

Responsive Web-Design

– Webseiten-Optimierung für mobile Geräte

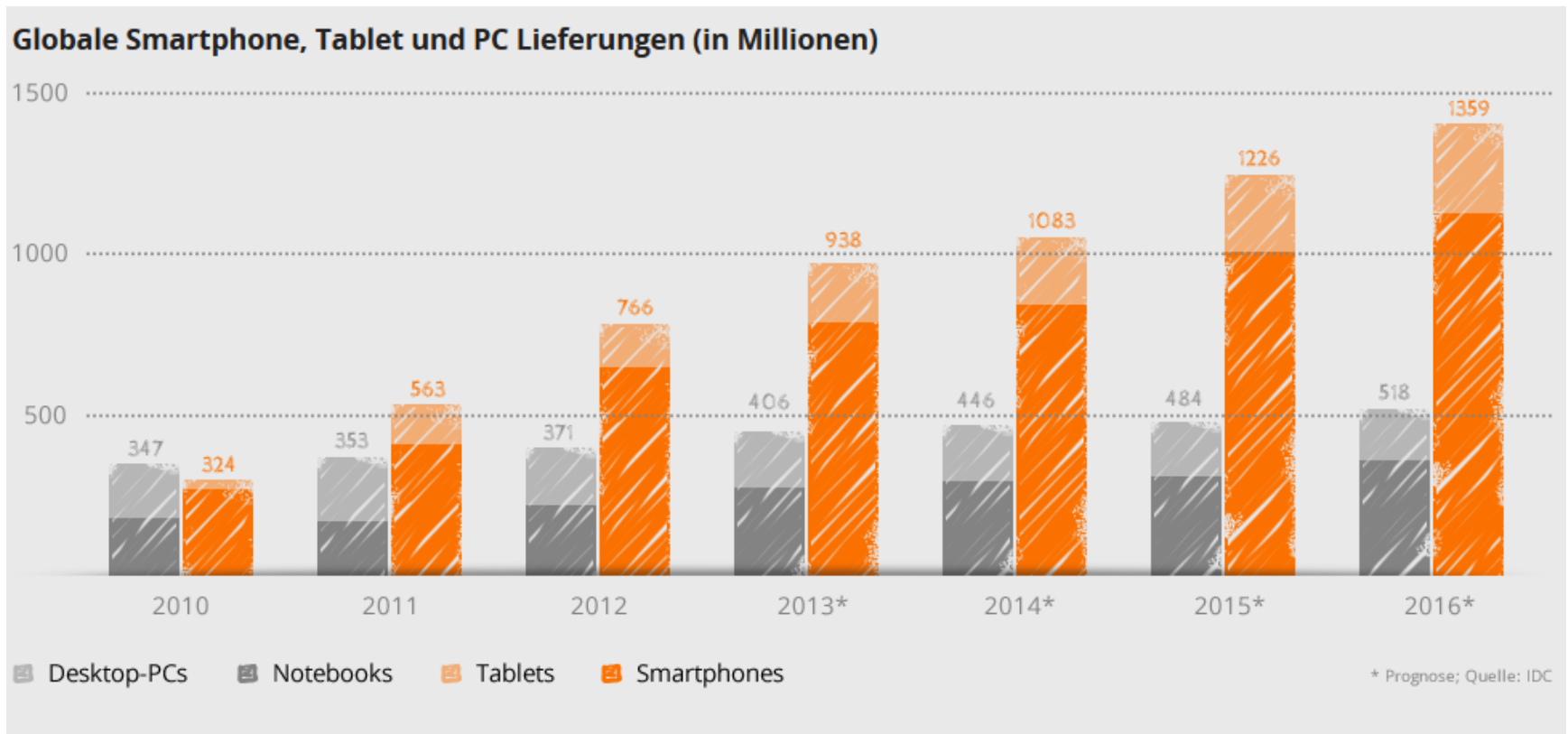
Workshop

webgrrls Workshop am 15. März 2014 in Düsseldorf

Responsive Web-Design

- Responsive = reaktionsfähig, ansprechbar
- Es gibt kein identischen Layout, wohl aber gleichbleibende Benutzerfreundlichkeit.
- Die Inhalte werden je nach Ausgabegerät neu verteilt; „form follows function“.
- CSS3-Media-Queries ermöglichen die Reaktionsfähigkeit.

Wozu responsive Web-Design?



Quelle: <http://www.responsive-webdesign.mobi/warum-responsive-webdesign/>

CSS Media Queries

- media = 1. Teil der Angabe

- Angaben für Bildschirmbreite (Viewport):
 - **Als externe Datei einbinden** (in HTML):
`<link rel="stylesheet" media="screen and (max-width: 1024px)" href="small.css">`
 - **Inline CSS nutzen** (im Stylesheet):
`@media screen and (max-width : 1024px) { hier CSS einfügen }`
 - **Ältere Browser sicher ausschließen** mit „only“:
`@media only screen and (min-width: 768px) { hier CSS einfügen }`
 - **Mehrere Angaben:**
`@media only screen and (min-width: 768px) and (max-width: 1024px) { ... }`
 - **Spezialangaben nach Bildschirm-Auflösung:**
`@media only screen and (-webkit-min-device-pixel-ratio: 1.5) // Apple ,
only screen and (min-device-pixel-ratio: 1.5) { ... }`

Übersicht: <http://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

CSS Media Queries

-meta = 2. Teil der Angabe

- Meta-Angabe im <head>-Bereich:

- Standardangabe:

- <meta name="viewport" content="width=device-width">

- Diverse Skalierungsprobleme abfangen:

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

- Für iOS noch einen drauf setzen – Zoom abschalten:

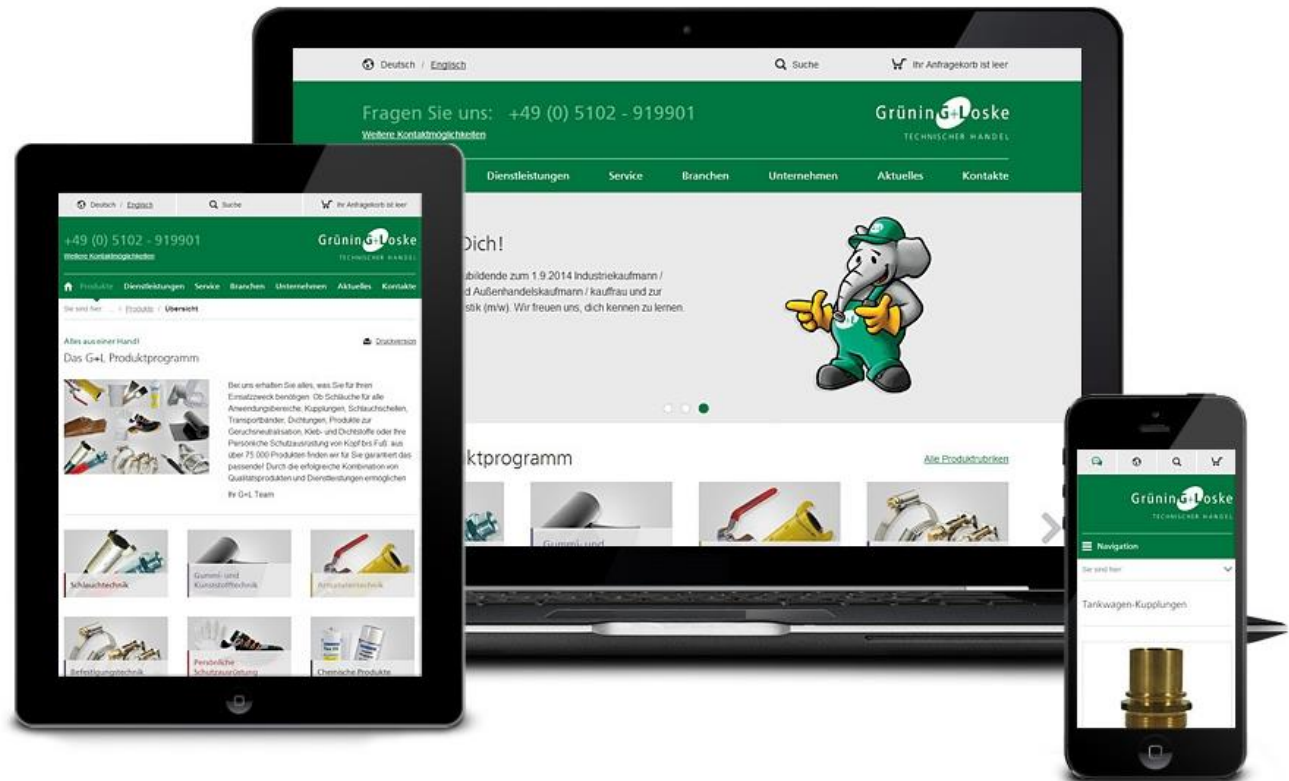
- <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=0">

- Zoom erlauben (Usability) – kann Probleme mit iPad & Co. geben:

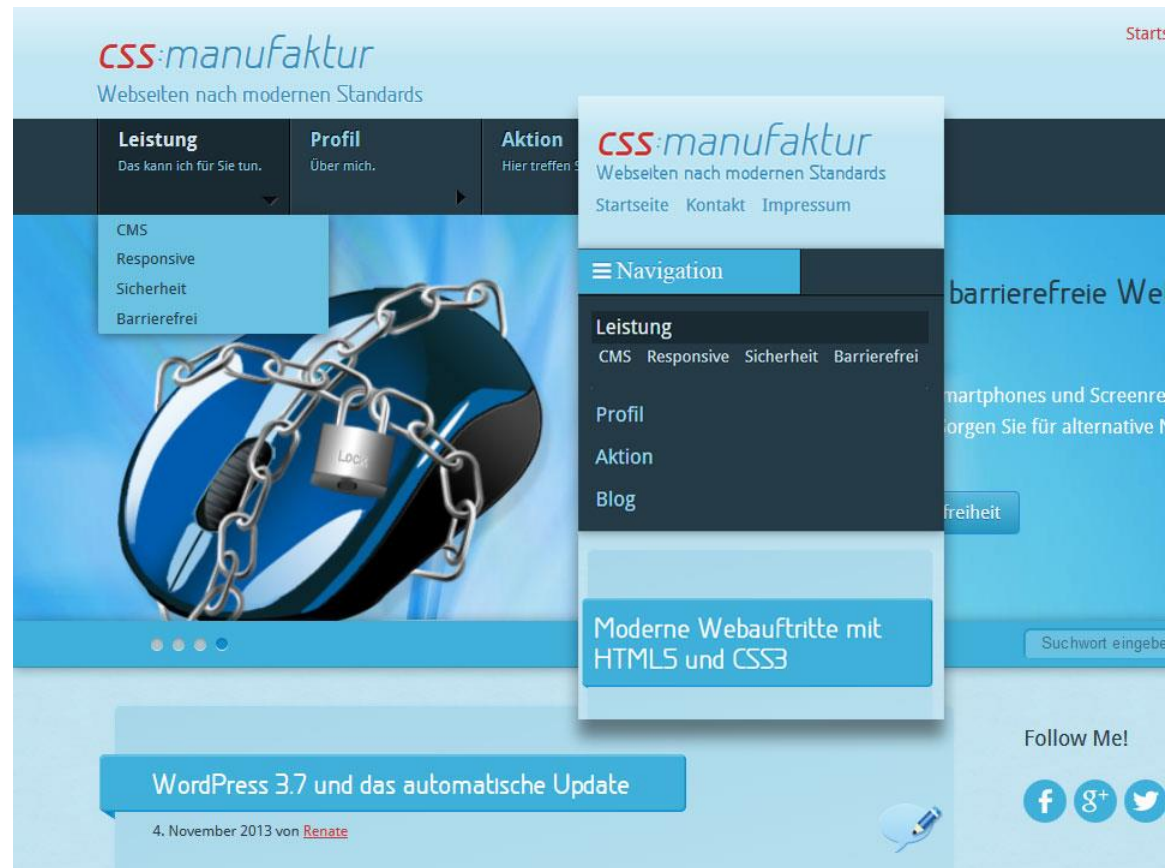
- <meta name="viewport" content="width=device-width, initial-scale=1.0, user-scalable=1">

Detailinfos: <http://maddesigns.de/meta-viewport-1817.html>

Beispiel 1 - <http://www.gruelo.de/>



Beispiel 2 - <http://www.css-manufaktur.de/>



Mobile First ↔ Desktop First

2 unterschiedliche Ansätze

- Aufrüsten (Mobile First) => Progressive Enhancement
 - Basis ist, was alle Geräte können.
 - Neuste Techniken und breitere Layouts werden stufenweise hinzugefügt.
- Abspecken (Desktop First) => Graceful Degradation
 - Darstellung auf dem Desktop ist die Basis.
 - Anforderungen werden stufenweise reduziert.
 - Neuste Techniken fallen weg oder werden ausgeblendet.

„The Infinite Grid“

- das Endlosraster

- **Hierarchie:** Was steht wo im Layout?
- **Dichte:** Wie viele Details sollen gezeigt werden?
- **Interaktion:** Linkliste oder Dropdown? Slider oder Bilderalbum?
- **Breite:** Flexibel = mit max-width bzw. anteilig in Prozent?
Oder fix mit festem Pixelwert?
Fluides Grid ↔ Pixel-Grid

Quelle: <http://alistapart.com/article/the-infinite-grid>

Design System

- Kernkomponenten sind nicht veränderbar.
 - Dazu gehören: Typografie, Layout, Form und Farbe,
 - Ebenso die Inhalte und das “Look and Feel”.
- Veränderbar können dagegen sein:
 - Grid-Raster, Verteilung im Layout, Schriftgröße und –höhe und die Satzlänge.
- Der Inhalt muss immer gleich bleiben!

Quelle: <http://24ways.org/2012/design-systems/>

Design - Proportionen

- **Pixel** bestimmen ein Element nach seiner spezifischen Auflösung.
- **Ems** skalieren ein Element relativ zu seiner Fontgröße; **Rems** skalieren relativ zur Basis-Fontgröße des Dokuments. (IE7 + 8 können keine Rems)
- **Prozent** skalieren ein Element relative zum umgebenden Container.
- **VH** (Viewport-Hight) und **VW** (Viewport-Width) bestimmen ein Element in Relation zum Viewport. (Neue Einheit, wird noch nicht von allen Browsern unterstützt.)

Und was sagen die Browser?

- IE7 kann vernachlässigt werden.
- IE8 wird zunehmend unwichtiger, kann aber für Win XP je nach Anforderung noch wichtig sein.
- Desktop – Level A Browser unterstützen CSS3-Media Queries.
- Gute Unterstützung der modernen Techniken bei den meisten mobilen Plattformen – Ausreißer z.B. Opera Mini, daher guter Test-Browser für Mobile First-Ansatz.

Werkzeuge

- Frameworks
 - <http://html5boilerplate.com/>
 - <http://getbootstrap.com/>
- Browser-Tools
 - <http://pea.rs/content/blog-post>
- Prototyping mit Grid
 - <https://gridsetapp.com/> kostenpflichtig
 - <http://responsive.is/> kostenpflichtig
- Wireframes/Mockups
 - <https://moqups.com/home/>

Empfehlungen

- Weniger als 45 Zeichen Breite fühlt sich „eng“ an. Optimales Lesegefühl sind ca. 60 Zeichen. Was darüber hinaus geht wird zunehmend „unleserlich“.
- Zu viel Weissraum rechts und links lässt den Inhalt unwichtig erscheinen.
- Fallback bei neuen CSS3-Komponenten und fehlendem JavaScript einplanen.
- Usability im Auge behalten!

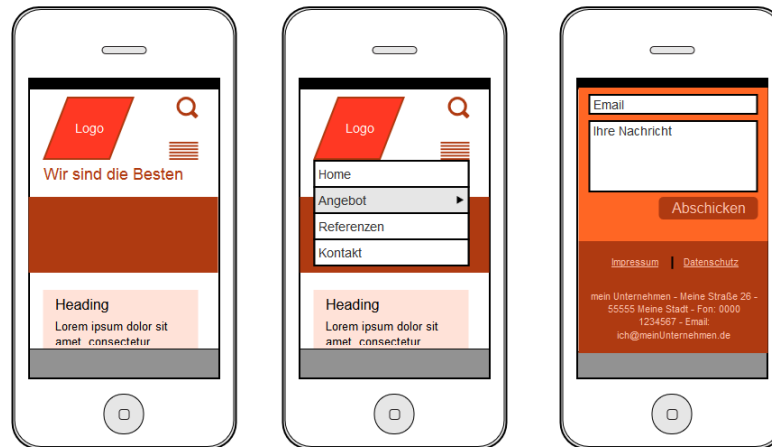
Realisierung - Konzeption

- Navigationskonzept
 - Breiter Bildschirm – horizontale Navigation am Kopf der Seite
 - Schmalere Bildschirm – Dropdown-Menü/Klappmenü/Off-Screen am Kopf
 - Oder Link-Menü am Fuß der Seite
- Screens in den wichtigsten Breiten planen
- Normgrößen für Breakpoints ↔ eigene Breakpoints
- Zuerst die Extreme – größte und kleinste Breite – dann die Zwischengrößen und Übergänge planen.

Hilfsmittel

- JavaScript-Unterstützung für mobile Navigation (onclick-Handler)
- Modernizr zur Abfrage der Browser-Unterstützung für diverse CSS3-Komponenten

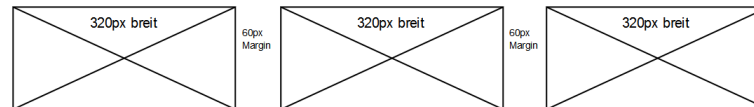
Zuerst die Extreme - klein



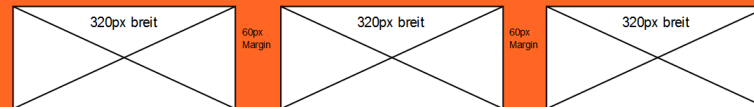
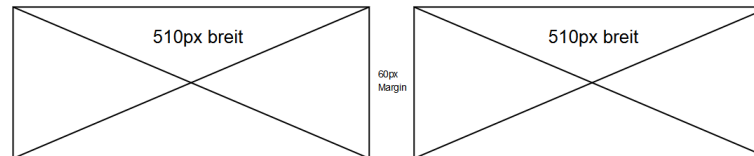
Link zum Moqup: <https://moqups.com/Renaade/KBFt3UN3/>

Zuerst die Extreme - groß

1080px Breite					
1080px Breite					Fluides Grid berechnen:
Grid	1 Spalte	Umrechnung	Margin	Umrechnung	Spaltenverteilung
1 SP	1080px	100%	0	0	1
2 SP	510px	47.2222222222%	1*60px	5.55555557%	2
3 SP	320px	29.62962962% 2*29.62962962%+1*5.5555557% = 64.81481481%	2*60px 1*60px	11.1111114% (*2) 5.5555557% (*1)	3/3 1/3+2/3 2/3+1/3
4 SP	225px	20.83333333% 41.66666666%(*2) 62.49999999%(*3)	3*60px 2*60px 1*60px	16.66666671% (*3) 11.1111114% (*2) 5.5555557% (*1)	4/4 2/4+2/4 1/4+3/4 3/4+1/4



Hier steht die tollste Überschrift



1080px Breite

Ausschnitt fluides Grid berechnen

1080px Breite

Fluides Grid berechnen:

Grid	1 Spalte	Umrechnung	Margin	Umrechnung	Spaltenverteilung
1 SP	1080px	100%	0	0	1
2 SP	510px	47.22222222%	1*60px	5.55555557%	2
3 SP	320px	29.62962962% 2*29.62962962%+1*5.55555557% = 64.81481481%	2*60px 1*60px	11.11111114% (*2) 5.55555557% (*1)	3/3 1/3+2/3 2/3+1/3
4 SP	225px	20.83333333% 41.66666666%(*2) 62.49999999%(*3)	3*60px 2*60px 1*60px	16.66666671% (*3) 11.11111114% (*2) 5.55555557% (*1)	4/4 2/4+2/4 1/4+3/4 3/4+1/4

Weiterführende Links

(Frameworks)

- HTML5-Framework <http://framework.gregbabula.info/>
- Framework Boilerplate <http://html5boilerplate.com/>
- Responsive Grid-System <http://responsiveboilerplate.com/>
- Framework auf Basis von Boilerplate <http://www.rootstheme.com/>
- JS-Bibliothek <http://www.modernizr.com/>
- Übersicht Frameworks <http://usablica.github.io/front-end-frameworks/compare.html>
- Skeleton – noch ein Framework <http://www.getskeleton.com/>

Weiterführende Links

(Wordpress)

- “Bones” - Wordpress-Theme auf der Basis von Boilerplate
<http://themble.com/bones/>
- Tutorial zu Bones <http://code.tutsplus.com/tutorials/making-a-theme-with-bones-getting-started--wp-26545>
- „Roots“ - Wordpress-Theme auf Basis von Boilerplate und Bootstrap
<http://roots.io/starter-theme/>
- „Skeleton“ – Wordpress-Theme auf Skeleton-Basis
<http://themes.simplethemes.com/skeleton/>
- Tutorial zu Skeleton <http://webdesign.tutsplus.com/tutorials/skeleton-to-wordpress-getting-up-and-running--webdesign-7943>

Weiterführende Links

(Tools + Snippets + Polyfills)

- Tools <http://www.splashnology.com/article/useful-html5-css3-toolbox-for-web-developers/250/>
- CSS + JS minimieren <http://refresh-sf.com/yui/>
- Konvertierung Pixel in REM <http://foliovision.com/2013/03/responsive-design-calculator>
- Konvertierung Zeichen (Entities) <http://www.evotech.net/articles/testjsentities.html>
- HTML5-Basislayout Builder <http://switchtohtml5.com/>
- Jede Menge Code-Schnipsel <http://perishablepress.com/code-snippets/>
- Boilerplate-Schnipsel erklärt <http://www.coopweb.de/2012/05/11/20-schnipsel-aus-html5-boilerplate-die-man-nutzen-sollte/>
- Polyfill „Media-Queries in IE 6-8“ <https://github.com/scottjehl/Respond>

Weiterführende Links

(Testing)

- Responsive Design Testing <http://www.browserstack.com/responsive>
- Dto. <https://www.responsinator.com/>
- Browser-Resizer <http://www.resizemybrowser.com/>
- Welcher Browser kann was in HTML5/CSS3 <http://caniuse.com/>

Weiterführende Links

(Tutorials + Präsentationen + Artikel)

- Präsentation Webinare „Responsive Webdesign“
<http://www.slideshare.net/djessse/unwissenheit-ist-ein-segen-responsive-webdesign>
- Navigationsbeispiele für Mobilgeräte
<http://webkrauts.de/artikel/2013/navigation-im-responsive-design-teil-1>
<http://webkrauts.de/artikel/2013/navigation-im-responsive-design-teil-2>
- Tutorial mobile Navigation mit Drop-Down
<http://osvaldas.info/drop-down-navigation-responsive-and-touch-friendly>
- Flex-Slider einbauen <http://www.elmastudio.de/programmierung/den-responsive-bilder-slider-flexslider-nutzen-%E2%80%93-so-funktioniert/>
- Design-System aufbauen
<http://css-tricks.com/design-systems-building-future/>
- display: table-footer-group
<http://webkrauts.de/artikel/2014/flexbox-light-das-neue-layoutprinzip-schon-heute-anwenden>
- Artikel über Berechnung vom REM <http://www.elmastudio.de/programmierung/css/css-tipp-rem-als-einheit-fur-schriftgroesse-nutzen/>

Weiterführende Links

(sonstige Links)

- User Group bei Google+
<https://plus.google.com/+RwdpraxisDeNews/posts>
- Navigationsbeispiele für Mobilgeräte
<http://webkrauts.de/artikel/2013/navigation-im-responsive-design-teil-1>
<http://webkrauts.de/artikel/2013/navigation-im-responsive-design-teil-2>
- Design-System aufbauen
<http://css-tricks.com/design-systems-building-future/>
- display: table-footer-group
<http://webkrauts.de/artikel/2014/flexbox-light-das-neue-layoutprinzip-schon-heute-anwenden>
- Buchempfehlungen:
<http://www.hanser-fachbuch.de/buch/Responsive+Webdesign/9783446430150>
<http://www.hanser-fachbuch.de/buch/Mobile+Webseiten/9783446431188>