

Responsive Web Design - eine App für alle Devices

Aktuelle Technologien zur Entwicklung verteilter Java-Anwendungen

Sascha Siemens

09. Juni 2017

Agenda

- Bedeutung des responsive Web-Design
- Charakteristika eines responsive Webauftritts
- Technische Umsetzung
 - Voraussetzungen
 - Die Viewport-Einstellung
 - Einsatz von relativen Größenangaben
 - Verwendung von „Media Queries“
- Einführung in das responsive Web-Framework „Bootstrap“
 - Was ist „Bootstrap“?
 - Einführung in die Container-Klassen
 - Das Bootstrap Grid-System
 - Navigationsleisten ansprechend gestalten
 - Überblick über das Framework
- Fazit und Ausblick



Bedeutung des responsive Web-Designs



Geschichtliche Entwicklung

- Anfang 2000: Anstieg der Entwicklung von plattformübergreifenden Web-Applikationen für Desktop-Systeme
- Integration von komplexer Geschäftslogik in Web-Anwendungen
- Zunahme der Bedeutung von Web-Anwendungen

- 2009: Beginn des Smartphone- und Tablet-Booms

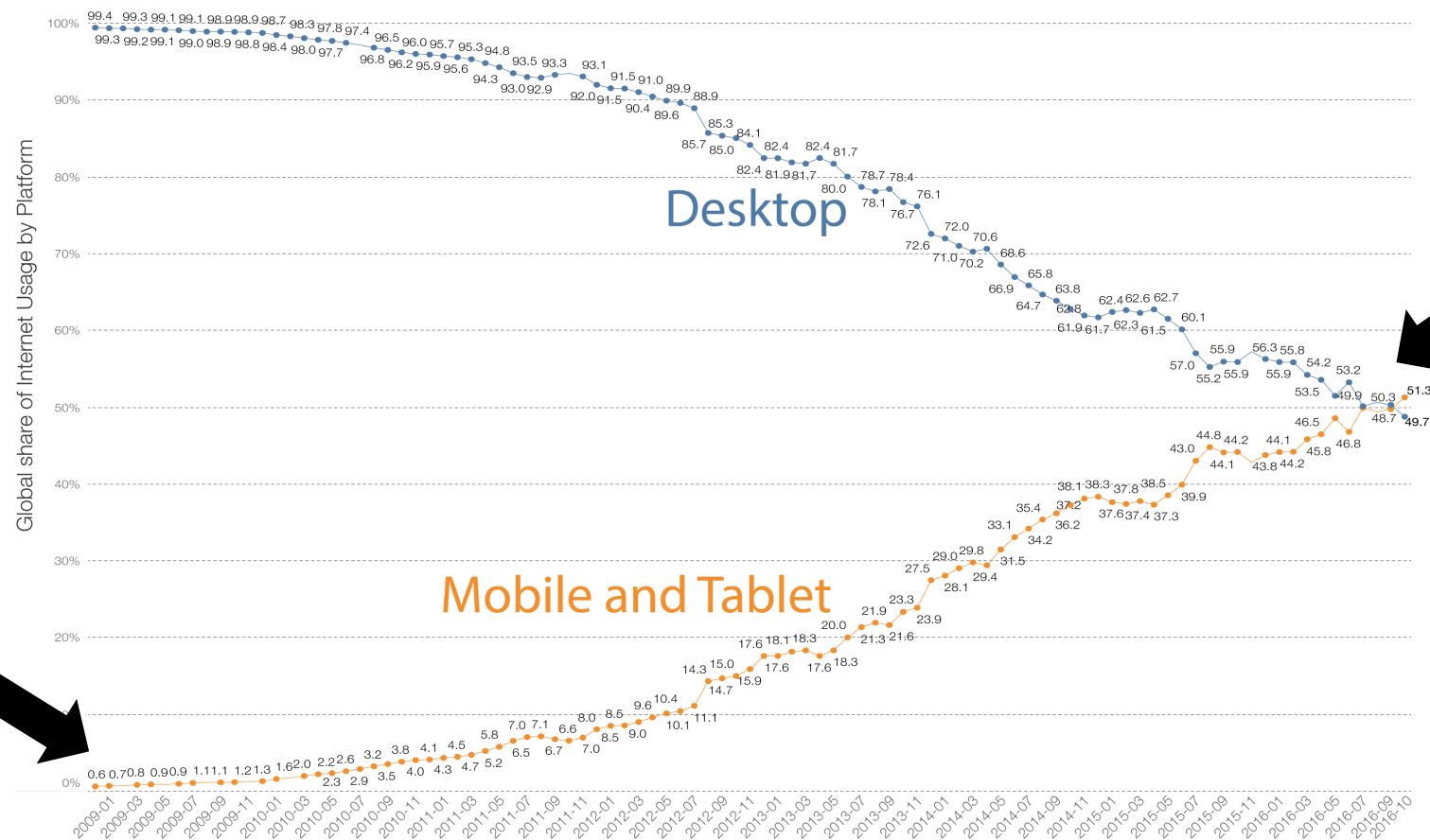
→ Problem: Bestehende Webanwendungen sind nicht für die Darstellung auf mobilen Geräten optimiert



Entwicklung der Gerätenutzung

Global share of internet usage by platform worldwide (2009 to October 2016)

These estimates are published by StatCounter.com



2009

2016

Data source: StatCounter (accessed 1 November 2016)

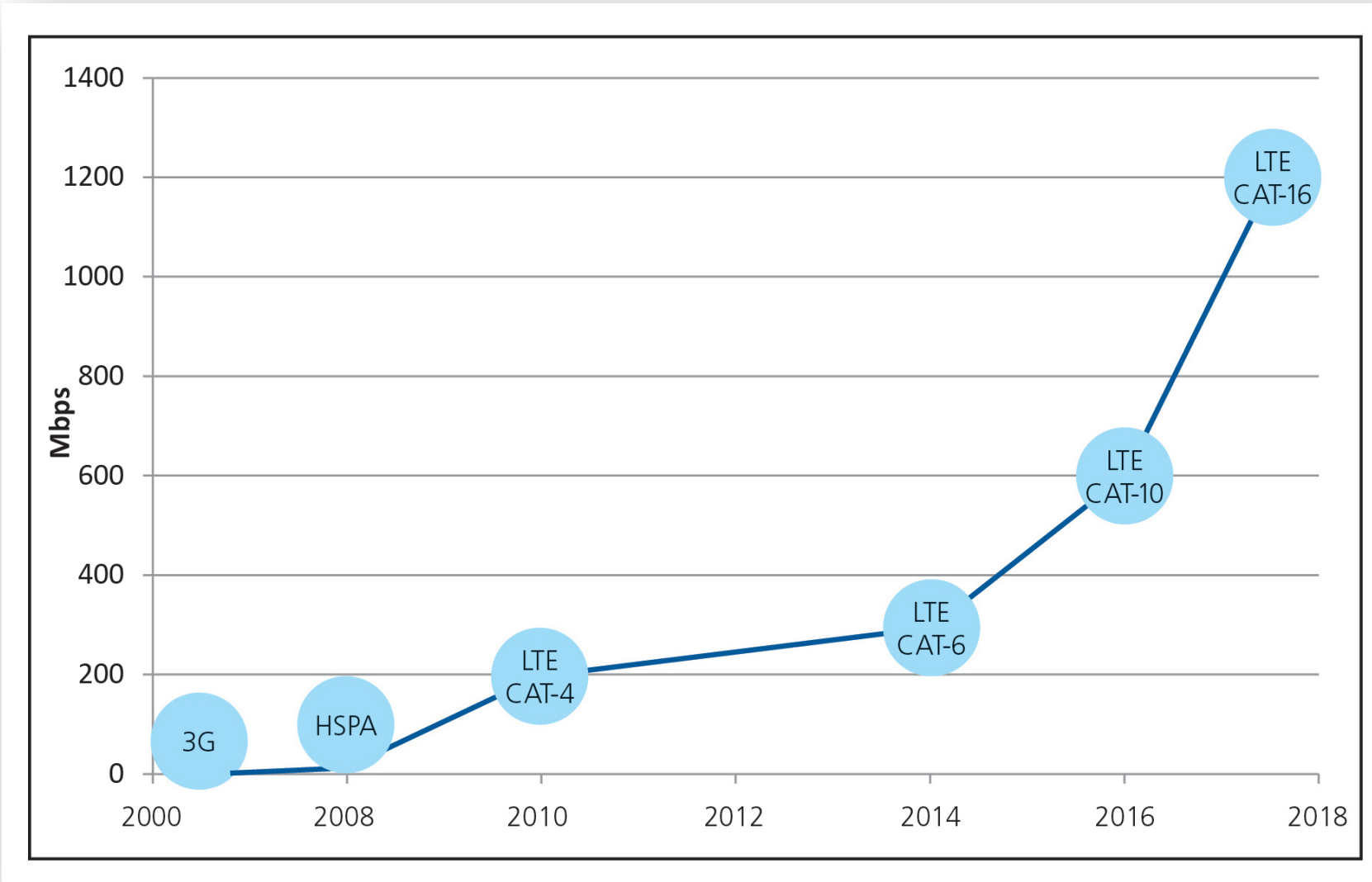
The interactive data visualization is available at OurWorldinData.org. There you find the raw data and more visualizations on this topic.

Licensed under [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) by the author Max Roser.

Quelle: https://ourworldindata.org/wp-content/uploads/2015/07/Desktop-vs-tablet-and-phone.twb_.png



Entwicklung der mobilen Funknetz-Standards



Quelle: <http://www.all-electronics.de/wp-content/uploads/2016/06/UBA015Fig2.jpg>



Entwicklung der LTE-Netzabdeckung

Entwicklung der LTE-Verfügbarkeit



Der Graph zur Entwicklung der LTE-Verfügbarkeit vermittelt Ihnen einen Eindruck von der monatlich ermittelten LTE-Netzabdeckung in Deutschland. Schwankungen nach oben und unten sind auf Grund der monatlich veränderten Stichprobe immer möglich, eine allgemeine Tendenz nach oben ist zu erwarten.

Zuletzt aktualisiert am 01.06.2017 00:02:28 aus 842752 Abfragen.

Quelle: <https://www.4g.de/service/statistiken/>



Bedeutung für die Web-Entwicklung

- Smartphones und Tablets sind mit Desktop-Systemen gleichgestellt bzw. haben diese bereits überholt
- Die Entwicklung für mobile Geräte darf nicht vernachlässigt werden und stellt einen ernst zu nehmenden Punkt sowohl für den privaten als auch den geschäftlichen Bereich dar
- Bestandssoftware muss hinsichtlich der Bedienung auf Geräten mit kleinen Bildschirmen geprüft und angepasst werden, um weiterhin am Markt bestehen zu können



Responsive Web Design als Akzeptanzkriterium

- Bei der Auswahl einer Web-Applikation, wird bereits bei der Anforderungsanalyse auf ein responsive Web Design geachtet, selbst wenn mobile Geräte noch nicht im Einsatz sind → Investitionsschutz
- Benutzung muss unabhängig vom Gerät einfach und intuitiv sein → Trendentwicklung: „Bring your own device“ / Entkoppelung Hard- von Software
- “Look and feel“ muss für Endanwender/ Mitarbeiter angenehm und ansprechend sein → Gewinnung der Akzeptanz der Benutzer unabhängig vom Funktionsspektrum



Darstellungsprobleme auf mobilen Geräten (1)



- Gesamte Webseite wird dargestellt, aber:
 - Text unleserlich
 - Unübersichtlicher Webauftritt
 - Benutzer müssen zoomen
 - Hyperlinks können nicht angesprochen werden
 - Navigationsleiste beansprucht viel Platz

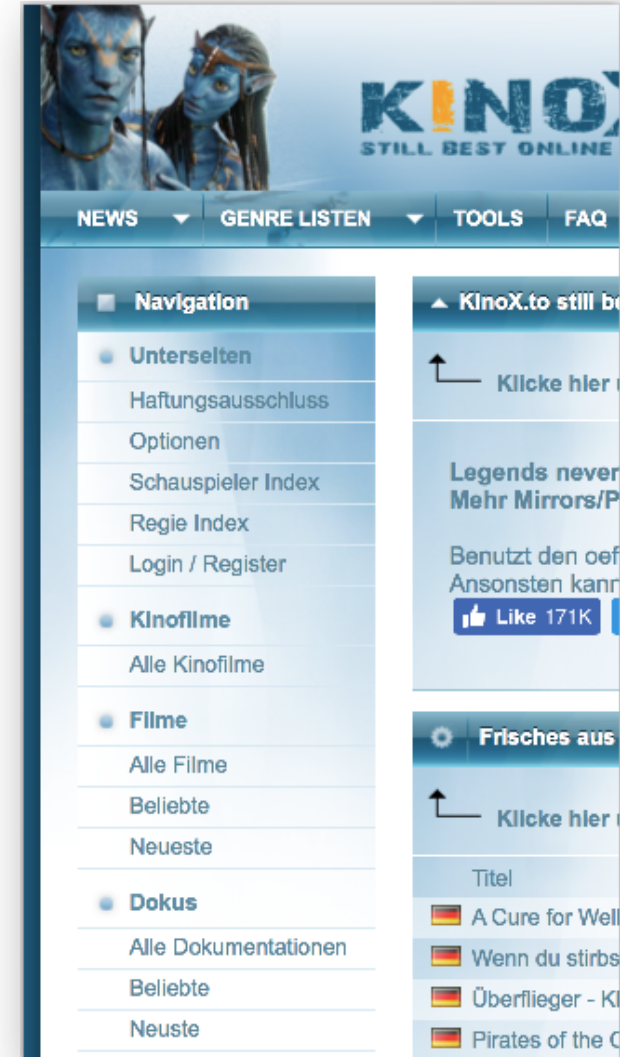
Quelle: https://upload.wikimedia.org/wikipedia/en/b/b4/Internet_Explorer_Mobile_9.png



Darstellungsprobleme auf mobilen Geräten (2)

Text und Hyperlinks lesbar, aber:

- Gesamte Website nicht sichtbar
- Unübersichtlicher Webauftritt
- Benutzer müssen zoomen
- Ansicht muss nach jedem Seitenaufbau neu angepasst werden
- Benutzer sind gewohnt Webseiten vertikal zu bewegen. (mittels Mausekranz)
- Die horizontale Navigation der Website gilt unter Webentwicklern als schlechter Stil
- Navigationsleiste beansprucht viel Platz vom sichtbaren Inhalt



Erste Lösungsansätze / Workarounds

Entwickeln einer Desktop- & Mobilversion:

Vorteil:

Ansprechende Webseite für Smartphones sowie Desktop-Systeme

Nachteil:

Enormer Entwicklungsaufwand, da Änderungen an zwei Stellen gepflegt werden müssen

Erstellung einer mobilen Version, welche auch für Desktop-Systeme genutzt wird:

Vorteil: Entwicklung nur für mobile Version nötig

Nachteil: Bedienung auf Geräten mit großer Auflösung nicht ansehnlich

→ Keine effiziente sowie tragbare Lösung!!!

Lösung: Ansatz des responsive Web-Designs



Charakteristika eines responsive Webauftritts



Definition: Responsive Web-Design

- Ansprechendes (engl. „responsive“) Web-Design, unabhängig von
 - Displaygröße
 - Gerät
 - Betriebssystem
 - Web-Browser
 - Ausrichtung (Quer- oder Hochformat)
- Entwicklung eines Web-Frontends für alle Geräte
- Optimale Darstellung auf allen Geräten
 - Desktop-Systemen
 - Tablets
 - Smartphones



Charakteristika eines ansprechenden Webauftritts

- Logos, Banner oder Header minimiert bzw. minimalisiert anzeigen
- Suchleisten, Login-Formulare oder andere statische Elemente durch Symbole ersetzen
- Navigationsteil der Webseite durch aufklappbare Elemente ersetzen
- Beiträge oder Inhalte, welche auf der Desktop-Seite nebeneinander dargestellt werden, sollten untereinander angeordnet werden
- Meta-Informationen ausblenden (z.B. Veröffentlichungsdatum von Beiträgen oder Namen von Autoren)
- Beiläufige Angaben sollten ausgeblendet oder können durch dynamische Elemente aufgeklappt werden (Impressum, Copyright)



Responsive Webauftritt der Hochschule München

HOME HM | UNTERNEHMEN | PRESSE | ALUMNI | INTRANET | IMPRESSUM

ENGLISH VERSION

TECHNIK > WIRTSCHAFT > SOZIALES > DESIGN >

Header max.

NAVIGATION

- ICH BIN STUDIERENDE/R
- ICH BIN STUDIENINTERESSENT/IN

- > Die Hochschule München
- > Studienangebote
- > International
- > Lehre
- > Forschung & Entwicklung
- > Weiterbildung
- > Job & Karriere
- > Kontakt
- > Sitemap

Datenschutz & Webtracking

UNTERNEHMERISCH

Unternehmerisch denken und handeln
... an der EXIST-Gründerhochschule und am SCE

VERÖFFENTLICHUNGEN

Lesestoff von und zur Hochschule
Publikationen, Magazine und Broschüren der Hochschule München

DIGITALISIERUNG

Hochschule im digitalen Zeitalter
Die HM auf dem Weg in die digitale Zukunft - gehen Sie mit!

AKTUELLES

News >

Veranstaltungen >

Veröffentlichungen >

NEWS

[09|05|2017]

Lehrraum der Zukunft
Die "Kapelle" am Campus Pasing wurde als einer von drei Räumen an der HM für innovative Lehrformen ausgestattet

Mehr >

VERANSTALTUNGEN

15.05.2017
Digitalisierung Ringvorlesung
Robotics & beyond: globaler Fortschritt, Wachstum und Arbeit in Zeiten der Digitalisierung

16.05.2017
Dialogforum 2017: People on the move
Flüchtlingsengagement: Hilfe, die ankommt? Diskutieren Sie mit am 16. Mai

17.05.2017
Wege zur Promotion mit HAW-Abschluss
Informieren Sie sich über die Möglichkeiten und Unterstützungen einer Promotion

Desktop-Version

UNTERNEHMERISCH

Unternehmerisch denken und handeln
... an der EXIST-Gründerhochschule und am SCE

VERÖFFENTLICHUNGEN

Lesestoff von und zur Hochschule
Publikationen, Magazine und Broschüren der Hochschule München

DIGITALISIERUNG

Hochschule im digitalen Zeitalter
Die HM auf dem Weg in die digitale Zukunft - gehen Sie mit!

NEWS

15.05.2017
Digitalisierung Ringvorlesung
Robotics & beyond: globaler Fortschritt, Wachstum und Arbeit in Zeiten der Digitalisierung

16.05.2017
Dialogforum 2017: People on the move
Flüchtlingsengagement: Hilfe, die ankommt? Diskutieren Sie mit am 16. Mai

17.05.2017
Wege zur Promotion mit HAW-Abschluss
Informieren Sie sich über die Möglichkeiten und Unterstützungen einer Promotion

Mobile-Version



Demo 1

[Webauftritt der Hochschule München](#)



Technische Umsetzung

Hinweis: Grundlegende Kenntnisse in HTML und CSS werden vorausgesetzt!



Technische Voraussetzungen

- HTML 5 und CSS 3
- Konkret:
 - Die Viewport-Einstellung
 - Relative Größenangaben
 - „Media-Queries“
- JavaScript ist keine Voraussetzung für ein responsive Web-Design, allerdings erfreut sich JavaScript zunehmender Beliebtheit um Webseiten dynamischer und benutzerfreundlicher zu gestalten



Die Viewport-Einstellung

- Jedes Gerät hat eine vom Bildschirm abhängige Auflösung
- Beim Aufruf einer Webseite wird der Inhalt der Webseite anhand der Bildschirmauflösung angezeigt.
- Ist die Breite des Webauftritts größer als die Breite der Bildschirmauflösung, so wird nur ein Teilbereich der Webseite angezeigt → Zoomen notwendig
- Abhilfe schafft das Viewport-Tag im HTML-Header indem eine „virtuelle“ Auflösung angegeben wird, die das Endgerät anstelle der eigenen Auflösung darstellen/annehmen soll.



Die Viewport-Einstellung

Hat ein Webauftritt z. B. eine feste Auflösungsbreite von 1024 Pixel, so kann mittels folgendem Viewport-Tag im HTML-Header eine virtuelle Auflösung angegeben werden:

```
<html>
<head>
<meta name="viewport" content="width=1024">
</head>
<body>
  Content of Website with a width of 1024 px!
  ...
</body>
</html>
```

Das Gerät zeigt trotz niedrigeren Auflösung eine virtuelle Auflösung der gesamten Breite der Webseite an.

Vorteil: Kein horizontales scrollen nötig

Nachteil: Text eventuell nicht mehr lesbar, da Darstellung zu klein

Hinweis: Nicht alle (mobilen) Webbrowser folgen dieser Konvention! Ist kein Viewport-Tag angegeben, so wird vom Geräte-Hersteller/Web-Browser ein Standardwert für das fehlende Viewport-Tag angenommen.



Die Viewport-Einstellung im responsive Kontext

- Mit Bezug auf das responsive Web-Design hat sich folgendes Viewport-Tag etabliert:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- **width=device-width** entspricht dabei der Auflösung der tatsächlichen Bildschirmbreite des jeweiligen Geräts und spielt im Zusammenhang mit den „Media Queries“ eine wichtige Rolle (Erklärung folgt)
- **initial-scale=1.0** entspricht dabei den standardmäßigen Zoom-Faktor der Webseite. (1.0 entspricht keinem Zoomverhalten)

Hinweis: Diese Konvention des Viewport-Tag gilt als Voraussetzung für alle responsive Web-Designs



Weitere Viewport Konfigurationmöglichkeiten

- Als „content“-Wert können weitere Eigenschaften verwendet werden:

Eigenschaft	Beschreibung
height	Zeigt virtuellen Viewport einer speziellen Höhe an
device-height	Zeigt die Größe des tatsächlichen physikalischen Geräts an
minimum-scale	Minimaler Zoomfaktor des Benutzers
maximum-scale	Maximaler Zoomfaktor des Benutzers
user-scalable	Erlaubt den Benutzer zu zoomen (yes / no)



Relative Größenangaben



Relative Größenangaben

- Feste Größenangaben erlauben keine dynamische Anpassung an die jeweilige Bildschirmgröße
- Relative Größenangaben sind immer vorzuziehen
- Dies gilt insbesondere für
 - <div>-Container
 - Tabellen
 - Bilder
 - Schriftgrößen
 - Abstandsangaben (Paddings & Margins)
- Ausnahmen: Sehr kleine Elemente (z. B.: dünne Rahmenlinie)



Relative Größenangaben bei Schriften

- Für Schriftgrößen wird empfohlen eine Referenzgröße für die <html>-Wurzel zu definieren (z. B. **font-size: 16px;**)
- Alle weiteren Schriftgrößen innerhalb des HTML-Dokuments sollen mittels der „**em**“-Größenangabe anstelle mit „**px**“ definiert werden, um eine dynamische Abhängigkeit von der Referenzschriftgröße zu erhalten
- Beispiel CSS-Datei:

```
html {  
  font-size: 16px; /* Default font size */  
}
```

```
div.double-size {  
  font-size: 2em; /* Double font size */  
}
```

- Da geschachtelte Elemente die jeweiligen Schriftgrößen vererben, kann mittels dem „**em**“-Wert auf die Standardschriftgröße zugegriffen werden.



Relative Größenangaben

- Für
 - <div>-Container
 - Tabellen
 - Bilder
 - Etc.
- wird empfohlen, prozentuale Angaben für die Höhe/Breite anstelle von festen Pixel-Werten zu verwenden, um eine automatische Anpassung an die Bildschirmgröße zu erhalten.
- Beispiel anhand eines Bildes:
``



Demo 2

Beispiel für Viewport-Tag und relativen Größenangaben



Einsatz von Media Queries

- Zentrale Komponente des responsive Web-Designs
- Verwendung verschiedener CSS-Templates anhand verwendeter Geräte-Auflösung und -Ausrichtung
- Verwendung folgender CSS-Syntax:
**@media not | only *mediatype* and (*media feature*) {
 CSS-Code;
}**
- Beispiel: Grüne Hintergrundfarbe, wenn Auflösungsbreite größer als 600 px

```
@media all and (min-width: 600px) {  
    body {  
        background-color: lightgreen;  
    }  
}
```



Media Queries – Media-Types und -Features

Mediatypes	Beschreibung
all	Verwendet für alle Geräte
screen	Verwendet für Computer, Tablets und Smartphones
print	Ansicht für Druckfunktionen
speech	Verwendet für maschinelles Vorlesen des Inhalts

Gängigsten Mediafeatures	Beschreibung
min-width	Mindestbreite in Pixel des Viewports
min-height	Mindesthöhe in Pixel des Viewports
min-device-width	Mindestbreite in Pixel des Geräts
min-device-height	Mindesthöhe in Pixel des Geräts
orientation	Die Ausrichtung des Viewports (Hoch- oder Querformat)

Quelle und vollständige Liste unter: https://www.w3schools.com/cssref/css3_pr_mediaquery.asp



Demo 3

Verwendung von Media Queries



Media Queries - Fazit

- Sehr mächtiges und nützliches Toolkit
- Steigende Komplexität bei großen Webseiten mit sehr vielen Elementen und mehreren Media-Queries
- Aufwändiges Troubleshooting im Fehlerfall
- CSS-Struktur sollte vorab im Detail geplant werden

Lösung:

Einsatz eines responsive Web-Frameworks (z. B.: „Bootstrap“)



Einführung in das responsive Web-Framework „Bootstrap“



Was ist „Bootstrap“?

- Meist bekannteste responsive Web-Framework, entwickelt von Twitter im Jahr 2011
- Fokus auf Designentscheidungen
- Veröffentlicht als Open-Source-Projekt
- Basierend auf HTML5, CSS3 sowie JavaScript mit vorgefertigten Funktionsbausteinen
- 2014 beliebteste Projekt auf GitHub
- Download unter www.getbootstrap.com



Verwendung von Bootstrap - Offline Variante

Entpacktes Bootstrap-Archiv:

```
css
├── bootstrap-theme.css
├── bootstrap-theme.css.map
├── bootstrap-theme.min.css
├── bootstrap-theme.min.css.map
├── bootstrap.css
├── bootstrap.css.map
├── bootstrap.min.css
└── bootstrap.min.css.map

fonts
├── glyphicons-halflings-regular.eot
├── glyphicons-halflings-regular.svg
├── glyphicons-halflings-regular.ttf
├── glyphicons-halflings-regular.woff
└── glyphicons-halflings-regular.woff2

js
├── bootstrap.js
├── bootstrap.min.js
└── npm.js
```

Um Bootstrap verwenden zu können, muss die „bootstrap.css“ bzw. die „bootstrap.min.css“-Datei im HTML-Header als CSS-Stylesheet importiert werden:
<link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">

Die „bootstrap.min.css“-Datei enthält keine unnötigen Zeilenumbrüche sowie Leerzeichen → Schnellere Ladezeit



Verwendung von Bootstrap – Online CDN

- Import über Online CDN (Content-Delivery-Network):
Anbieter: MaxCDN
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
- Vorteil: Laden aus dem Cache, sofern andere Webseiten ebenfalls Bootstrap mit gleichem CDN nutzen → schnellere Ladezeit
- Einige Bootstrap-Funktionen benötigen eigene JavaScript-Bibliotheken, diese können ebenfalls offline oder über einen Online CDN eingebunden werden. (benötigt jQuery – Download unter www.jquery.com)

Beispiel mittels Online CDN:

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>  
<script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>
```



Bootstrap – Die Container-Klasse



Bootstrap – Die Container-Klasse

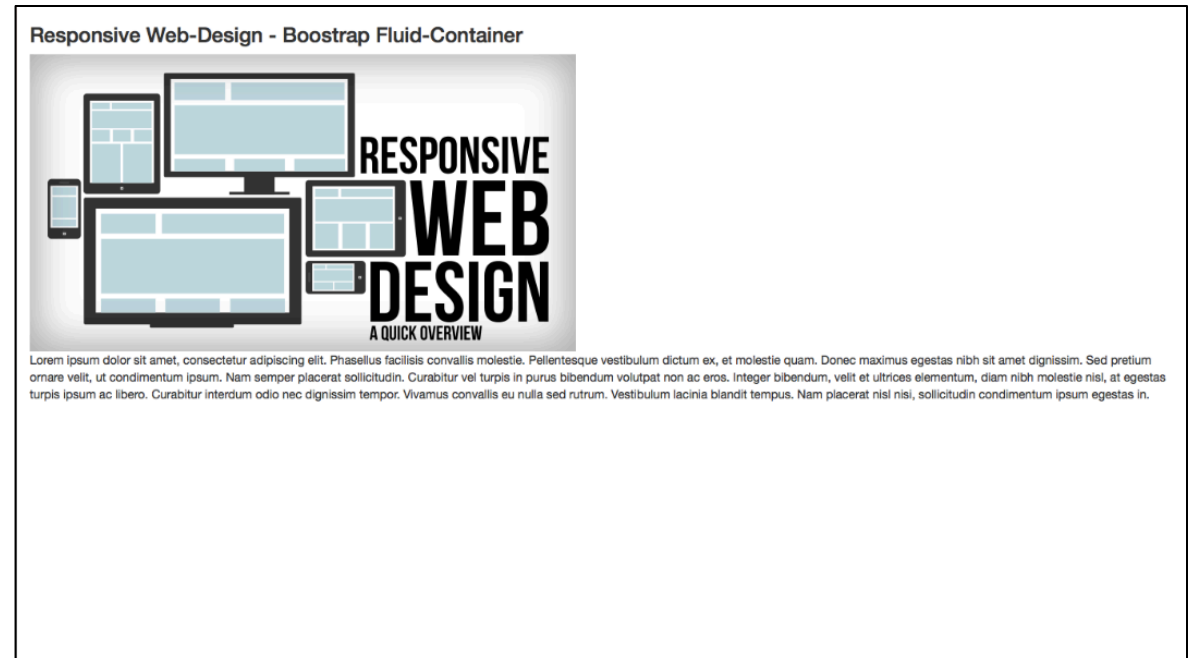
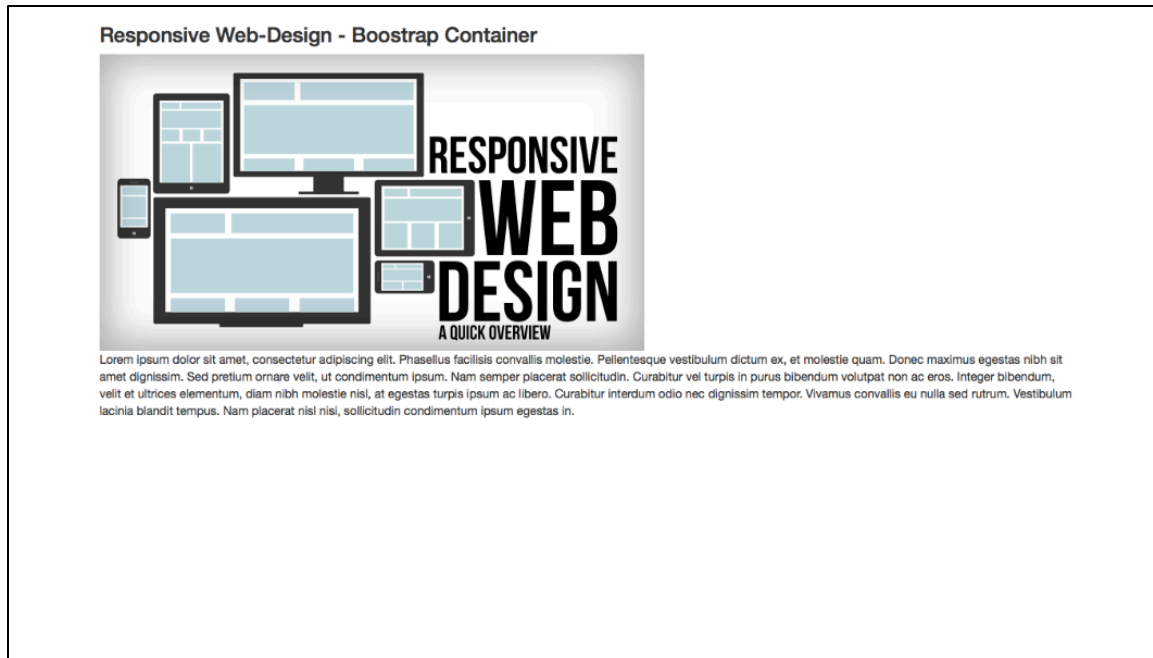
- Um Bootstrap nutzen zu können, muss der Webauftritt in einer „Div-Container“-Klasse entstehen
- Minimal-Beispiel einer Bootstrap HTML-Datei:

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">
  <meta charset="UTF-8">
  <title>Responsive Web-Design Beispiel</title></head>
<body>
  <div class="container">
    Hier entsteht der responsive Webauftritt!
  </div>
</body>
</html>
```



Bootstrap – Arten von Container-Klassen

- Bootstrap unterscheidet zwei Arten von Container-Klassen:
 - **.container** Container mit kleinem Abstand (links und rechts) zum Seitenrand
 - **.container-fluid** Container ohne Seitenabstand



Hinweis: Container-Klassen dürfen nicht geschachtelt werden



Bootstrap – Das Grid-System



Bootstrap – Das Grid-System

- Layout-Design mit 12 Spalten (Summe der Spalten-Größen muss 12 ergeben)

col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1	col-sm-1
col-sm-2		col-sm-2		col-sm-2		col-sm-2		col-sm-2		col-sm-2	
col-sm-3			col-sm-3			col-sm-3			col-sm-3		
col-sm-4				col-sm-4				col-sm-4			
col-sm-5					col-sm-2		col-sm-5				
col-sm-6						col-sm-6					
col-sm-7							col-sm-2		col-sm-3		
col-sm-8								col-sm-2		col-sm-2	
col-sm-9									col-sm-3		
col-sm-10										col-sm-2	
col-sm-11											col-sm-1
col-sm-12											



Bootstrap – Grid-Klassen

Bootstrap definiert vier Grid-Klassen: **xs**, **sm**, **md** und **lg**

	Sehr kleine Auflösung (Smartphones)	Kleine Auflösung (Tablets)	Mittlere Auflösung (Desktop-Systeme)	Sehr große Auflösung (Desktop-Systeme)
Klassenprefix	col-xs- <i>{Spaltenbreite}</i>	col-sm- <i>{Spaltenbreite}</i>	col-md- <i>{Spaltenbreite}</i>	col-lg- <i>{Spaltenbreite}</i>
Auflösungs- Schwellwerte (horizontal)	< 768 px	>= 768 px	>= 992 px	>= 1200 px
Spalten- Anordnung	Spalten werden immer nebeneinander und niemals untereinander angezeigt	Unterhalb der Auflösung werden die Spalten untereinander angeordnet, sonst nebeneinander		



Bootstrap – Beispiel Grid-System

- Grid-Objekte entstehen innerhalb einem `<div class="row"></div>` Element
- Spalten werden innerhalb eines Row-Elements definiert:
`<div class="col-[xs, sm, md, lg]-[1-12]">Content</div>`

- **Beispiel:**

```
<div class="container">
  <div class="row">
    <div class="col-xs-4">Dieser Text steht immer nebeneinander.</div>
    <div class="col-xs-4">Dieser Text steht immer nebeneinander.</div>
    <div class="col-xs-4">Dieser Text steht immer nebeneinander.</div>
  </div>
  <div class="row">
    <div class="col-md-6">Wenn Displaybreite >= 992 px nebeneinander, ansonsten untereinander</div>
    <div class="col-md-6"> Wenn Displaybreite >= 992 px nebeneinander, ansonsten untereinander </div>
  </div>
</div>
```



Demo 4

Verwendung des Bootstrap Grid-Systems



Bootstrap – Navigationsleisten & Überblick über das Framework



Bootstrap - Navigationsleisten

- Navigationsleisten können in Bootstrap mittels der Klasse „**navbar**“ erstellt werden
- Beispiel:

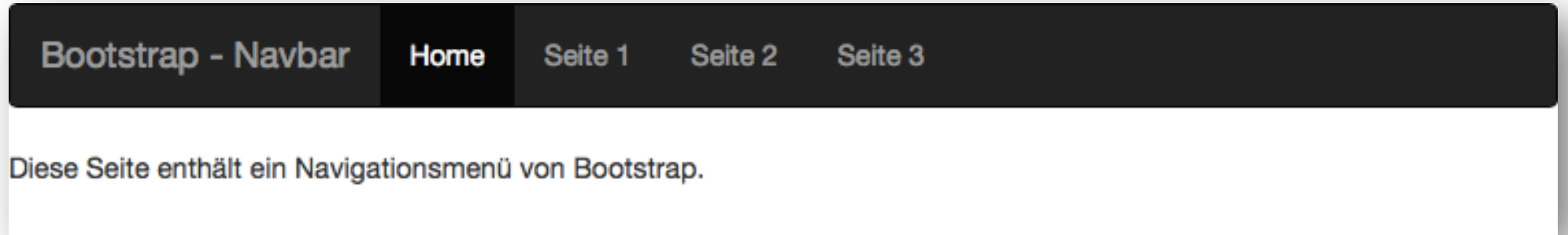
```
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#Navigationsleiste">
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
        <span class="icon-bar"></span>
      </button>
      <a class="navbar-brand" href="index.html">Bootstrap - Navbar</a>
    </div>
    <div class="collapse navbar-collapse" id="Navigationsleiste">
      <ul class="nav navbar-nav">
        <li class="active"><a href="#">Home</a></li>
        <li><a href="#">Seite 1</a></li>
        <li><a href="#">Seite 2</a></li>
        <li><a href="#">Seite 3</a></li>
      </ul>
    </div>
  </div>
</nav>
```

- JavaScript wird für die dynamischen Navigationsleisten benötigt!

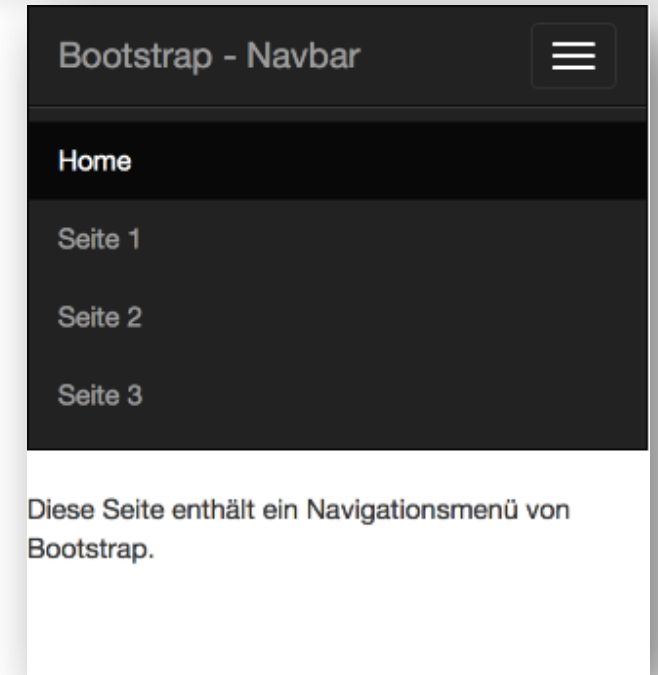
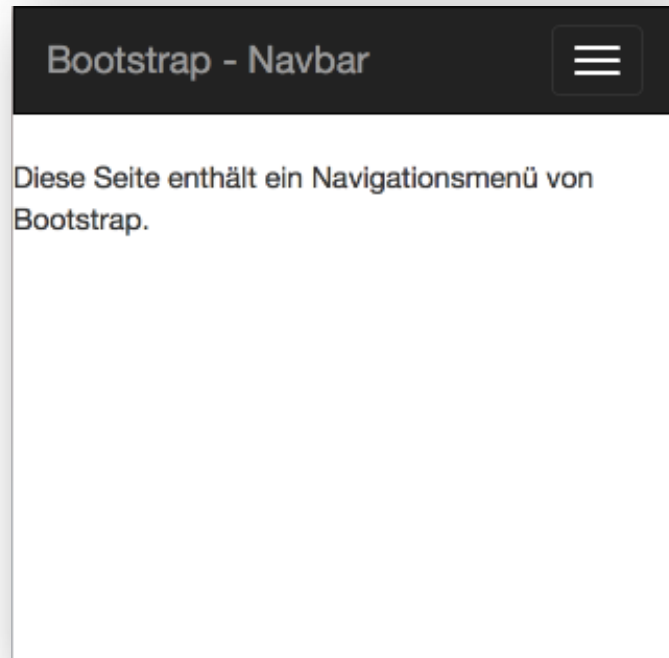


Bootstrap – Navigationsleisten Beispiel

Desktop-Version



Mobile-Version



Demo 5

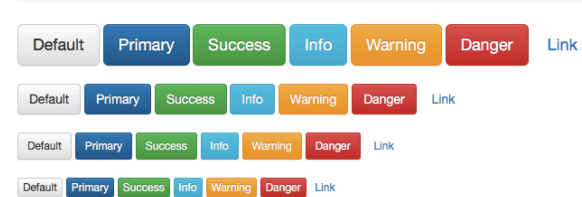
Navigationsleisten



Bootstrap – Überblick über das Framework

- Weitere Komponenten von Bootstrap:
 - Standardisierte Buttons
 - Tabellenvorlagen
 - Drop-Down-Menüs
 - Benachrichtigungen / Alerts
 - Fortschrittsbalken (Progress bars)
 - Dynamische Bilder-Karussell (Carousel)
 - Symbole / Glyphicons
 - Einheitliche Formularfelder
 - Uvm.

Buttons



Tables

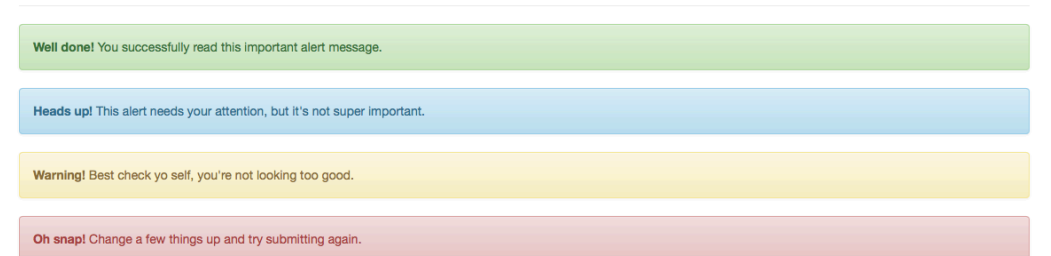
#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
	Mark	Otto	@TivBootstrap
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry the Bird		@twitter

Alerts



Demo 6

[Komponenten in Bootstrap](#)

[Beispiel Webseite mit Bootstrap](#)



Fazit

- Responsive Web-Designs sind für die Zukunft essentiell
- Der Einsatz von Bootstrap erleichtert die Umsetzung eines responsive Webauftritts
 - Einfaches und elegantes Framework
 - Leicht zu lernen
 - Fokus auf Design-Entscheidungen, anstelle von technischen Umsetzungen
 - Standardisiertes sowie getestetes Framework
 - Sehr große Community

Es wird daher empfohlen, responsive Web-Frameworks den Vorzug zu gewähren, da die Entwicklung der eigenen Webseite, schneller, einfacher und mit einem etablierten Standard gewährleistet wird



Quellen und weiterführende Informationen

- W3Schools: www.w3schools.com
- SelfHtml: www.selfhtml.org
- Offizielle Bootstrap Homepage: www.getbootstrap.com
- HTML-Validator: <https://validator.w3.org>
- CSS-Validator: <https://jigsaw.w3.org/css-validator/>
- Responsive Web Design Tester Plugin für Google Chrome:
<https://chrome.google.com/webstore/detail/responsive-web-design-tes/objclahbaimlfnbjdeobicmmlnbhamkg?hl=en>



Vielen Dank für Ihre Aufmerksamkeit!

