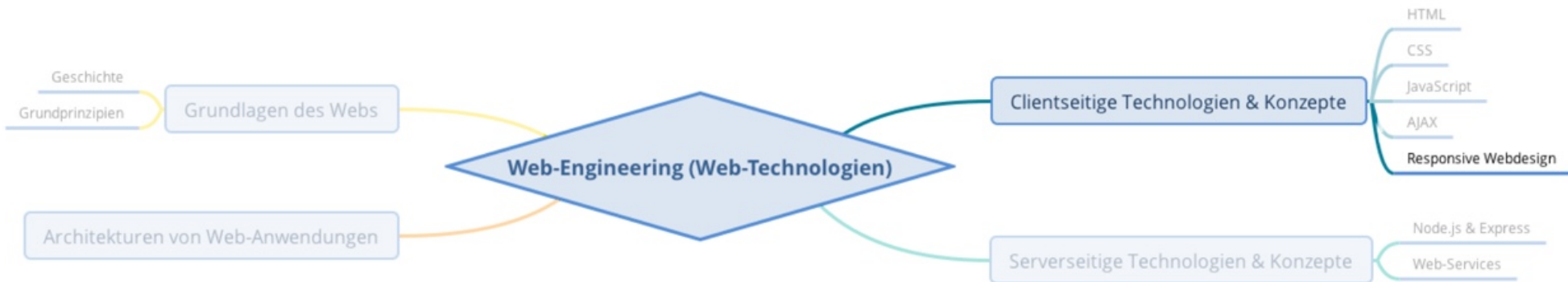




WEB- TECHNOLOGIEN

CLIENTSEITIGE TECHNOLOGIEN: RESPONSIVE WEBDESIGN

THEMEN DER VERANSTALTUNG



LERNZIELE

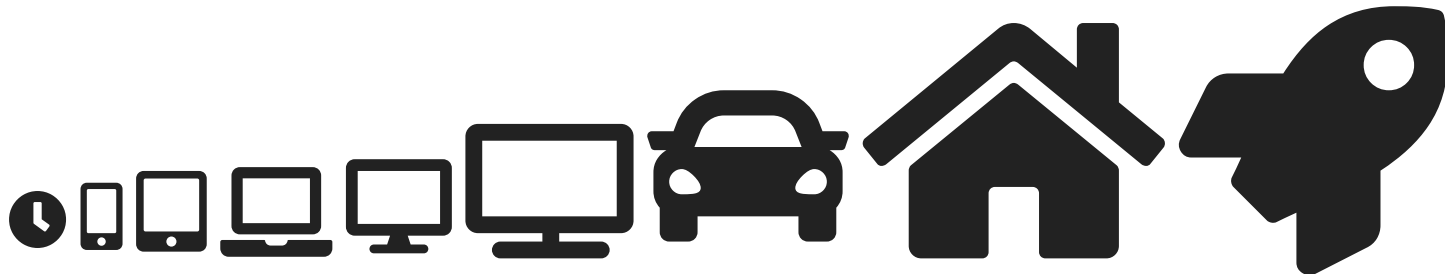
1. Verschiedene Arten von Layouts kennen und differenzieren können
2. HTML und CSS zur Erstellung responsiver Webseiten anwenden können

FRÜHER™

"Dieser Auftritt ist optimiert für Internet Explorer 4.x, 5.0 und Netscape ab 4.5x, bei einer Auflösung von 800 x 600 oder 1024 x 768 Pixeln, mittlerem Schriftgrad und 16,7 Mio. Farben."

➔ Hauptsächlich Desktop-Rechner

HEUTE



→ Große Vielfalt von Endgeräten


KRITERIEN BEIM WEBDESIGN

Bei Design und Entwicklung von Webseiten müssen heute eine Vielzahl von Kriterien berücksichtigt werden

Endgeräte

- Welche Endgeräte sollen unterstützt werden?
- z.B. Desktop-Rechner, Tablet, Smartphone

Browser

- Welche Browser sollen unterstützt werden bzw. sind verfügbar?
- Welche Funktionalitäten bieten die jeweiligen Browser? → caniuse.com 

KRITERIEN BEIM WEBDESIGN (2)

Displays der Endgeräte

Unterscheidende Faktoren bei Displays:

Faktor	Beispiele
Größe (in Zoll)	Smartphones: 4-6", Notebooks: 13-17"
Seitenverhältnis	16:9, 4:3
Auflösung (Pixelzahl)	1920 x 1080 (Full HD), 3480 x 2160 (4K)
Pixeldichte (in Pixel pro Zoll, ppi)	Smartphones: 300-600, Notebooks: 120-280
Ausrichtung	Hochformat (<i>portrait</i>), Querformat (<i>landscape</i>)

KRITERIEN BEIM WEBDESIGN (3)

Bedienung der Endgeräte

Wie werden die verschiedenen Endgeräte bedient? Beispiele:

- Maus
- Tastatur
- Touchscreen (Finger, Stift)
- Mikrofon (Sprache)

Netzanbindung

- In mobilen Netzen spielen z.B. geringe Übertragungsgeschwindigkeiten sowie beschränkte Datenvolumina eine Rolle
- Fehlen einer Netzanbindung (Offline-Fähigkeit)

KRITERIEN BEIM WEBDESIGN (4)

Viewport

- Einer Webseite steht nicht immer die komplette Bildschirmgröße zur Anzeige zur Verfügung:
 - Das Browserfenster kann ggf. verkleinert werden
 - Das Browserfenster stellt selbst noch Elemente dar (z.B. Adressleiste, Lesezeichenleiste)
- Der (visuelle) **Viewport** ist der tatsächlich sichtbare Bereich, welcher der Webseite zur Anzeige zur Verfügung steht
- Webseiten müssen für verschiedene Viewports funktionieren

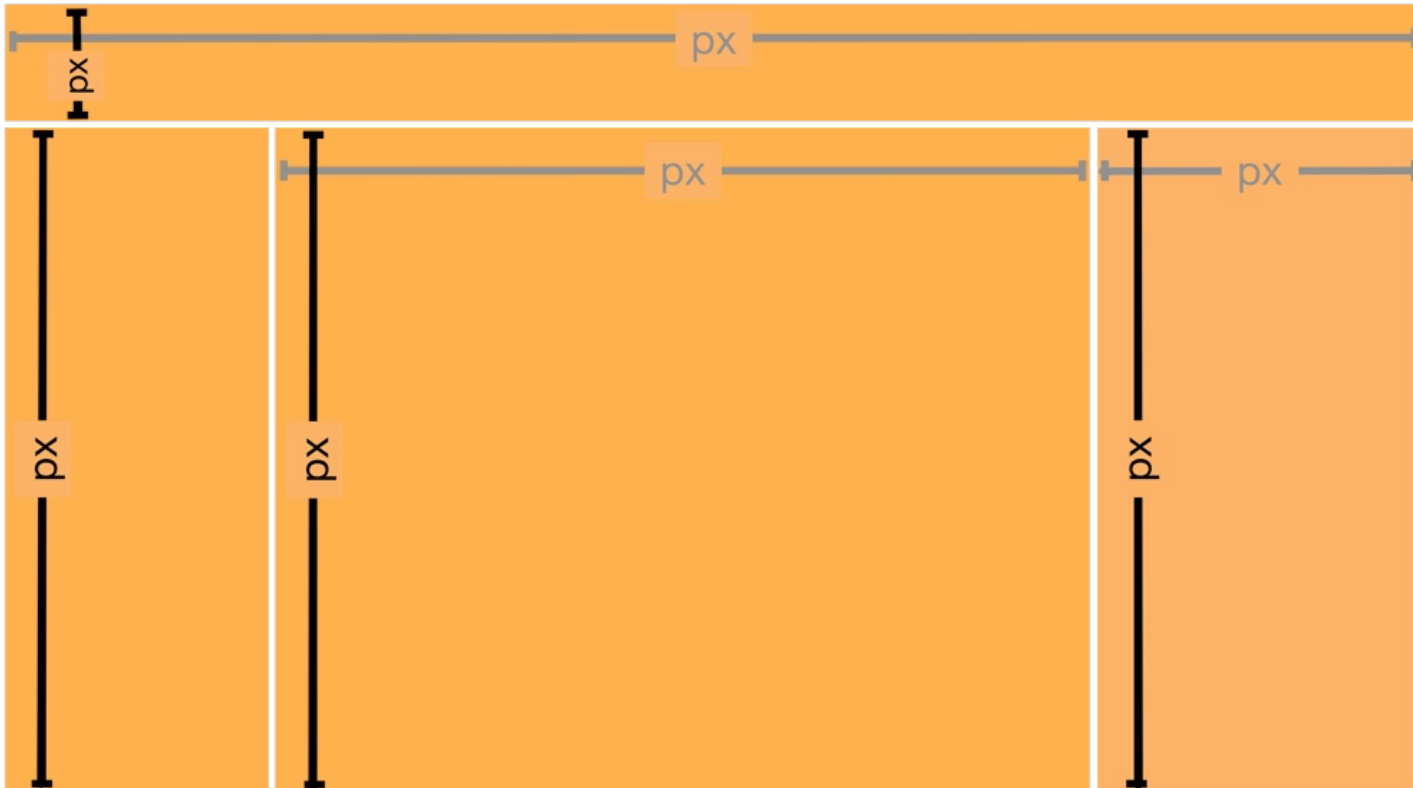
ARTEN VON LAYOUTS

Wir betrachten im Folgenden verschiedene Typen von Layouts und wie man diese umsetzen kann:

1. Fixes Layout
2. Fluides Layout
3. Adaptives Layout
4. Responsives Layout

FIXES LAYOUT

- Sämtliche Größenangaben in Pixeln
- Layout ist statisch und unabhängig vom verfügbaren Platz:
 - Rand bei zu großem Viewport
 - Abschneiden des Inhalts bei zu kleinem Viewport



FIXES LAYOUT: BEWERTUNG

- ➕ Das Layout kann pixelgenau gestaltet werden
- ➕ Wenig Aufwand bei der Umsetzung
- ➖ Layout passt sich nicht an verschiedene Displaygrößen, Viewports, etc. an
- ➖ Ggf. horizontales Scrollen notwendig (→ schlechte *User Experience*)
- ❓ *Anwendungsfälle?*
Anwendung in Situationen, bei denen nur für eine Displaygröße entwickelt werden muss, z.B.: Informationsterminal, Geldautomat

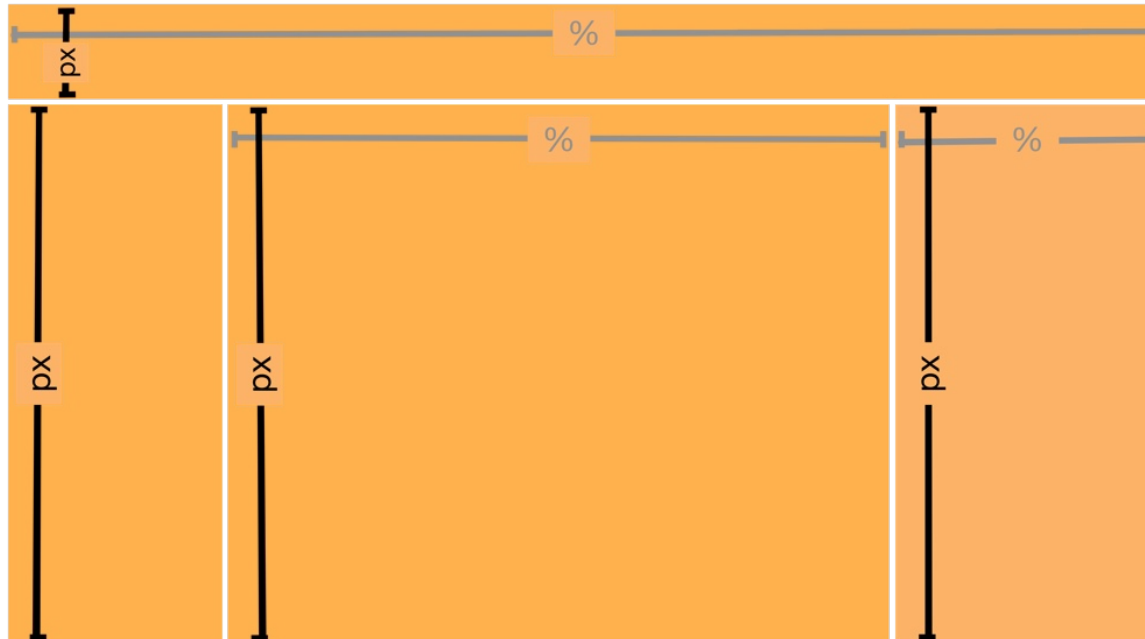
ARTEN VON LAYOUTS

Wir betrachten im Folgenden verschiedene Typen von Layouts und wie man diese umsetzen kann:

1. Fixes Layout
2. Fluides Layout
3. Adaptives Layout
4. Responsives Layout

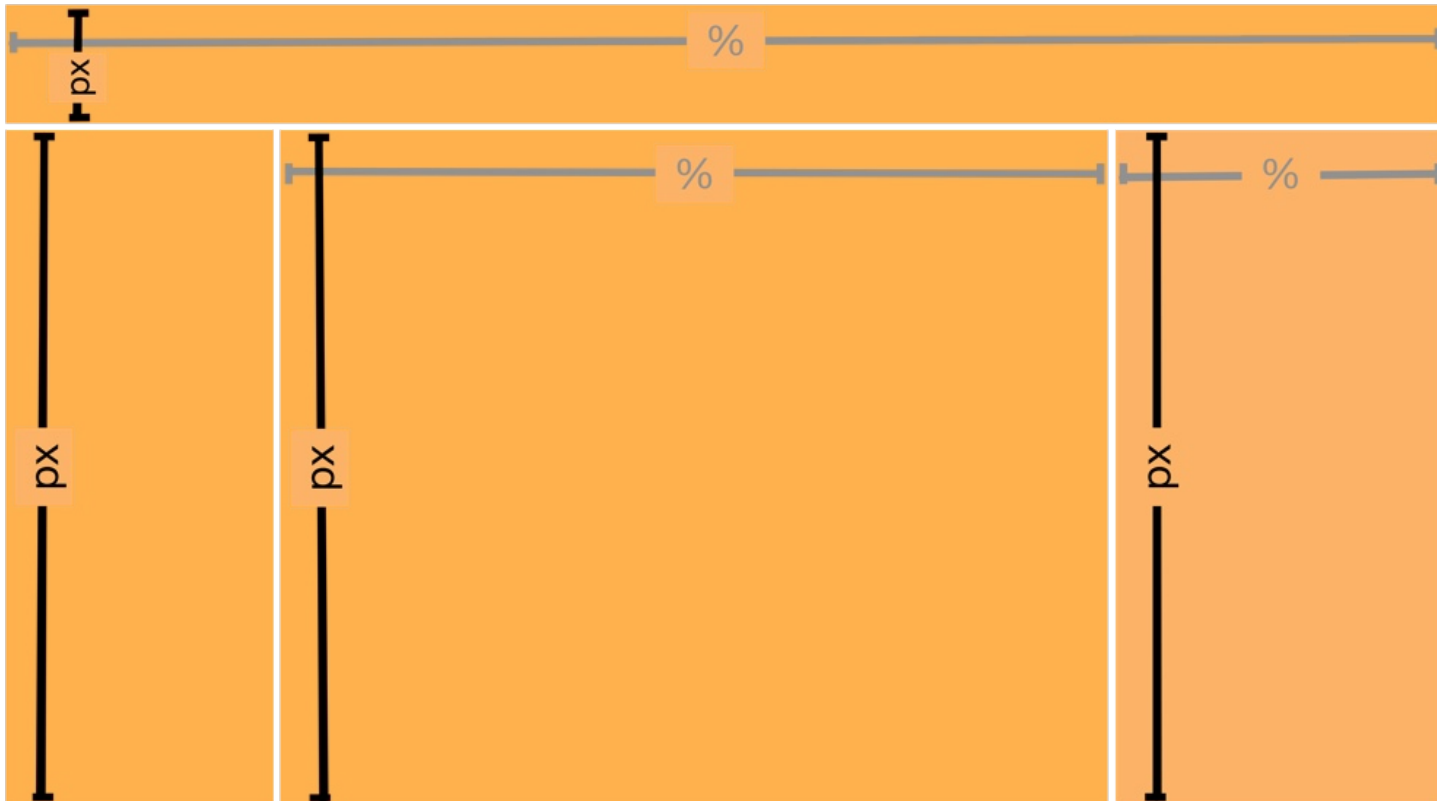
FLUIDES LAYOUT

- Auch: *Liquides Layout*
- Metapher: Flüssigkeit, die in ein Gefäß gegossen wird und sich gleichmäßig ausbreitet
- Größenangaben (z.B. Spaltenbreiten) mit relativen Angaben wie %, em)



FLUIDES LAYOUT (2)

- Angabe in %: Layout passt sich an den Viewport an
- Angaben in em: Layout passt sich ähnlich der Zoom-Funktion im Browser an die Schriftgröße an (auch *elastisches Layout* genannt)



FLUIDES LAYOUT: BEWERTUNG

- ➕ Anpassung an verschiedene Displaygrößen und Viewports
- ➕ Keine abgeschnittenen Inhalte oder ungewollte Ränder
- ➕ Vermeidung von horizontalem Scrollen
- ➖ Designaufwand kann höher sein: Flexibilität muss berücksichtigt werden
- ➖ Ggf. nicht optimale Darstellung bei sehr kleinen oder sehr großen Viewports
- ➖ Ggf. höherer Entwicklungsaufwand, da auch Gestaltungselemente wie z.B. Bilder flexibel sein müssen

FLEXIBLE BILDER

- Erinnerung (HTML): Größe von Bildern immer angeben, damit keine "Flackereffekte" entstehen
- Jedoch: Bei absoluten Angaben für `width` und `height` ist das Bild nicht flexibel (es behält immer die gleiche Größe bei)

Eine mögliche Lösung mit CSS:

```
img {  
    /* Maximale Breite des Bildes darf höchstens  
       der Breite des umgebenen Elements entsprechen */  
    max-width: 100%;  
    /* Höhe automatisch proportional zur Breite  
       anpassen */  
    height: auto;  
}
```

FLEXIBLE BILDER (2)

Eine mögliche Lösung mit CSS:

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

- ! Je nach Layout kann bei `max-width` ein kleinerer Wert als 100% sinnvoll sein
- Nachteil: Es wird weiterhin das vollständige Bild in Originalgröße geladen → bei mobilen oder schlechten Datenverbindungen nicht unbedingt optimal!

ARTEN VON LAYOUTS

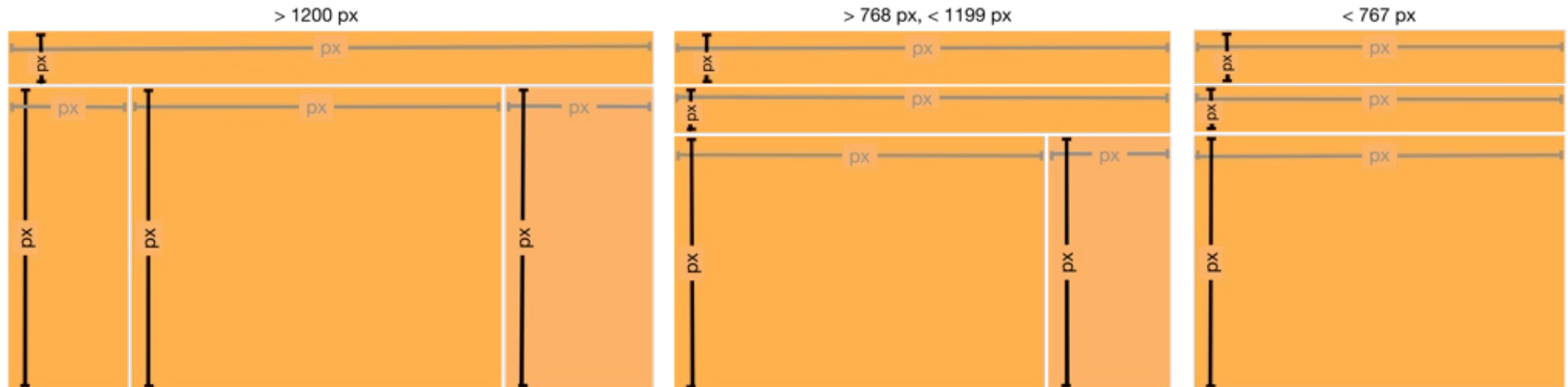
Wir betrachten im Folgenden verschiedene Typen von Layouts und wie man diese umsetzen kann:

1. Fixes Layout
2. Fluides Layout
3. **Adaptives Layout**
4. Responsives Layout

ADAPTIVES LAYOUT

- Bereitstellung *mehrerer* fixer Layouts für spezifische Displaygrößen (z.B. Desktop, Tablet, Smartphone)
- Realisiert über sogenannte **Breakpoints** (Umbruchpunkte), an denen die Ansicht auf ein anderes fixes Layout umspringt


Beispiel: Breakpoints bei Viewport-Breite von 1200 und 768 Pixeln



ADAPTIVES LAYOUT: BEWERTUNG

- ➕ Durch Verwendung fixer Layouts: Pixelgenaue Gestaltung
- ➕ Unterstützung mehrerer Endgeräte
- ➕ Es wird ggf. Entwicklungsaufwand eingespart, da Gestaltungselemente nicht flexibel sein müssen
- ➖ Optimierung nur für bestimmte Endgeräte → alle Anderen erhalten ggf. eine suboptimale Darstellung
- ➖ Ggf. höherer Aufwand, da mehrere Layouts erstellt werden müssen

BREAKPOINTS MIT CSS DEFINIEREN

- CSS bietet mit den **Media Queries** (Medienabfragen) ein [Modul](#) , mit welchem sich Breakpoints realisieren lassen
- Mit Hilfe einer Media Query können CSS-Regeln abhängig von Medientypen (*Media Types*) und Medienmerkmalen (*Media Features*) eingebunden bzw. aktiviert werden

MEDIA QUERIES: SYNTAX

- Eine Media Query besteht aus der optionalen Abfrage eines Media Types, gefolgt von einer beliebigen Menge von mit and verknüpften Ausdrücken:

```
<mediaquery> ::= [only|not]? <mediatype> [and <expression>]*  
                | <expression> [and <expression>]*
```

- Ein Ausdruck wird in Klammern notiert und enthält ein Media Feature sowie optional einen zugehörigen Wert (vereinfachte Syntax):

```
<expression> ::= (<mediafeature> [: <value>]?)
```

MEDIA TYPES

Unterscheidung verschiedener Ausgabemedien:

Media Type	Bedeutung
all	Alle Endgeräte
print	Drucker
speech	Ausgabe per Sprache, z.B. Screenreader
screen	Bildschirme

MEDIA FEATURES

Beispiele abfragbarer Medienmerkmale:

Media Feature	Bedeutung
<code>width,min-width,max-width</code>	Viewport-Breite
<code>height,min-height,max-height</code>	Viewport-Höhe
<code>device-width,min-device-width,max-device-width</code>	Display-Breite
<code>device-height,min-device-height,max-device-height</code>	Display-Höhe
<code>orientation</code>	Ausrichtung (Werte: <code>portrait</code> oder <code>landscape</code>)

MEDIA QUERIES: BEISPIELE

Bildschirm mit Viewport-Breite ≥ 1024 Pixel und Landscape-Ausrichtung:

```
screen and (min-width:1024px) and (orientation:landscape)
```

Kein Drucker mit Farbausgabe:

```
not print and (color)
```

! not negiert immer die gesamte Media-Query!

Bildschirm mit Viewport-Breite ≤ 480 Pixel:

```
only screen and (max-width:480px)
```

! only sorgt dafür, dass ältere Browser die Media Query ignorieren!

MEDIA QUERIES ANWENDEN

Bedingtes Einbinden externer Stylesheets per <link>-Element:

```
[...]  
<head>  
  <!-- Stylesheet wird nur eingebunden, wenn die Media Query  
       zutrifft (heruntergeladen wird es allerdings immer!) -->  
  <link rel="stylesheet" type="text/css" href="mobile.css"  
        media="screen and (max-width:480px)">  
</head>  
[...]
```

MEDIA QUERIES ANWENDEN (2)

Bedingtes Einbinden interner Stylesheets per <style>-Element:

```
[...]
<head>
  <!-- Stylesheet wird nur eingebunden, wenn die Media Query
       zutrifft -->
  <style type="text/css" media="screen and (min-width:960px)">
    [...]
  </style>
</head>
[...]
```

MEDIA QUERIES ANWENDEN (3)

Innerhalb von Stylesheets per @media-Regel:

```
nav {  
  display: flex;  
  flex-direction: row;  
}  
  
@media screen and (max-width:480px) and (orientation:portrait) {  
  /* Für kleinere Viewports werden die Elemente im Flex-Container  
    untereinander angezeigt */  
  nav {  
    flex-direction: column;  
  }  
}
```

- ! Eine @media-Regel kann mehrere CSS-Regeln enthalten, die nur eingebunden werden, wenn die zugehörige Media Query zutrifft.

VIEWPORT (DESKTOP)

- Der **Viewport** ist der tatsächlich sichtbare Bereich, welcher einer Webseite zur Anzeige zur Verfügung steht
- Entspricht dem Browserfenster abzüglich eigener Elemente wie z.B. Fensterdekoration, Adressleiste, Lesezeichenleiste

VIEWPORT (DESKTOP)

The screenshot shows the desktop version of the Fachhochschule Dortmund website. The browser window has a single tab titled 'Fachhochschule Dortmund' and the address bar shows 'https://www.fh-dortmund.de/de/index.php'. The website header includes the logo 'Fachhochschule Dortmund' with the tagline 'University of Applied Sciences and Arts' and the slogan 'we focus on students'. A search bar and a language selector are on the right. A horizontal navigation menu contains links to 'Startseite', 'Architektur', 'Design', 'Elektrotechnik', 'Informatik', 'Maschinenbau', 'Angewandte Sozialwissenschaften', 'Wirtschaft', and 'Informationstechnik'. The main content area is divided into three columns. The left column lists links for 'Studieninteressierte', 'Studierende', 'Promovierende', 'Presse', 'Forschung und Transfer', 'IDIAL', 'Hochschule', and 'Jobs & Karriere'. The middle column features a large grid of images representing various university activities, with text overlays such as '10. INTERNATIONALE WOCHEN', 'INKLUSION', 'Tage Talente', 'JUMEN', and 'DIGITAL'. The right column lists news items in German, including 'Design und Gender', 'Filmvorführung mit Stéphane Riethauser', 'Gleichstellung', 'Jubiläumsaktion zum Frauenwahlrecht', 'Digitale Woche Dortmund', '„Rover Hackathon“ und „Smart Energy“', 'Vortragsreihe', 'Gehört der Islam zu Deutschland?', 'Akademische Jahresfeier', 'Würdigung für besondere Leistungen', 'ruhrvalley', 'Mittelstand trifft Digitalisierung', 'Digitaler Wandel', 'QuartiersNETZ als Modellprojekt', 'Internationale Woche', 'FH begrüßte Gäste aus aller Welt', 'Genderforschung', 'Wie junge Männer über Sexualität denken', 'TalentKolleg Ruhr', 'Einladung zur Meilensteinveranstaltung', 'News', and 'Weitere aktuelle Beiträge'. At the bottom, there are logos for 'Deutschland STIPENDIUM', 'EUA', 'CODE of CONDUCT', 'FH-Dortmund', and 'HRK-Audit'.

Fachhochschule Dortmund
University of Applied Sciences and Arts
we focus on students

Suche
Schnellzugriff
Sprache auswählen | ▼

Startseite Architektur Design Elektrotechnik Informatik Maschinenbau Angewandte Sozialwissenschaften Wirtschaft Informationstechnik

Studieninteressierte
Studierende
Promovierende
Presse
Forschung und Transfer
IDIAL
Hochschule
Jobs & Karriere

Design und Gender
Filmvorführung mit Stéphane Riethauser
Gleichstellung
Jubiläumsaktion zum Frauenwahlrecht
Digitale Woche Dortmund
„Rover Hackathon“ und „Smart Energy“
Vortragsreihe
Gehört der Islam zu Deutschland?
Akademische Jahresfeier
Würdigung für besondere Leistungen
ruhrvalley
Mittelstand trifft Digitalisierung
Digitaler Wandel
QuartiersNETZ als Modellprojekt
Internationale Woche
FH begrüßte Gäste aus aller Welt
Genderforschung
Wie junge Männer über Sexualität denken
TalentKolleg Ruhr
Einladung zur Meilensteinveranstaltung
News
Weitere aktuelle Beiträge

Eingeloggt als Sven Jörges.
BildungsOffensive
Erstsemester
FHDO hilft
Studienplatzportal

Deutschland STIPENDIUM
EUA
CODE of CONDUCT
FH-Dortmund
HRK-Audit

VIEWPORT (DESKTOP)

The screenshot shows the desktop version of the Fachhochschule Dortmund website. The browser window has a single tab titled 'Fachhochschule Dortmund' and the address bar shows 'https://www.fh-dortmund.de/de/index.php'. The website header features the 'Fachhochschule Dortmund' logo with the tagline 'University of Applied Sciences and Arts' and the slogan 'we focus on students'. A 'Viewport' label is in the top right. A search bar and a language selector ('Sprache auswählen') are also present. A horizontal navigation menu includes links for 'Startseite', 'Architektur', 'Design', 'Elektrotechnik', 'Informatik', 'Maschinenbau', 'Angewandte Sozialwissenschaften', 'Wirtschaft', and 'Informationstechnik'. The main content area is divided into three columns. The left column lists links for 'Studieninteressierte', 'Studierende', 'Promovierende', 'Presse', 'Forschung und Transfer', 'IDIAL', 'Hochschule', and 'Jobs & Karriere'. The middle column features a large grid of colorful images representing various university activities and events. The right column contains a list of news items, each with a date and a title, such as 'Design und Gender', 'Filmvorführung mit Stéphane Riethauser', 'Gleichstellung', 'Jubiläumsaktion zum Frauenwahlrecht', 'Digitale Woche Dortmund', '„Rover Hackathon“ und „Smart Energy“', 'Vortragsreihe', 'Gehört der Islam zu Deutschland?', 'Akademische Jahresfeier', 'Würdigung für besondere Leistungen', 'ruhrvalley', 'Mittelstand trifft Digitalisierung', 'Digitaler Wandel', 'QuartiersNETZ als Modellprojekt', 'Internationale Woche', 'FH begrüßte Gäste aus aller Welt', 'Genderforschung', 'Wie junge Männer über Sexualität denken', 'TalentKolleg Ruhr', 'Einladung zur Meilensteinveranstaltung', 'News', and 'Weitere aktuelle Beiträge'. At the bottom of the page, there are several logos, including 'Deutschland STIPENDIUM', 'EUA', 'CODE of CONDUCT', 'FH-Dortmund', and 'HRK-Audit'.

Fachhochschule Dortmund
University of Applied Sciences and Arts
we focus on students

Viewport

Suche
Schnellzugriff
Sprache auswählen | ▼

Startseite Architektur Design Elektrotechnik Informatik Maschinenbau Angewandte Sozialwissenschaften Wirtschaft Informationstechnik

Studieninteressierte
Studierende
Promovierende
Presse
Forschung und Transfer
IDIAL
Hochschule
Jobs & Karriere

Design und Gender
Filmvorführung mit Stéphane Riethauser
Gleichstellung
Jubiläumsaktion zum Frauenwahlrecht
Digitale Woche Dortmund
„Rover Hackathon“ und „Smart Energy“
Vortragsreihe
Gehört der Islam zu Deutschland?
Akademische Jahresfeier
Würdigung für besondere Leistungen
ruhrvalley
Mittelstand trifft Digitalisierung
Digitaler Wandel
QuartiersNETZ als Modellprojekt
Internationale Woche
FH begrüßte Gäste aus aller Welt
Genderforschung
Wie junge Männer über Sexualität denken
TalentKolleg Ruhr
Einladung zur Meilensteinveranstaltung
News
Weitere aktuelle Beiträge

Eingeloggt als Sven Jörges.
BildungsOffensive
Erstsemester
FHDO hilft
Studienplatzportal

Deutschland STIPENDIUM
EUA
CODE of CONDUCT
FH-Dortmund
HRK-Audit
Internationalisierung der Hochschule 2019/20

VIEWPORT (MOBILES WEB)

- Die Anzeige von Webseiten auf mobilen Geräten unterliegt anderen Voraussetzungen, z.B.:
 - Kleinere Bildschirme
 - Browser-Apps werden typischerweise im Vollbildmodus angezeigt
- Auch nicht für solche Endgeräte optimierte Webseiten sollen dennoch anzeigbar und verwendbar sein

LAYOUT-VIEWPORT UND VISUELLER VIEWPORT

Aus dieser Motivation heraus wurde eine Unterscheidung zwischen zwei Arten von Viewports eingeführt:

1. Layout-Viewport

- Viewport, auf welchem die Rendering-Engine des Browsers operiert
- Gibt eine Basisgröße der Webseite vor (browserspezifisch, z.B. 980px)
- Anhand dieser Größe wird das Layout aufgebaut

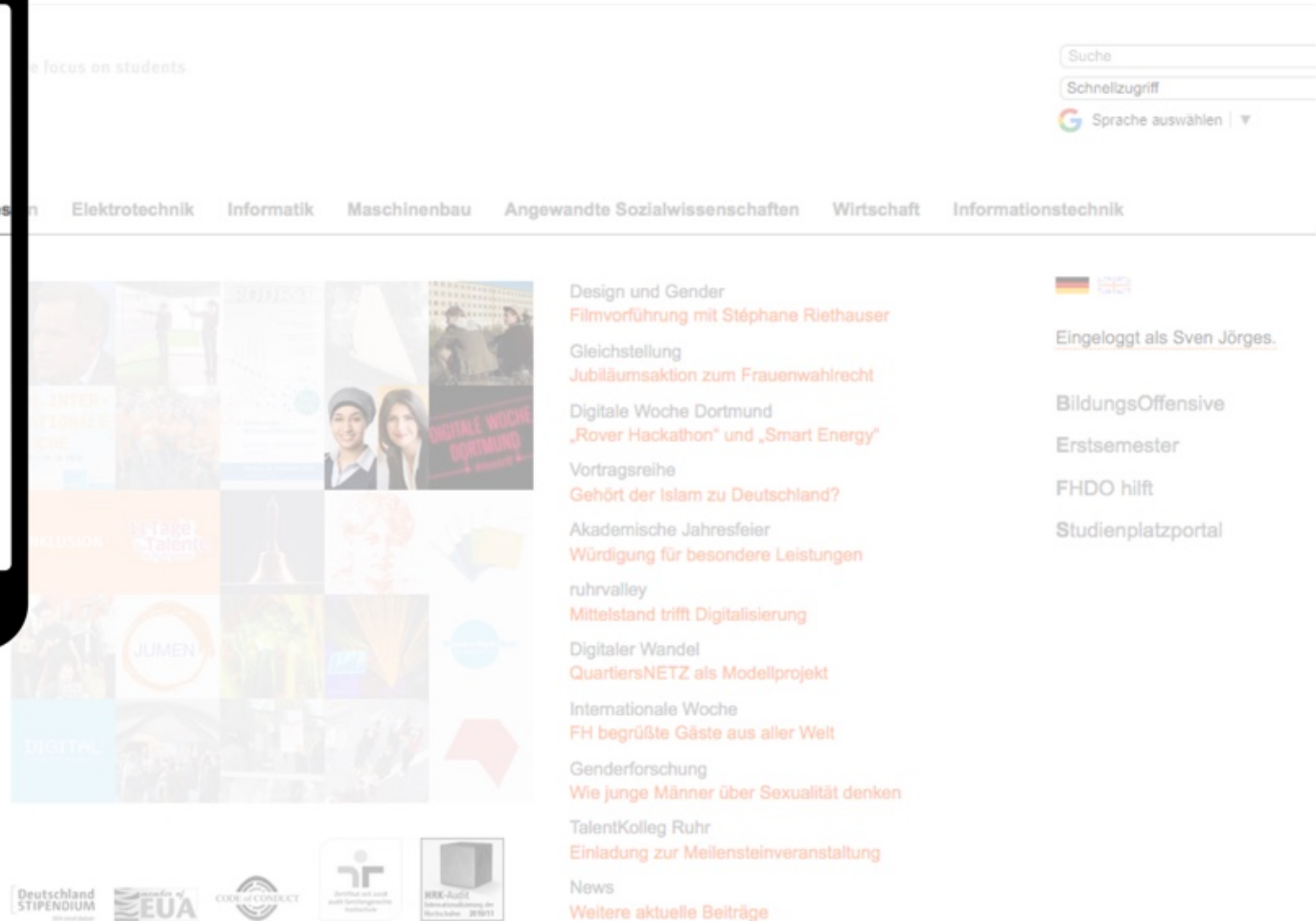
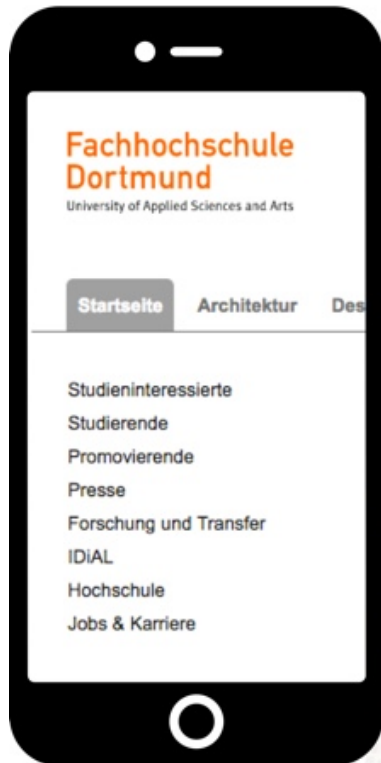
LAYOUT-VIEWPORT UND VISUELLER VIEWPORT

Aus dieser Motivation heraus wurde eine Unterscheidung zwischen zwei Arten von Viewports eingeführt:

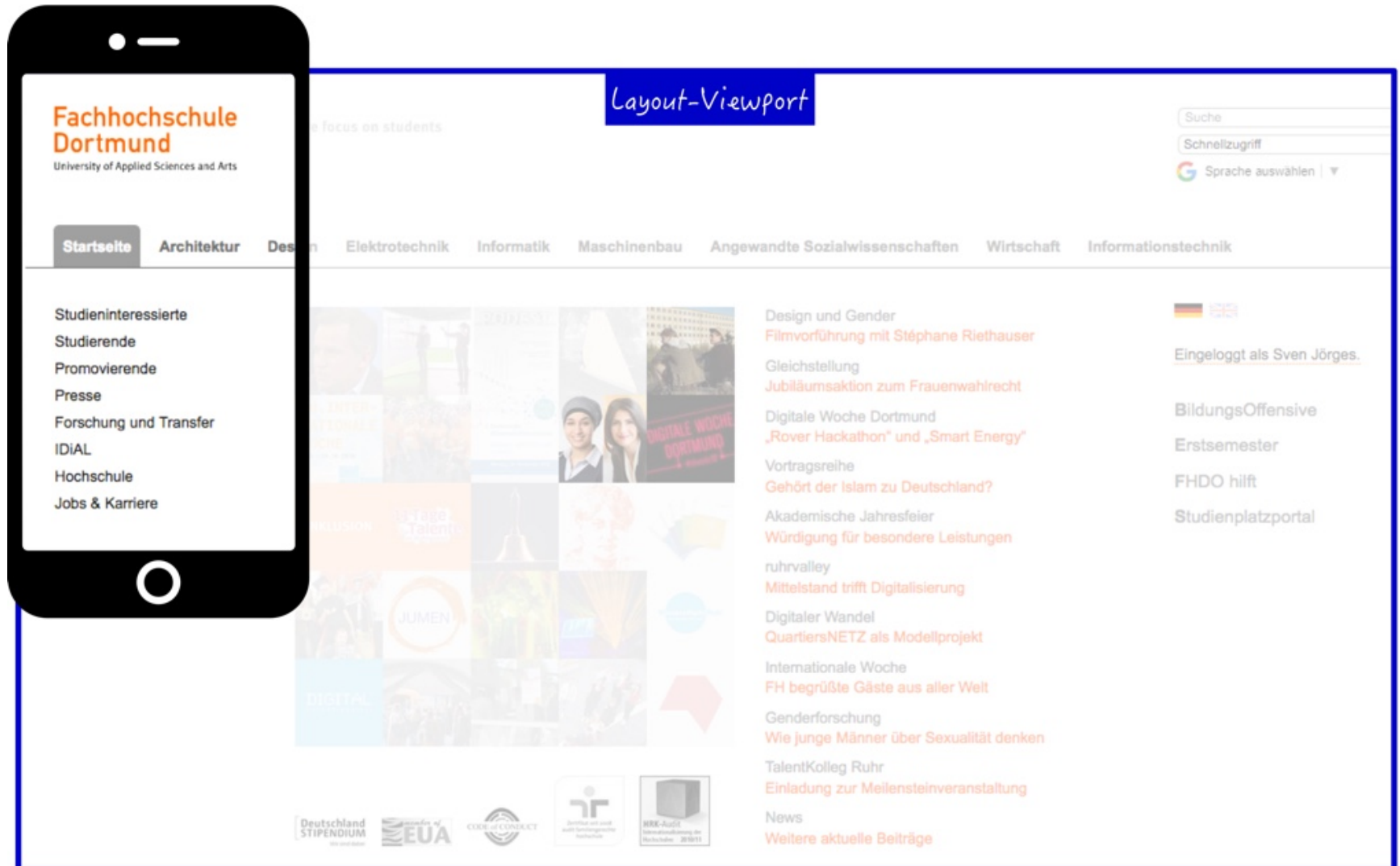
2. Visueller Viewport

- Tatsächlich sichtbarer Bereich auf dem Endgerät - typischerweise ein Ausschnitt des Layout-Viewports
- Der sichtbare Ausschnitt des Layout-Viewports kann z.B. durch Zoomen vergrößert oder verkleinert werden
- Ein gewünschter Bereich des Layout-Viewports kann in den visuellen Viewport "gescrolled" werden

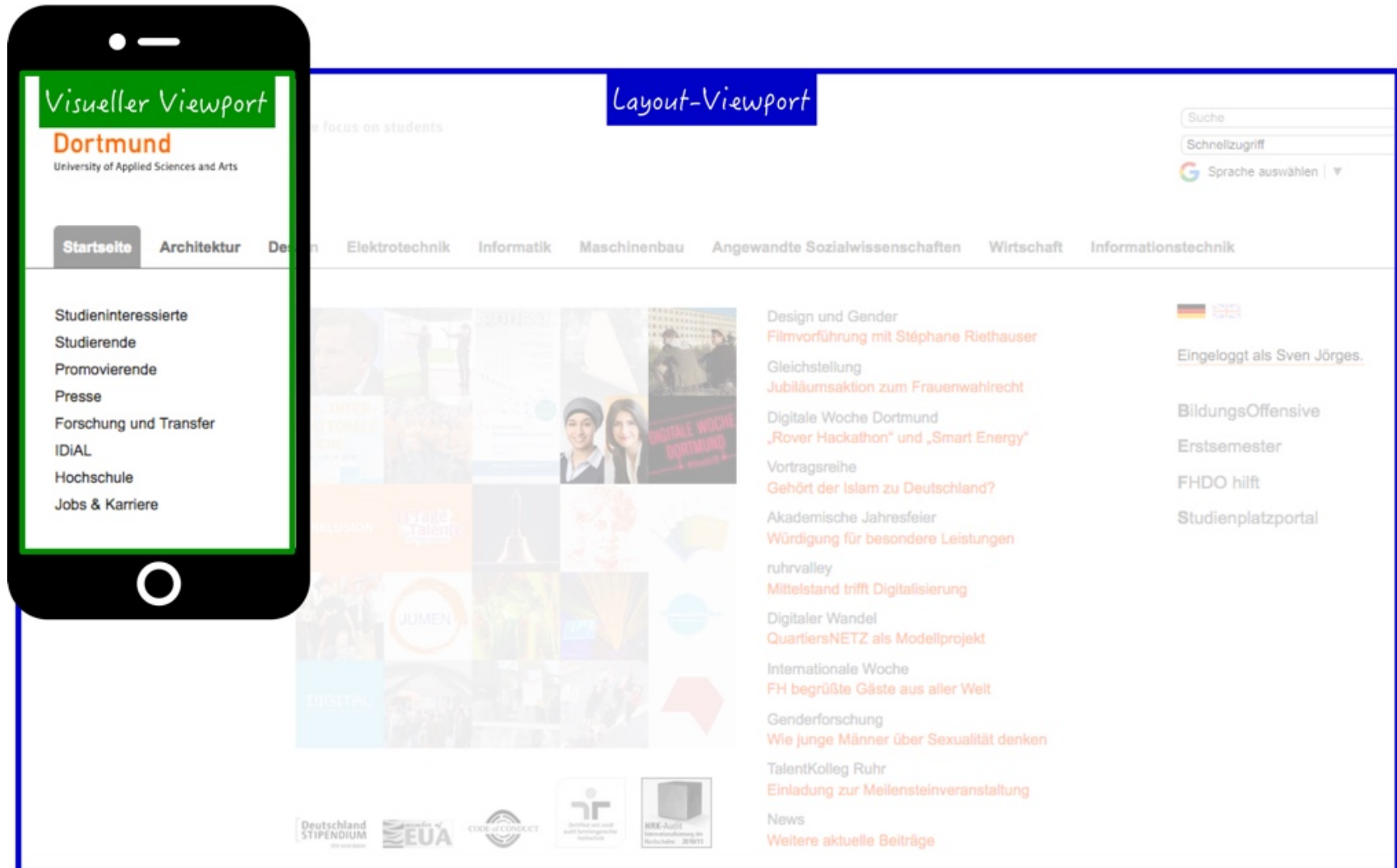
LAYOUT-VIEWPORT UND VISUELLER VIEWPORT



LAYOUT-VIEWPORT UND VISUELLER VIEWPORT



LAYOUT-VIEWPORT UND VISUELLER VIEWPORT



VIEWPORT UND MEDIA QUERIES

- Bei nicht für kleine Bildschirme optimierten Webseiten gehen mobile Browser oft so vor:
 1. Rendern der Webseite auf dem Layout-Viewport (z.B. 980px)
 2. Verkleinern (Rauszoomen) der Ansicht auf die Größe des visuellen Viewports
- Für optimierte Webseiten, die z.B. Media Queries einsetzen, ist dies kein passendes Vorgehen
- *Grund?* Media Queries für kleinere Viewport-Breiten (z.B. $< 980\text{px}$) würden nicht anschlagen!

VIEWPORT-METATAG

- Mit Hilfe des **Viewport-Metatags** kann u.A. die Größe des Layout-Viewports vorgegeben werden:

```
<meta name="viewport" content="width=320">
```

- Zudem sind weitere Einstellungen möglich, z.B.:
 - `initial-scale`: anfänglicher Zoomfaktor
 - `user-scalable`: legt fest, ob der Benutzer zoomen darf oder nicht
- ⚠ Das Viewport-Metatag ist aktuell (noch) nicht standardisiert, funktioniert aber dennoch in den meisten Browsern

VIEWPORT-METATAG (2)

Responsive Webseiten sollten typischerweise ein Viewport-Metatag wie folgt enthalten:

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

- `width=device-width` setzt die Breite des Layout-Viewports auf die Breite des visuellen Viewports des Endgerätes
- `initial-scale=1`: setzt den anfänglichen Zoomfaktor auf 1:1

ARTEN VON LAYOUTS

Wir betrachten im Folgenden verschiedene Typen von Layouts und wie man diese umsetzen kann:

1. Fixes Layout
2. Fluides Layout
3. Adaptives Layout
4. Responsives Layout

RESPONSIVES LAYOUT

- Responsiv \cong reaktionsfähig
- Mischung aus fix, fluid und adaptiv:
 1. Bei Größenänderungen passt sich das Layout fluid an (relative Größenangaben)
 2. Zusätzlich existieren Breakpoints, an denen das Layout sich adaptiert (z.B. Weglassen oder Umpositionierung von Elementen)
 3. Oft wird zur besseren Lesbarkeit noch eine Maximalgröße vorgesehen, ab welcher das Layout fix bleibt (d.h. es entstehen Ränder)

RESPONSIVES LAYOUT (2)

- ➕ Optimale Anpassung an verschiedene Display- und Viewportgrößen
- ➕ Höhere Flexibilität, da keine Beschränkung auf bestimmte Endgeräte
- ➖ Designaufwand kann höher sein: Flexibilität muss berücksichtigt werden, mehrere Layouts müssen erstellt werden
- ➖ Ggf. höherer Entwicklungsaufwand, da auch Gestaltungselemente wie z.B. Bilder flexibel sein müssen

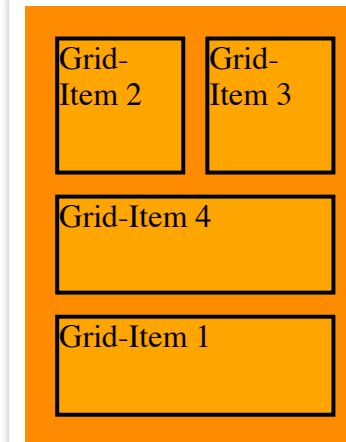
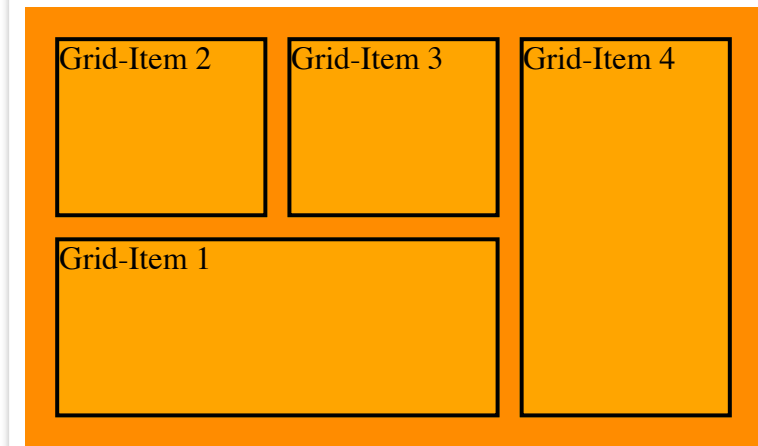
BEISPIEL: RESPONSIVES GRID

style.css

```
#container {  
  display: grid;  
  grid-template-areas: "area2 area3 area4"  
                      "area1 area1 area4";  
  /* background-color, padding, margin, height,... */  
  [...]  
}  
  
@media screen and (max-width:480px) {  
  #container {  
    grid-template-areas: "area2 area3"  
                        "area4 area4"  
                        "area1 area1";  
  }  
}  
  
#container > div { /* border, background-color, ... */ [...] }  
  
#item1 { grid-area: area1; }  
#item2 { grid-area: area2; }  
#item3 { grid-area: area3; }  
#item4 { grid-area: area4; }
```

seite.html

```
<!DOCTYPE html>  
<html>  
  [...]  
  
  <body>  
    <div id="container">  
      <div id="item1">Grid-Item 1</div>  
      <div id="item2">Grid-Item 2</div>  
      <div id="item3">Grid-Item 3</div>  
      <div id="item4">Grid-Item 4</div>  
    </div>  
  </body>  
</html>
```



RESPONSIVE BILDER

- ➖ Nachteil der CSS-Lösung für flexible Bilder: Es wird weiterhin das vollständige Bild in Originalgröße geladen → bei mobilen oder schlechten Datenverbindungen nicht optimal!
- 💡 Idee "Breakpoints für Bilder":
 - Bilder in mehreren Varianten vorhalten (z.B. verschiedene Größen, Auflösungen, Dateiformate)
 - Über Media Queries jeweils das passende Bild laden

RESPONSIVE BILDER (2)

Das `picture`-Element von HTML:

```
<picture>
  <source media="(min-width: 1024px)" srcset="assets/picture.jpg">
  <source media="(min-width: 640px)" srcset="assets/picture-medium.jpg">
  <source media="(min-width: 480px)" srcset="assets/picture-small.jpg">
  <!-- Fallback per img-Element (z.B. bei älterem Browser) -->
  
</picture>
```

- Über das `source`-Element werden verschiedene Varianten des Bildes definiert
- Attribut `srcset` definiert die URL zum Bild
- Attribut `media` enthält eine Media Query
- Browser geht die `source`-Elemente von oben nach unten durch und lädt die erste Variante mit zutreffender Media Query
- Als Fallback dient ein normales `img`-Element

FAZIT

Wir haben nun einen guten "Werkzeugkoffer" zur Realisierung verschiedenster Layouts:

Anforderung	Techniken
Flexible Inhalte (Fluide Layouts, Responsive Layouts)	<ul style="list-style-type: none">• Relative Größenangaben (z.B. %, em)• Flexbox• Grid• Flexible Bilder
Adaptivität (Adaptive Layouts, Responsive Layouts)	<ul style="list-style-type: none">• Media Queries• picture-Element von HTML

ANIMATIONEN DER VERSCHIEDENEN LAYOUTTYPEN:

[Artikel von UX Alpaca auf medium.com](#) 