Indisoft

Responsives Webdesign

04.09.2018

Dozent erste Woche: Elmar Becker (ist für Doku wichtig)

Elementgrößen

Prozent des Einzelelements=Zielgröße/Kontext\*100

Z.B. Navi Seite  
300/1080=27,77777778%  
  
Prinzipiell Wrapper 90%, damit angenehmer Seitenabstand  
  
Man kann für übergroße Bildschirme zunächst einen max-width definieren, damit die Zeilen nicht allzu arg auseinander gezogen werden ….

#wrapper{  
 max-width:1080;  
 width:90%;

}

Background-Komponenten:   
Als Oberdefinition:  
background: transparent url(images/bild.jpg) 750px bottom no-repeat;  
Die Breitenangabe gibt hier die gesamte Leerfläche des Containers an, nicht aber die einzelnen Seitenelemente

background-size:100% auto; x-Achse 100%, y-Achse auto  
background-size:100% 100% in beiden Achsen 100%, verzerrt dann aber;  
background-size:cover; das Bild füllt den Rahmen  
background-size: contain: Das Bild füllt einen der beiden Achsen, die andere wird autoformatiert.

Bilder formatieren:  
Entweder gibt man dem Rahmen eine eigene id/class, oder aber man versieht das Bild selbst mit eigener id. Dann

max-width: 100%;  
height: auto;

Damit bei Umbrüchen die padding-Abstände nicht stets zur width dazugerechnet werden, gibt man an:  
box-sizing:border-box;  
Hiermit werden die padding-Abstände hinzugerechnet …

CSS-Klasse first-of-type nimmt sich des ersten Kindelements des gleichen Namens an, das ein Tag hat

Vidoetraining:  
hochauflösende Fotos …  
Ethan Marcot:  
RD von responsiver Architecture  
flexible Raster, Medien, Queries  
Die Information steht im Mittelpunkt, keine Informationen völlig verstecken!  
Kompakte Navigationen und flache Hierarchien in der Navigation

Bei alternativen Versionen (Desktop/Smartphone) muss ein Link in der Startseite vorhanden sein, wo es hingehen soll. Weiterhin muss von Desktop zu Smartphone und umgekehrt verlinkt sein.  
Responsive Webdesign umgeht die Schwierigkeiten, daher präferiert Google RD.  
Es gibt noch adaptives Design, hier wird je nach Gerät das Design ausgewählt. Meist werden nur Komponenten adaptiv ausgetauscht, z.B. Medienfiles. (Hybrid, RESS = Responsive + Server Side),   
adaptives Layout ist recht statisch, weil immer von einer zur anderen Größendimension gesprungen wird. Letztlich ist es auch Platzverschwendung, da immer Flächen, die nicht bei den Größen passen, leer bleiben.  
Vorteil: es ist unkomplizierter.

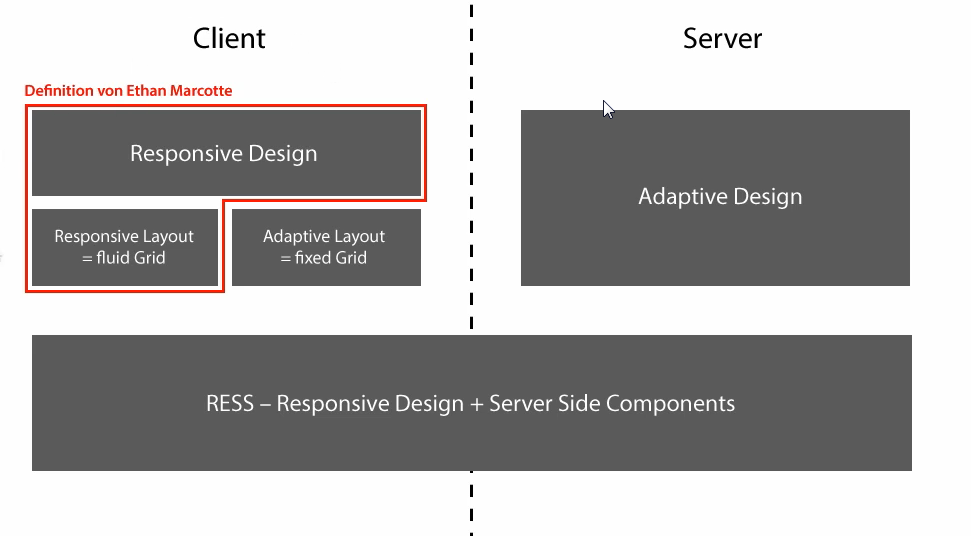
Verschiedene Layouttypen  
Static, fluid und elastic  
Fluid hat Ober- und Untergrenze, Planung ist komplexer, die Elemente werden in den Proportionen auch verändert.Oft werden alle 3 Layouts kombiniert, fluid und elastic sind echte responsive Techniken.

Es gibt für die verschiedenen Größen Breakpoints.

Man kann im Link verschiedene Medientypen angeben …   
Man kann @import ein zweites Stylesheet in ein anderes CSS einbinden …

In der CSS-Datei @media print für Druckausgabe … Hier kann ich mit „display:none; „ bestimmte Element ausblenden/umformatieren …  
media queries können verschiedene Möglichkeiten abfragen: neben and(min/max-width) gibt es and (orientation:landscape/portrait)

Responsives Layout im Gegensatz zu adaptivem formatiert für alle Größen = keine Platzverschwendung. Es steht das Design und Inhalt im Vordergrund, nicht das Gerät. Nachteil: komplexere Planung und Gestaltung, Pflege für den Kunden anspruchsvoller.



Auf dem Server ist letztlich das adaptive Layout gespeichert, beim Client wird dann möglichst nach Ethan Marcotte dargestellt.  
RESS ist dann die Hybridlösung

In der Regel wird heutzutage mobile first entwickelt. Beim Inhalt spricht man von „content first“ …  
Bei mobile-first wird weniger Code benötigt, weil kleiner Größe ohnehin als Standard 100%-Formatierungen haben, daher muss man hier nicht sonderlich viel am Anfang formatieren.  
Man braucht deutlich weniger Code und es lässt sich viel schneller entwickeln.  
Da bis IE 8 keine Media-Querys verstanden werden, sollte ggf. mit Respond.js nachgerüstet werden, es werden hiermit aber nur die media-queries min-und max-width geladen (was ja in der Regel reicht).

Für ein Touch-Screen sollten es für Schaltflächen mindestens 44px eingeräumt werden.  
Slider sind wichtig, Pfeile rechts und links reichen nicht.  
Mouseover-Effekte sind nicht mehr sinnvoll, da es natürlich auf Touchs ankommt.

In der Chrome-Entwickler-Ansicht kann man gut direkt in der Ansicht entwickeln, dann nur noch kopieren und in den Quelltext einfügen.

[www.caniuse.com](http://www.caniuse.com) zur Sicherstellung von Browserkompatibiltäten, [www.simevidas.jsbin.com/gufoko/quite](http://www.simevidas.jsbin.com/gufoko/quite) als Möglichkeit eines Autoprefixers, aber letztlich ist   
[www.modernizr.com](http://www.modernizr.com) Standard, der Klassen und Funktionen abprüft; damit ggf. png gegen svg ausgetauscht wird, svg ist hochauflösend  
<https://www.webdesignrankings.com/resources/rqrwd/>

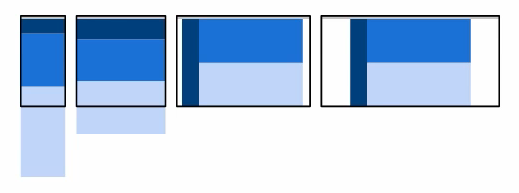
In der CSS kann man mit   
#element{  
 width:calc(100% - 440px -50px);   
}  
einem Element eine Weite zuweisen, die jene eine anderen Elementes abzieht.

10.09.18  
Ab jetzt Dozent Michael Hassel  
  
Wenn man Container positioniert, dann kann man in der Reihenfolge bei Veränderung responsive strukturieren, man kann dann aber auch (mit einer Klasse) ein float:right festlegen, sodass verschiedene Elemente nacheinander nach rechts gefloatet werden.  
  
**Mostly fluid**

Bei Google beschrieben, oder aber auch Responsive Patterns bei [https://github.com/bradfrost/this-is-responsive/](https://github.com/bradfrost/this-is-responsive/patterns.html)

Unterelemente kann man in css über nth-of-type ansprechen, z.B.  
.className:nth-of-type(3)

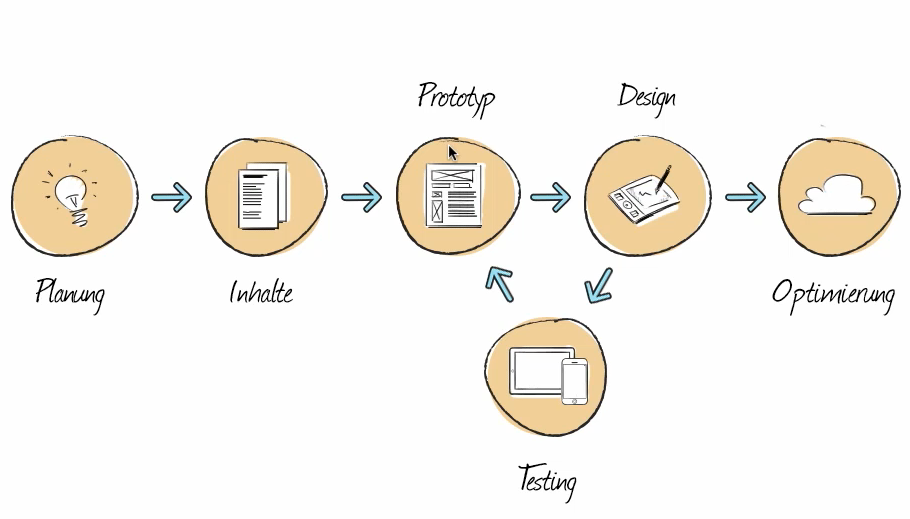
Column-Drop-Layout:  
Es rücken nach und nach die Spalten, die in der Desktop-Ansicht nebeneinander liegen, untereinander …

Layout-Shifter   
  
  
  
Kacheldesign

Je nach Anzahl der Kacheln width:x%; padding:0 0 x% 0;

Fixieren des Footers am unteren Rand. Container min-height:100% bewirkt, dass der Inhaltsbereich immer so hoch ist wie die Webseite, solange der Inhalt nicht länger/höher ist.  
  
Mit vh kann ich in der CSS eine viewport-height festlegen, bei 100% ist das immer die jeweilige Browserhöhe, vw gibt eine Bildschirmbreite …

Planung eines Webauftritts bei responsivem Design (Feedbackphase viel länger und auch nach Seitenerstellung i.d.R. öfter)



**Frameworks/Prototypen**

Vorlagen über   
<https://foundation.zurb.com/prototyping.html>

Prototypen am besten so abgespeckt und funktional/häßlich wie möglich entwerfen, damit Kunde weiß, dass es noch viel grafische Arbeit gibt.  
<http://demo.protosharesites.com/dr-flue-mobile/Photos/index.html>  
<http://getbootstrap.com/docs/4.1/examples/>

Entweder selbstentwickeln oder auf Codevorlagen und Pattern zurückgreifen  
<http://bradfrost.github.io/this-is-responsive/patterns.html>

Speziell für Navigationen  
[www.responsivenavigation.net/index.html](http://www.responsivenavigation.net/index.html)

Arbeiten mit Frameworks

Ebenfalls über bootstrap oder foundation

<http://getbootstrap.com/docs/4.1/components/alerts/>

Prototyping-Tools (hier wird aber kein Quellcode angewandt, sondern muss im Nachhinein entwickelt waren. Aber zur Konzeption optimal,   
[www.balsamiq.com](http://www.balsamiq.com)  
[www.axure.com](http://www.axure.com)  
[www.creative.adobe.com](http://www.creative.adobe.com)

**Design-Module**

Seiten sind über Module aufgebaut. Alle Module zusammen nennen sich Content Inventory.  
(idealerweise sollten Module mit gleicher Funktion auch gleich oder wiedererkennbar ähnlich aussehen)

Es gibt Ansätze zur Gestaltung, z.B. bradfrosts Atomic Design. Er geht von atomaren Strukturen aus, die sich dann aufeinander aufbauen.  
Es gibt die Schritte  
Atom-Molekül-Organismus-Template-Seite  
<http://demo.patternlab.io/>  
Hier kann man die einzelnen Schritte anschauen und ggf. zusammenstellen.

**Styles/Design**

Über   
[www.bootswatch.com](http://www.bootswatch.com)  
kann man auch online Looks der einzelnen Elemente anpassen  
Bootstrap Standarddesign  
<https://bootswatch.com/default/>

Über  
<http://styletil.es/>  
<http://samanthatoy.com/washington-examiner/>

kann man verschiedene Style optimieren und entwickeln. Hier kann man den Look einzelner Module entwickeln. Man kann hier verschiedene Stilvariationen ausarbeiten.

Andere Styleguides:  
<https://www.starbucks.com/static/reference/styleguide/>  
<https://www.mozilla.org/en-US/styleguide/>  
<https://design.firefox.com/>

**Breakpoints**

Design-Größen:  
320px IPhone4  
767px IPad hoch  
1023px IPad Quer

Da die Information im Mittelpunkt steht, wird umgebrochen, wenn das Design auf dem Gerät einen Breakpoint braucht.  
Breakpoints niemals aufgrund von Geräten erstellen (es gibt ohnehin mittlerweile viel zu viele)  
  
**Testing**

Am PC kann man hochauflösende Displays schlecht simulieren. Auch der Betrachtungsabstand ist nicht simulierbar.  
Auf echten Geräten testen: parallel über synchronisieren von Inhalten über Edge Inspect CC (Adobe)  
<https://creative.adobe.com/de/products/inspect>  
Die Erweiterung muss dann bei Chrome als PlugIn eingerichtet werden. Es muss dann eine App auf dem mobilen Endgerät eingerichtet werden …  
Man muss dann im gleichen WLAN die Seite entwickeln und es wird dann sogleich synchronisiert.  
  
Es gibt weiterhin   
[www.vanamco.com/ghostlab](http://www.vanamco.com/ghostlab)  
Dieses Tool braucht keinerlei App auf einem Endgerät. Man kann dann aber jegliche Webseite mit einer index.html/.php testen. V.a. kann man verschiedene Browser testen!!!!  
  
**Raster**  
  
Nicht-Raster-geordnete Layouts werden unruhiger, vom Gehirn anstrengender und unprofessioneller von der Gestaltung her wahrgenommen.

CSS-Grid-System

Für jede mögliche Spaltenbreite stellt das CSS-Grid eine Klasse zur Verfügung. Bestimmte Elemente erhalten dann keine eigene Breite mehr, sondern ihnen wird eine Rasterbreite zugewiesen, was zu einem einheitlicheren Layout führt.  
Demonstration des Gridsystems auf  
<https://960.gs/>  
Hier gibt es ein 12- und ein 16-spaltiges Grid-System.

Das 960.gs ist nicht mehr ganz aktuell, weil es immer noch mit px arbeitet statt mit %-Werten.  
Im responsiven Design wird mit verschiedenen Layouts gearbeitet, die bei diversen Breakpoints unterschiedliche Raster zur Verfügung stellt.