

# RESPONSIVE WEBDESIGN – FACHBEGRIFFE UND UMSETZUNG

## 1 Kurzzusammenfassung

Um Webseiten responsive zu gestalten, sollten Sie

- die Breiten der Spalten in Angaben mit % oder em machen
- am besten mit den CSS Techniken Flexbox oder Grid-CSS arbeiten
- oben im head der HTML-Datei folgende Angabe hinzufügen:  

```
<meta name="viewport" content="width=device-width, initial-scale=1" >
```
- mit @media in den CSS Breakpoints hinzufügen
  - das heißt erst einmal in den CSS alle „normalen“ Angaben hinzufügen: Farben, Schriften, ein paar Abstände
  - dann @media (min-width: 768px) { CSS Angaben für größere Geräte, wie Mehrspaltigkeit etc. } usw
  - Arbeiten Sie mobile first (progressive enhancement, siehe unten)
- Testen Sie die Webseiten mit der Device Ansicht in Chrome



## 2 Layoutoptionen

<http://www.liquidaptive.com/>

1. **static Layout, fixed Layout: Feste Breite** – ein festes Pixelmaß wird angegeben, Elemente verschwinden außerhalb des sichtbaren Bereiches
2. **Fluid Layouts** (auch liquid Layouts genannt) – Breiten werden in Prozent und nicht in Pixeln angegeben. Breiten passen sich deshalb automatisch an. (Reicht allein nicht aus, um es an Smartphone etc. anzupassen); die Spalten werden immer schmaler
3. **Adaptive Layout** – Der Aufbau einer adaptiven Website richtet sich nach dem Viewport des jeweiligen Endgeräts. Dabei beschränkt man sich jedoch auf eine feste Anzahl von Breakpoints zwischen denen das Layout jeweils statisch bleibt. Eine fluide Anpassung an alle Auflösungen wie beim responsive Webdesign findet nicht statt.
4. **Responsive Layout** – Kombination aus fluid und adaptive, es werden flexible Breiten in Prozent oder em angegeben und mit Breakpoints gearbeitet
5. **Hybride Layouts** – Kombination der vorherigen Layouttypen. Z. B. Werbespalte mit fester Breite, Rest fluid.

### 3 Grundprinzipien beim mobilen Design

#### Graceful Degradation

Begins with the development on the desktop or in the powerful environment. Then it is developed on the tablet and continued for the smartphone. More and more content and features are removed here.

#### Progressive Enhancement

Begins with the development on the smartphone and continues to the desktop version. More and more functionality is added.

Wird nach dem Prinzip Progressive Enhancement entwickelt gilt „Mobile First“. Es sollte immer nach diesem Motto „Progressive Enhancement“ gearbeitet werden. Z. B. müssen dann Hintergrundbilder, die nur auf einem breiten Bildschirm zu sehen sind, erst geladen werden, wenn die Webseite breiter dargestellt wird.

### 4 Gerätepixel und CSS-Pixel

Gerätepixel verhalten sich so, wie Sie es von einem Pixel erwarten:

Wenn der Bildschirm 1024px breit ist, können Sie zwei 512px Elemente nebeneinander darin unterbringen.

CSS-Pixel kümmern sich um den sichtbaren Bereich innerhalb des Browserfensters. Das bedeutet, dass CSS-Pixel unter Umständen nicht genau den Gerätepixeln entsprechen.

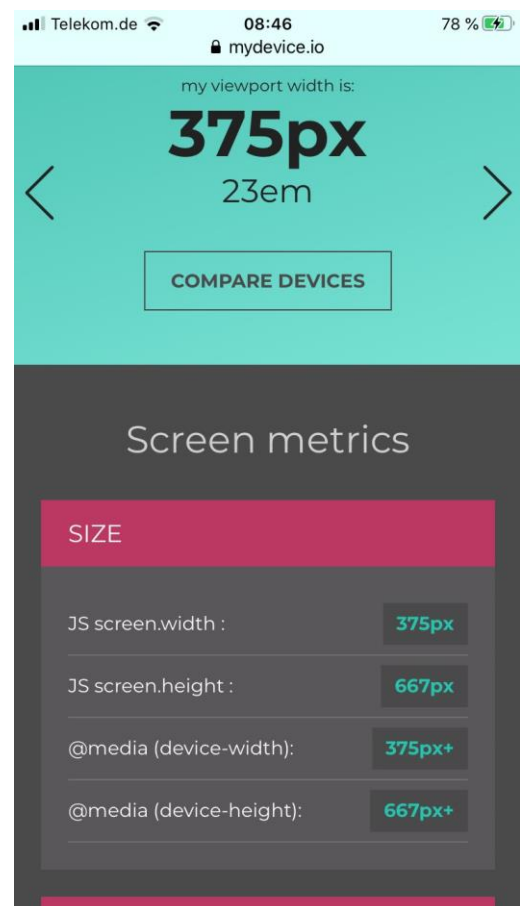
Auf vielen Geräten entspricht zwar ein CSS-Pixel genau einem Gerätepixel, aber auf vielen hochauflösenden Displays, wie etwa dem Retina-Displays des iPhones, entspricht ein CSS-Pixel in Wahrheit zwei Gerätepixeln.

Eine gute Webseite zum Testen ist die Webseite

<https://www.mydevice.io/>.

Beim Iphone 7 sieht man z. B. folgende Ansicht, das Gerät reagiert also auf den Breakpoint 375px.

Und mein gerade aktueller Laptop hat folgende Angaben:



Screen metrics			
SIZE	PIXEL RATIO	DENSITY	MISC
JS screen.width : 1280px	CSS pixel-ratio: 1	Resolution (dpi) : 96.00dpi	Root font size : 16px
JS screen.height : 720px	JS pixel-ratio : 1.0000	Resolution (dppx) : 1.00dppx	Orientation: landscape
@media (device-width): 1280px+		Resolution (dpcm) : 37.80dpcm	Device Aspect-Ratio : 1.78
@media (device-height): 720px+			

## 5 Viewport

Als Viewport (übersetzt etwa Sichtöffnung) wird je nach Anwendungsgebiet ein Ausschnitt eines Bildes, eines Videos, einer virtuellen oder realen Welt bezeichnet oder der für die Darstellung zur Verfügung stehende Bereich.

In der Programmierung wird der Anzeigebereich im Fenster der Anwendung, der für die Darstellung des Anwendungsinhaltes tatsächlich zur Verfügung steht, als Viewport bezeichnet. Ist der Inhalt breiter oder höher als der Viewport, werden Scrollbalken angezeigt.

So gilt etwa im Webdesign die Regel

„Bildschirm-Auflösung > Desktopgröße > Browserfenstergröße > Anzeigebereich (Viewport).“  
Inhalte außerhalb des Viewports sind aufgrund des Clippings nicht sichtbar.

### 5.1 Visueller und Layout-Viewport

Man spricht von zweierlei Viewports: den visuellen und den Layout-Viewport.

Der Layout Viewport ist insofern den Gerätepixeln ähnlich, als seine Maße immer dieselben bleiben, unabhängig von der Ausrichtung und Vergrößerung. Der visuelle Viewport variiert dagegen. Das ist der Teil der Seite, der tatsächlich auf dem Bildschirm angezeigt wird.

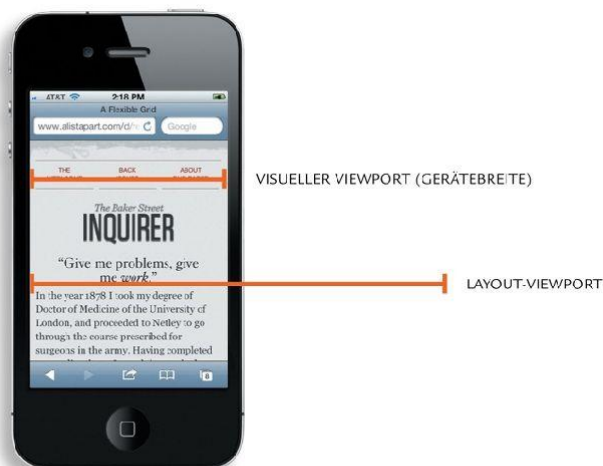


Abbildung 3.4 Mobile Geräte haben zwei verschiedene Viewports, die sich deutlich unterscheiden können.

## 5.2 Viewport-Tag und Eigenschaften

Mit dem Viewport-Metatag können wir die Skalierung und den Layout-Viewport vieler Geräte steuern.

```
<meta name="viewport" content="Direktive, Direktive " />
```

Eigenschaft mit möglichem Wert	Auswirkung
width=300px width=device-width	Die Breite des viewports wird auf die Gerätebreite gesetzt.
maximum-scale=1.0	Eine maximale Skalierung kann festgelegt werden
initial-scale=1.0	Die Seite soll beim Öffnen nicht skaliert werden (1.0) es können auch andere Skalierungen festgelegt werden
user-scalable=0 user-scalable=1	Das Skalieren durch den Nutzer wird unterbunden (0) – dies sollte aus Gründen der Barrierefreiheit vermieden werden.

Ein häufiger Grund dafür, dass Entwickler die Eigenschaften user-Scalable oder maximum-Scale verwenden, ist ein hartnäckiger Bug in iOS. (In iOS6 wurde dieser Fehler behoben.)

Wenn Sie für den Viewport einen Wert festlegen, bei dem der Benutzer die Seite skalieren kann, wird die Seite auf mehr als 100% skaliert, wenn Sie das Gerät ins Querformat drehen.

## 6 Übungen zum Viewport

⇒ Öffnen Sie eine von Ihnen erstellte Responsive Webseite im Editor und setzen Sie den viewport wie folgt:

⇒ `<meta name="viewport" content="width=600px, initial-scale=1;" />`

⇒ Öffnen Sie in Chrome und betrachten Sie mit dem Device Mode und stellen Sie eine Handygröße ein.

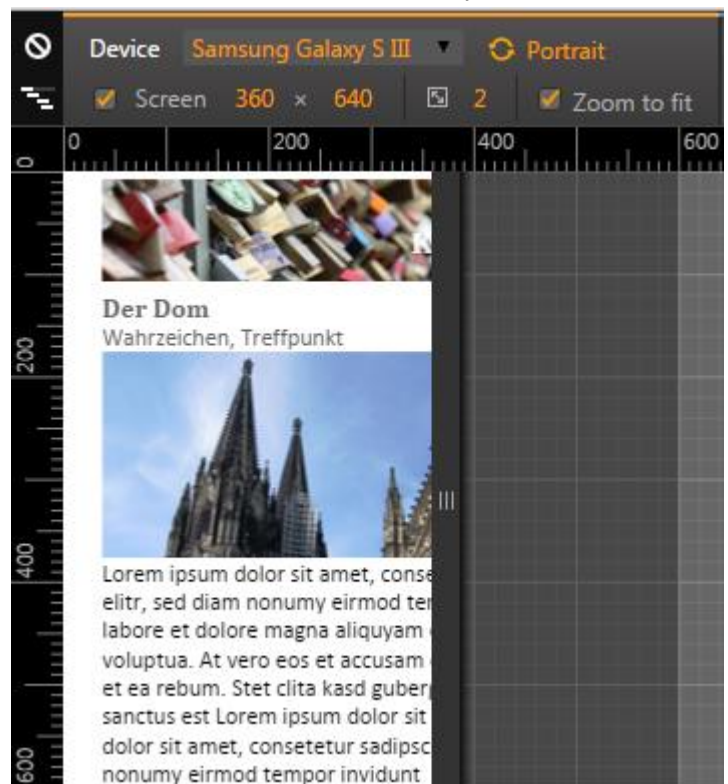
Sie können erkennen, dass die Seite nicht auf die Breite skaliert wird und der viewport breiter ist als die Breite des Gerätes. Dies ist nicht sehr günstig.

⇒ Ändern Sie auf `width=device-width` und betrachten Sie das Ergebnis.

Die Angabe `width=device-width` ergibt also Sinn, da wir die Breite des Gerätes nicht genau kennen.

⇒ Lassen Sie `initial-scale` einmal weg oder geben Sie einen anderen Wert an. Sie können damit die Anfangsskalierung festlegen.

⇒ Testen Sie auch die Angaben für `user-scalable` und `maximum` bzw. `minimum-scale`.



## 7 Media Angaben, Media Queries

Syntax allgemein:

a) Einfügen als Link auf Stylesheet

```
<link rel="stylesheet" type="text/css" media="screen" href="sans-serif.css">
<link rel="stylesheet" type="text/css" media="print" href="serif.css">
```

b) Einfügen mit @media in den CSS

Allgemein:

```
@media [not|only] Typ [and] (Ausdruck) {
  Regeln
}
```

Beispiel:

```
@media (min-width:768px) {
  /* hier alle Angaben für einen mittelgroßen Bildschirm
}

@media (min-width:992px) {
  /* hier alle Angaben für einen normalen Desktop

    body {background-image: url("paper.gif"); }

}
```

c) Verknüpfungen innerhalb media sind möglich

```
<link rel="stylesheet" media="screen and (color)" href="example.css"
/>
```

d) mit @import in den CSS

```
@import url(color.css) screen and (color);
```

### 7.1 Liste der Medientypen

Typ	Zielgerät
all	alle Geräte (Standard)
braille	Braille-Lesegerät
embossed	seitenweise Braille-Drucker
handheld	Handheld-Geräte (typischerweise kleiner Bildschirm und möglicherweise monochrome)
print	Druck oder Druckvorschau
projection	projizierte Darstellung
screen	farbiger Computerbildschirm
speech	Sprachsynthesizer
tty	Medien mit einem Buchstabenraster mit fester Zeichenbreite (Terminals oder Fernschreiber)
tv	Fernsehgeräte

## 7.2 Medienausdrücke / Media-Features

Mögliche Angabe	Beispiel	Beschreibung
width	<code>@media screen and (min-width: 400px) and (max-width: 700px) { ... }</code>	Breite des Anzeigebereichs eines Geräts. (eventuell Browserfenster)
height	Angaben in px oder cm	Höhe des Anzeigebereichs eines Geräts.
device-width	z. B. 320	Breite des Bildschirms des Ausgabegerätes.
device-height	z. B. 600	Höhe des Bildschirms des Ausgabegerätes
orientation		Werte: portrait landscape
aspect-ratio	z. B. 16/9	Verhältnis vom Breite zu Höhe (width/height)
device-aspect-ratio	<code>@media screen and (device-aspect-ratio: 16/9) { ... }</code> <code>@media screen and (device-aspect-ratio: 32/18) { ... }</code>	Verhältnis von device-width zu device-height
color	<code>@media all and (color) { ... }</code> <code>@media all and (min-color: 1) { ... }</code>	Beschreibt die Anzahl der Bits pro Farbe des Ausgabegerätes. Ist das Ausgabegerät kein Farbgerät ist der Wert 0.
color-index		Anzahl der Einträge in einer Farbtabelle des Ausgabegerätes.

monochrome	<pre>&lt;link rel="stylesheet" media="print and (color)" href="http://..." /&gt; &lt;link rel="stylesheet" media="print and (monochrome)" href="http://..." /&gt;</pre>	Schwarz/weiß-Gerät
resolution	<pre>@media print and (min- resolution: 300dpi) { ... }</pre>	Auflösung des Ausgabegerätes, Einheiten dpi und dpcm Dots per inch Dots per centimeter
scan	<pre>@media tv and (scan: progressive) { ... }</pre> <p>Value: progressive, interlace</p>	Für TV media types
grid		Abfrage, ob das Ausgabegerät ein Raster oder eine Bitmap ist. Rastergeräte sind z. B. tty- terminal oder Telefone mit einer festen Schriftgröße dann ist die Ausgabe 1