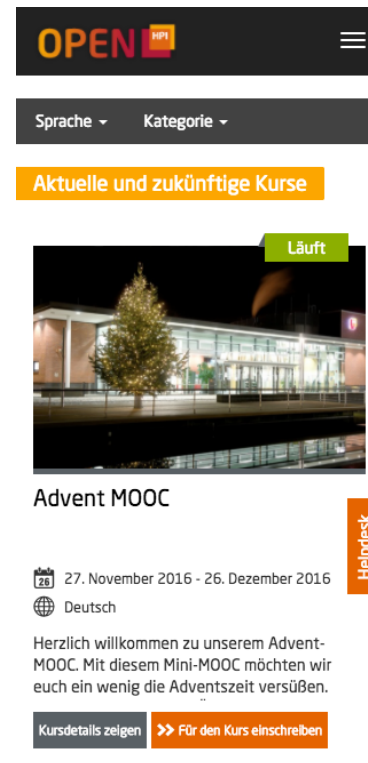
CSS Media Queries (2/3)

Kombination von Media Queries und CSS Raster Layout erlaubt tatsächliches reagierendes Design, z.B. bei openHPI

Spalten nebeneinander



Spalten untereinander



CSS Media Queries (3/3)

Kombination von Media Queries und CSS Raster Layout erlaubt tatsächliches reagierendes Design, z.B. bei openHPI

```
/* Für Desktop / Tablet */
.col-1 {width: 8.33%;}
.col-2 {width: 16.66%;}
.col-3 {width: 25%;}
[...]
.col-11 {width: 91.66%;}
.col-12 {width: 100%;}
```

Spalten nebeneinander

```
/* Für Smartphones */
@media only screen and (max-width: 768px) {
  [class*="col-"] {
    width: 100%;
  }
}
```

Spalten untereinander

- Bildschirmbreiten, an denen sich Gestaltung einer Webseite ändert, nennt man **Breakpoints**

CSS Frameworks

- Für Responsive Design und andere wiederkehrende CSS-Anwendungsfälle gibt es **Frameworks** – Sammlungen von vorgefertigten Gestaltungselementen und Hilfsmitteln
- Bekanntes Framework: **Bootstrap** (von Twitter)
 - kommt auch bei openHPI zum Einsatz
 - enthält neben CSS-Elementen auch eine Reihe nützlicher Javascript-Komponenten
 - Tools für Responsive Design:
 - umfassendes Grid-System mit Media Queries
 - erlaubt Ausblenden von Seiten-Elementen bei bestimmten Bildschirmbreiten
 - Hilfsmittel für responsive Images und Videos
 - bietet Unterstützung für → **CSS Präprozessoren**