Mobiles Web - Ein Blick zurück

Die Ursprünge von HTML reichen zurück bis ca. 1990. Der ursprüngliche Fokus lag bei der Veröffentlichung von statischen Inhalten in Form von wissenschaftlichen Texten und Bildern (Universität CERN). Über Verlinkungen sollten thematisch ähnliche Quellen leicht erreichbar sein.

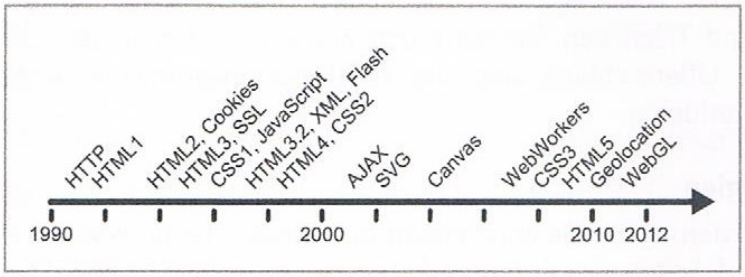


Bild 1.1: Zeitliche Einordnung der Entstehung einiger grundlegenden Technologien für Web-Applikationen.

Das Erstellen von Apps war nicht das Ziel. Das wäre auch gar nicht gegangen. Das zustandslose http-Protokoll passt nicht dazu. Jede Anfrage ist isoliert und statisch.

Seit 2005 kam ein neuer Begriff hinzu, AJAX (Asynchronous JavaScript and XML). Mit Hilfe von dynamischen und asynchronen Anfragen bringt AJAX Dynamik in den Browser. Die HTML-Seite wird weiterhin auf dem Server erzeugt und der größte Teil bleibt statisch. Kleine Bereiche werden je nach Benutzeraktionen nachgeladen und „on the fly“ ausgetauscht.

Typische Beispiele sind Vorschlagslisten („google-Suche“), die sich abhängig von der Eingabe anpassen und Vorschläge automatisch vervollständigen.

Aus diesen ersten Anfängen haben sich die aktuellen mächtigen Frameworks, allen voran jQuery, entwickelt.

Weitere Gründe, warum sich das Web so entwickelt hat:

- JavaScript wird schneller, da die Entwickler von Browsern, vor allem Google, das Interpretieren von JavaScript im Browser deutlich beschleunigten
- Produktive Frameworks helfen bei Routineaufgaben und geben Strukturen vor.

Evolution of the Web: www.evolutionoftheweb.com

Vor der Smartphone-Revolution, die Apple 2007 mit dem iPhone auslöste, war die Welt des Webdesigners noch einfach. Eine Website wurde in erster Linie für den Desktoprechner entwickelt und häufig wurde eine Standardbreite von 960 Pixeln verwendet.

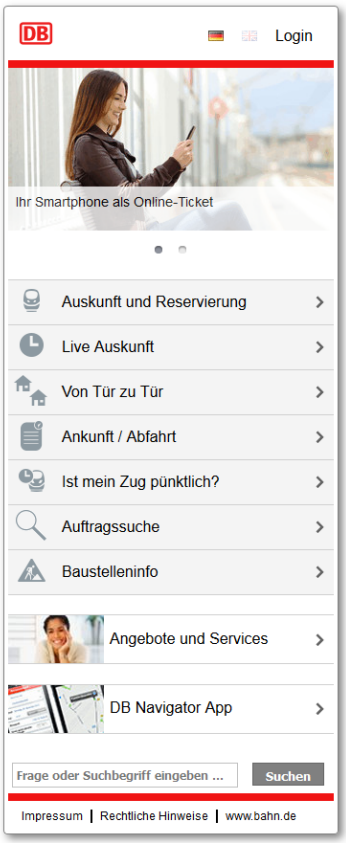
Es gibt heute im Wesentlichen **drei Kategorien** von Geräten zum Betrachten einer Website:

- Smartphones wie das iPhone und Android-Geräten
- Tablets
- Desktop-PC bzw. Notebooks

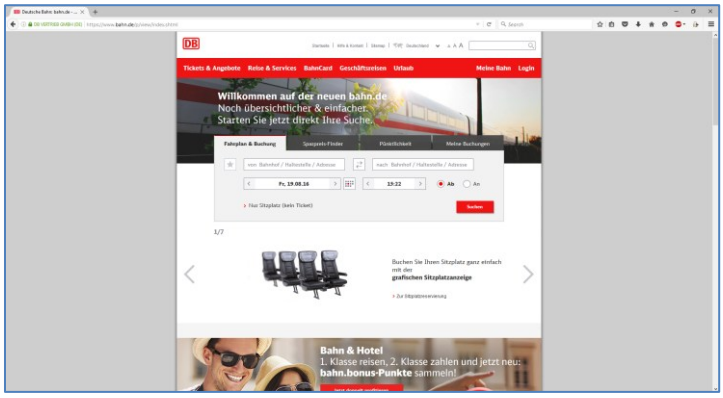
Jedes dieser Geräte besitzt unterschiedliche Bildschirmmaße. Außerdem können Websites auf Smartphones und Tablets auch noch sowohl im Hochkant- als auch im Querformat (landscape) betrachtet werden, was die Anforderungen an den Webdesigner noch einmal erhöht.

Erste Lösungen für Mobilität:

Ein Weg, das Problem der verschiedenen Gerätegrößen zu lösen, war es, unterschiedliche Versionen der Website zu erstellen, z.B. eine Desktopversion und eine Mobile-Version. Das kostete aber viel und war doppelte Arbeit.



Beispiel:
www.bahn.de bzw. mobile.bahn.de



Mobile Webapplikation

Auch wenn es mobile Apps bereits seit den 1990er Jahren gibt, liegt der Beginn nicht so weit zurück. Am **10. Juli 2008** begann der weltweite Siegeszug mobiler Apps durch die **Eröffnung des Apple App Store**. Seit diesem Tag können iPhone- und iPod-Benutzer mobile Apps für ihr mobiles Endgerät von einer zentralen Stelle laden. Gleich am ersten Wochenende nach der Eröffnung wurden mehr als 10 Millionen mobile Apps daraus heruntergeladen.

Der Gründer von Apple, Steve Jobs, präsentiert **2007** das erste iPhone:
https://www.youtube.com/watch?v=MnrJzXM7a6o

Zwei Jahre später drängten dann die Android-basierten Smartphones auf den Markt, einhergehend mit einer großen Vielfalt an unterschiedlichen Geräten und Displaygrößen, die heute abgerundet werden von einem Angebot an Tablet-PCs, ebenfalls mit sehr unterschiedlichen Auflösungen und Formfaktoren.

Diese Situation macht die Erstellung von nativen Apps mit einer breiten Plattformunterstützung sehr aufwändig, da Code nicht einfach wiederverwendet werden kann. Jede Plattform bringt ihre eigenen Sprachen, Entwicklungswerkzeuge und Eigenarten mit. Hat man aber eine größere Gerätevielfalt abzudecken, kommt schnell der **Wunsch nach einer Cross-Plattform-Technologie** auf, um eine **App nur einmal entwickeln** und dann mit keinen oder nur minimalen Anpassungen überall verwenden zu können. Hierzu bieten sich natürlich die Browser an und das mittlerweile allgegenwärtige HTML5, das antritt, das Cross-Plattform-Versprechen einzulösen.

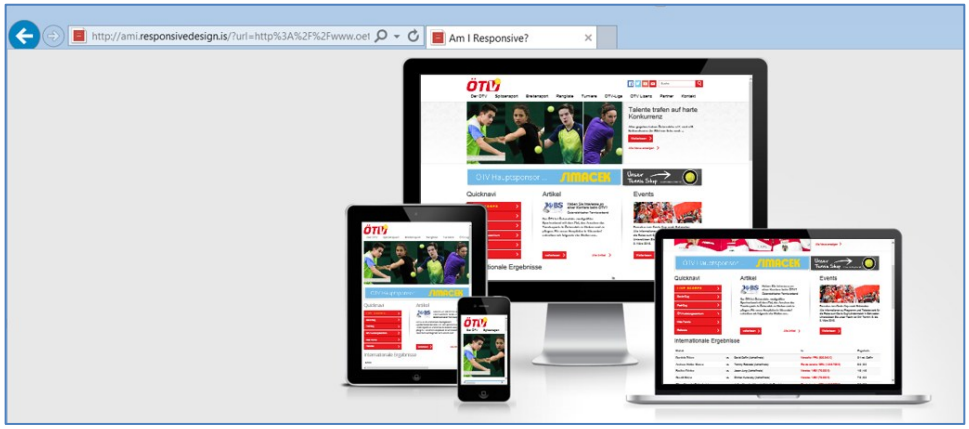
1)Responsives Webdesign

In seinem Artikel für A List Apart im 24. Mai 2010 (http://www.alistapart.com/articles/responsive-web-design) verwendete Ethan Marcotte zum ersten Mal den Begriff „Responsive Webdesign“ und seitdem erlebte er einen wahren Boom. Responsive Webdesign verwendet grundsätzliche CSS, um die Anpassungen an den unterschiedlichen Bildschirmen durchzuführen – und zwar über Formatierungen, die nur unter bestimmten Umständen gelten.

Mit dem Begriff „responsives Webdesign“ wird das selbstanpassende Verhalten entsprechender Websites benannt. Man versteht darunter, dass ein Web-Layout automatisch auf unterschiedliche Bildschirmgrößen (Viewports) reagiert und sich entsprechend anpasst. Das gilt nicht nur für die Bildschirmgrößen, sondern auch für das Format, d.h. ob sich ein Gerät in Landscape-Modus oder im Portrait-Modus befindet, also quer oder hochkant gehalten wird.

Ein „responsives“ Layout funktioniert gleichermaßen auf einem Smartphone, auf einem Tablet oder einem Desktop-Rechner, wobei die Darstellung für die jeweilige Umgebung optimiert wird.

Beispiel: www.oetv.at



Ein guter Vergleich wie es aussieht: http://ami.responsivedesign.is/

Was ist responsives Webdesign?

- Design passt sich „automatisch“ an das verwendete Endgerät an (Monitor, Tablet, Smartphone)
- Design sieht überall „gut“ aus (Schrift ist lesbar, Bilder passen sich im Verhältnis richtig an usw.)
- Seite ist auf allen Endgeräten „gut bedienbar“ (Hinein- und Herauszoomen ist nicht notwendig)
- Navigation (Menü) passt sich an das Endgerät an
- Inhalte (HTML) bleiben unverändert
- Anpassungen werden per CSS und JavaScript (jQuery) vorgenommen

Responsive Webdesign – die „Zutaten“

- HTML 5
- CSS 3 (inkl. Media Queries – „Breakpoints“ für die Devices)
- JavaScript
- „flüssige“ Bilder (in Bootstrap mit der Klasse class=“img-fluid“)
- flexibles Grid-System, „flüssiger“ Raster (Spalten für Design)
  (Bootstrap, Foundation)

1a)Entwicklung responsiver Websites

Dafür sind zwei Systeme besonders sinnvoll:

- Mobile First und
- Breakpoints für den Inhalt (Media Query).

Desktop First

Die typische Vorgehensweise in den Anfangszeiten war es, ein Desktop-Layout zu nehmen und zusätzliche Angaben für kleinere Viewports zu ergänzen. Das Desktop-Layout ist damit der Normalfall. Darauf basierend werden die abweichenden Layouts erstellt. Im Bedarfsfall werden bestimmte, auf kleinem Bildschirm nicht benötigte Inhalte per display: none versteckt.

Mobile First

Im Gegensatz zum Desktop First-Ansatz steht beim Mobile First-Ansatz das Layout für Smartphones und Co. am **Anfang der Überlegungen** und des Designprozesses. Der Aufbau sieht anders aus. Zuerst einmal würden wir eine etwas andere Quellcode-Anordnung wählen: der Inhaltsbereich steht zuerst, danach kommt die Navigation.

Webdesigner Luke Wroblewski prägte schon 2009 den Begriff „Mobile First“ und riet dazu, nicht mit der „großen“ Desktopsite zu beginnen, sondern mit der Mobilversion. Er nennt drei Hauptgründe dafür:

1. Die Nutzung mobiler Geräte ist in den letzten Jahren prozentual stark angestiegen. Somit sind Unternehmen mit diesem Konzept für den wachsenden Markt gerüstet.
2. Das Design für mobile Geräte zwingt Entwickler dazu, sich auf das Wesentliche zu beschränken, da der Platz auf mobilen Geräten begrenzt ist.
3. Wenn schon bei der Planung auf die vielen Features der Smartphones eingegangen wird, z.B. GPS. Multi-Touch und Beschleunigungssensor, ergibt sich ein Mehrwert.

1b)Rasterlayout - Grid

Bei responsivem Design erfolgt die Anpassung meist über ein Gestaltungsraster, bei dem der Bildschirm in 12 oder 16 Spalten aufgeteilt wird, in sogenannte „Grids“. Die einzelnen Elemente werden in das Raster eingepasst, je nach Gerätegröße wird entschieden, wie viele Spalten ein Element belegen soll.

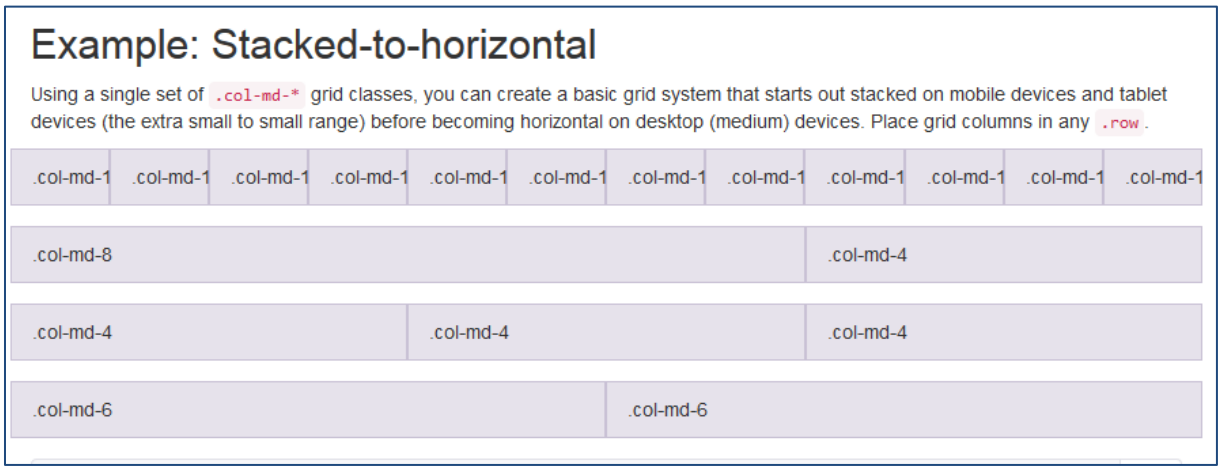
- In der kleinsten Bildschirmgröße wird das Design meist linearisiert, also alle Elemente werden untereinander angezeigt. z.B. wird auf einem Smartphone alles einspaltig dargestellt.
- Während es auf großen Displays mehrspaltige Layouts gibt.

http://www.nitinh.com/2013/03/vertically-responsive-design-keeping-things-above-the-fold/

Raster (*grids*) sind ein grundlegendes Design-Element.

Typische Rasterlayouts findet man in Bootstrap, Foundation usw.

www.getbootstrap.com,
http://foundation.zurb.com



Beispiel Bootstrap - das Raster im Detail

Das Raster wird durch zwölf (12) gleichbreite Spalten gebildet. Vordefinierte Klassen können benutzt werden, um Elemente ab einer bestimmten Spalte und über eine Anzahl Spalten hinweg zu platzieren.

Bootstrap-**Breakpoints** sind die **Standard-Schwellenwerte für Bildschirmbreiten**, ab denen Bootstrap **zusätzliche (responsive) CSS-Regeln** aktiviert. Bootstrap arbeitet dabei **mobile-first** und nutzt überwiegend **min-width-Media-Queries**: Styles gelten zuerst für kleine Screens und werden dann für größere „oben drauf" gelegt.

Bei Bootstrap gilt: **Breakpoints wirken "ab" dem Wert und für alles darüber** (weil min-width).

Bootstrap-Breakpoints (Standard)

- **sm**: ≥ 576px
- **md**: ≥ 768px
- **lg**: ≥ 992px
- **xl**: ≥ 1200px
- **xxl**: ≥ 1400px

Beispiel:

```html
<div class="col-12 col-md-6">...</div>
```

- col-12 gilt **immer** (xs und größer)
- col-md-6 gilt **ab md (≥ 768px)** – und bleibt auch bei lg/xl/xxl aktiv

1c)Das Grundprinzip des Responsive Webdesigns = Media Queries

Media Queries sind CSS-Regeln, mit denen du das Design **an Geräteeigenschaften anpasst** – vor allem an die **Bildschirmbreite** (Responsive Design), aber auch z. B. an Höhe, Ausrichtung, Auflösung oder „Dark Mode“.

Mithilfe der in CSS3 definierten media-Befehle werden dabei die Eigenschaften des Anzeigegerätes abgefragt und die Website automatisch angepasst. So kann beispielsweise eine Webseite bei viel verfügbarem Platz dreispaltigem und bei wenig verfügbarem Platz einspaltig angezeigt werden.

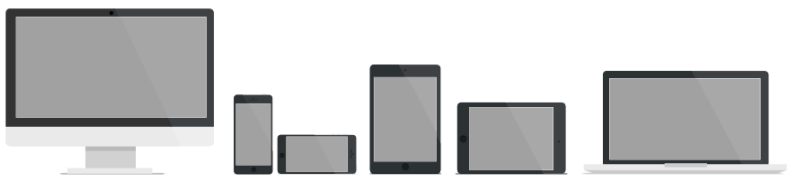
Häufige Arten von Media Queries betreffen die Breite: min-width / max-width

- min-width: **ab** dieser Breite (mobile-first)
- max-width: **bis** zu dieser Breite

```css
@media (max-width: 600px) { /* Smartphone */ }
@media (min-width: 992px) { /* Desktop */ }
```

https://www.mediaevent.de/css/media-query.html

Media Queries passen das Layout der Webseite durch Breakpoints im CSS an Monitorklassen an. An den Breakpoints springt das Design um. Es gibt gute Frameworks, die das Grundgerüst für das Layout automatisch erzeugen



Interessante Links:
http://grochtdreis.de/

http://grochtdreis.de/weblog/2012/03/20/am-ende-ist-doch-alles-html-2/

http://www.responsive-webdesign-praxis.de/