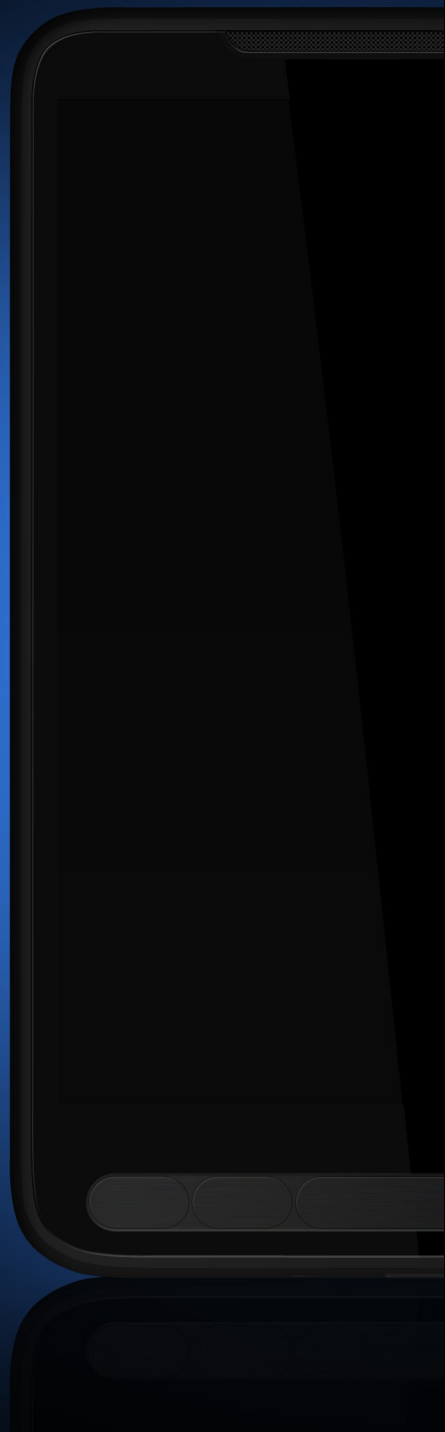
Interface-Optimierung bei mobilen Endgeräten



Darauf sollte man achten

Darstellung über CSS anpassen

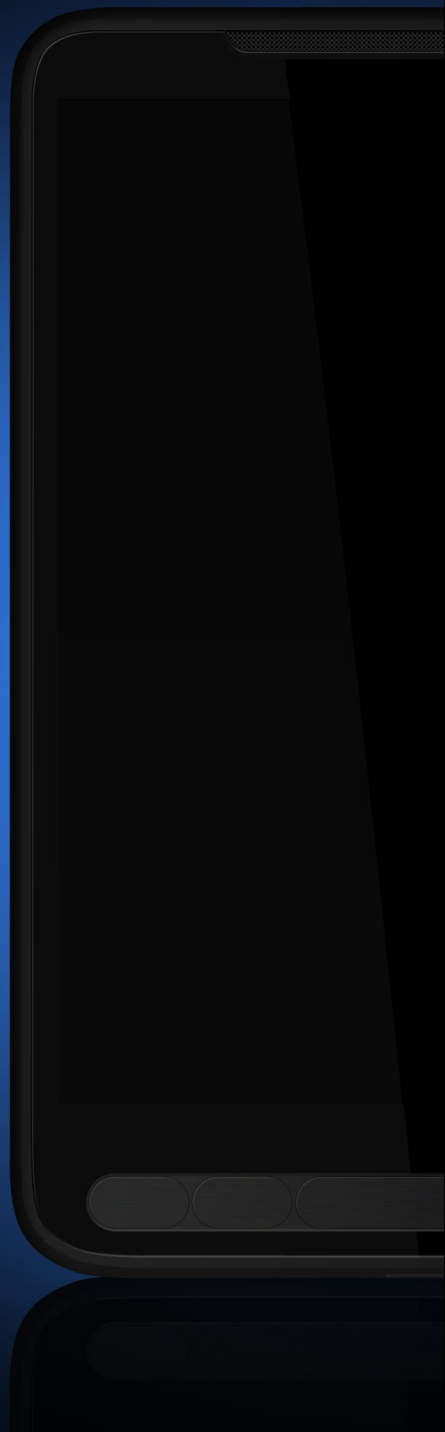
Durch optimierte Breiten kann man sehr einfach
für Mobiltelefon oder iPad optimierte Seiten ausliefern.

Auslieferung kleinerer/komprimierter Bilder

Minimiere damit die Menge der Datenübertragung.
Erreichen kann man das z.B. indem man die Bildqualität heruntersetzt.
Durch PHP-Lösungen, können Bilder automatisiert kompimiert werden.

Vermeiden Sie Flash zur Darstellung von Inhalten

Falls man dennoch nicht auf Flash verzichten will,
sollte man auch eine alternative Darstellung anbieten.



Darauf sollte man achten

Wichtige Funktionen nicht von Javascript abhängig machen

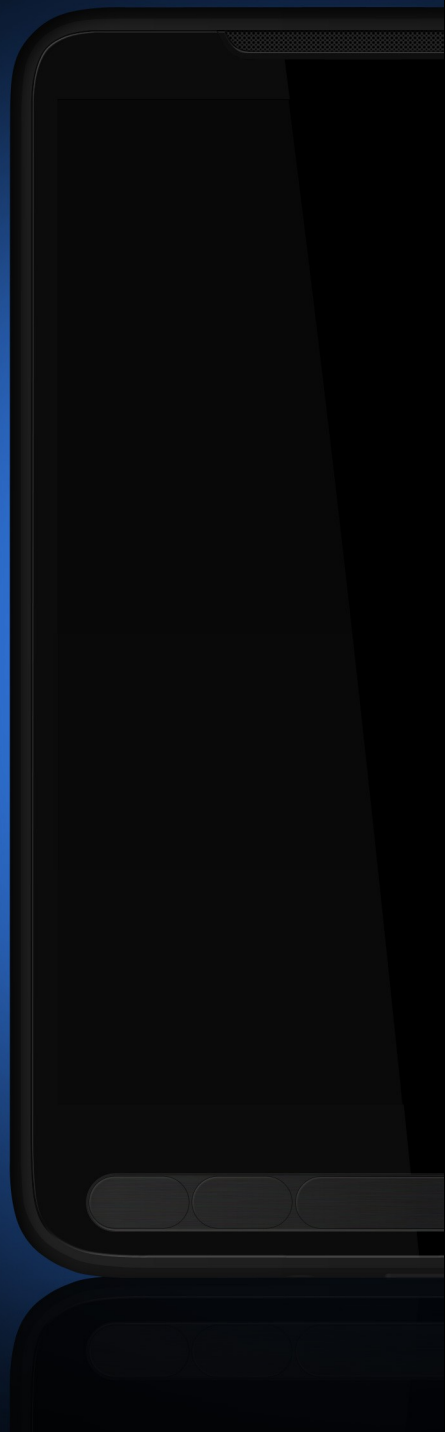
Formulare, Menüs und Inhalte,
die über Effekte nachträglich eingeblendet werden
(accordion, slides, fades) auch ohne JavaScript ermöglichen.

Pseudoklassen wie z.B. :hover nicht vorhanden

Auf mobilen Endgeräten gibt es bestimmte Pseudoklassen nicht.
Hover-Navigationen sind somit nicht umsetzbar.

Usern auch original Layout anbieten

Auch wenn man sich noch so viel Mühe bei einer mobilen Version gibt,
gibt es User, die gerne die originale Webseite besuchen würden. Diese darf man
nicht aussperren bzw. vergraulen durch stark vereinfachte mobile Designs.



Darauf sollte man achten

Verzicht auf negative Margins

Mithilfe von negativen Margins horizontal zentrierte Elemente werden auf kleinen Bildschirmen falsch dargestellt. Das bedeutet: bestimmte Bildbereiche sind nicht mehr erreichbar.

Daumen ist kein Mauszeiger

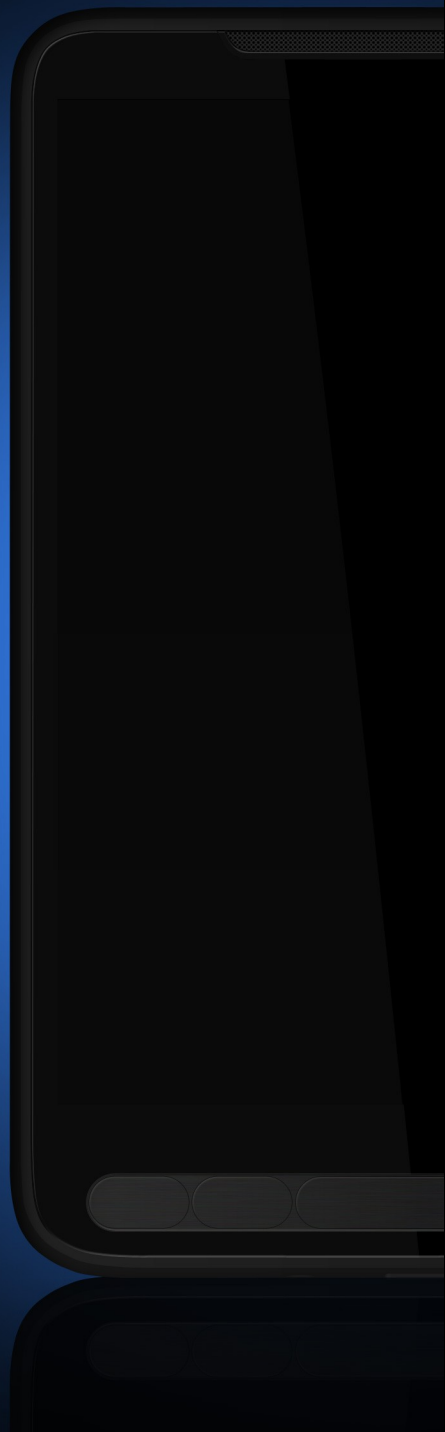
Ein Daumen hat nicht die Filigranität eines Mauszeigers.

Daher sollte man bei Buttons und Links auf ausreichende Dimensionen achten.

Beim Iphone sind gängige Größen etwa:

Daumengröße – 40x40px

Schriftgröße – 20px



Zusammengefasst

Design for One Web

Rely on Web standards

Stay away from known hazards

Be cautious of device limitations

Optimize navigation

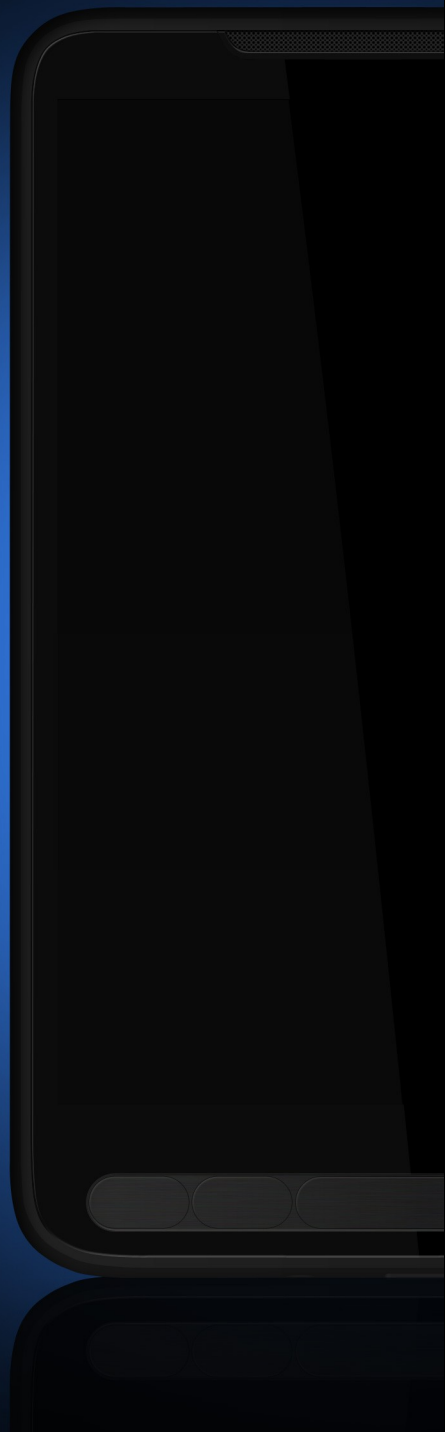
Check graphics & colors

Keep it small

Use the network sparingly

Help & guide user input

Think of users on the go



Die einfache Lösung

Unterschiedliche Stylesheets

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

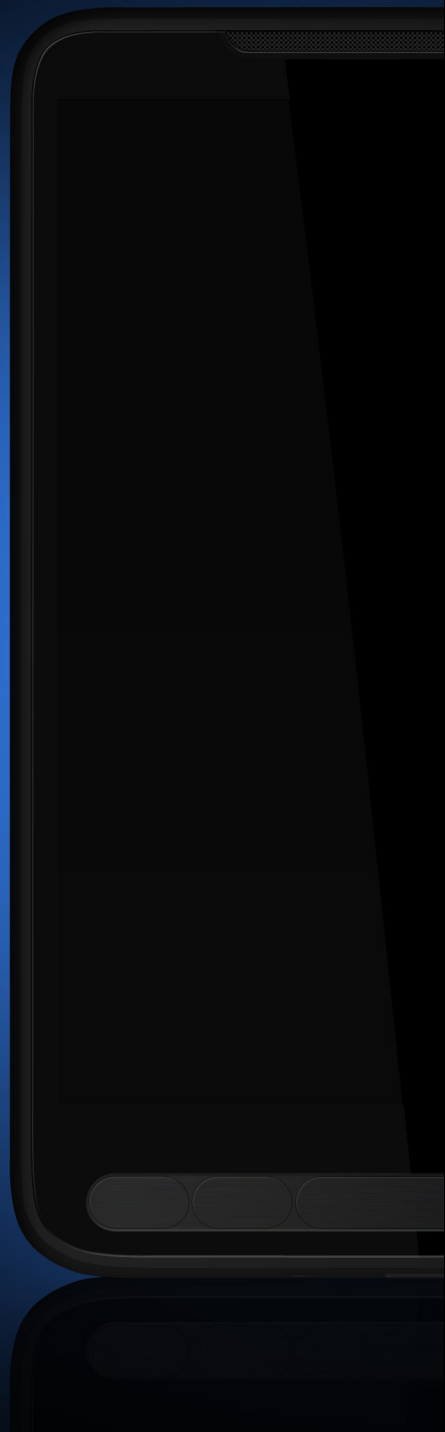
Vorteile

Schnell und ohne großen Aufwand realisierbar

Inhalte müssen nur einmal angelegt werden

Nachteile

Inhalte lassen sich nur bedingt anpassen



Einbindung von Stylesheets

Vorraussetzungen

Ein globales Stylesheet (*global.css*), in dem alle übergreifenden Definition, wie Schriftdefinitionen, allgemeine Parameter etc. vorgenommen werden. Für die Darstellung am Mobiltelefon kommt noch ein zusätzliches Stylesheet (*mobile.css*) dazu.

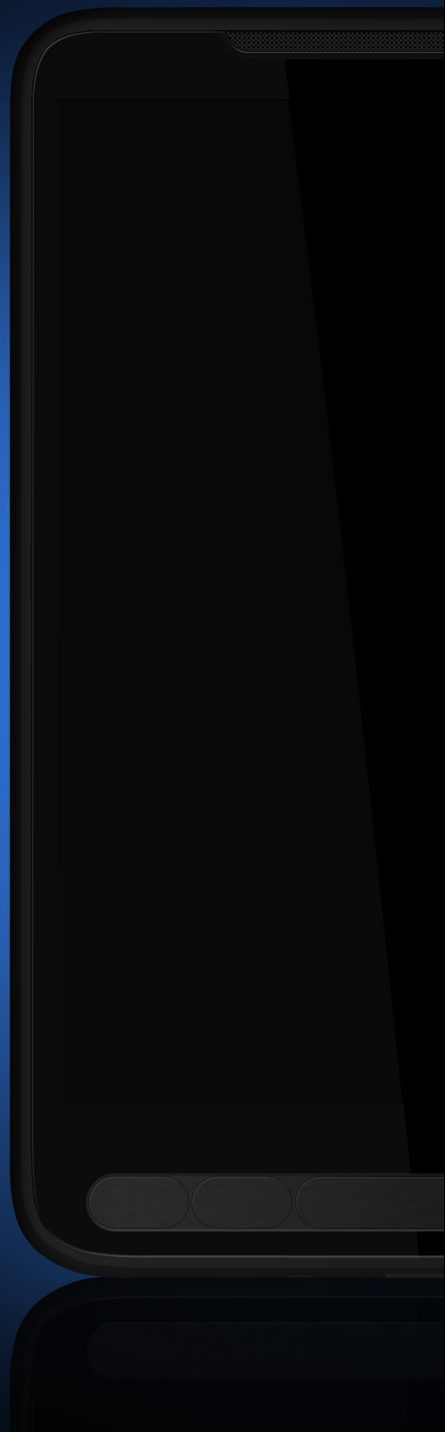
Einbindung des Stylesheets

Das Stylesheet für das Mobiltelefon wird mit einem speziellen media-Typ eingebunden:

```
<link rel="stylesheet" type="text/css"
media="only screen and (max-device-width: 480px)" href="mobile.css" />
```

Der Media-Typ ist hier: only screen and (max-device-width: 480px).

Das heißt, nur für die Bildschirmdarstellung auf Geräten, die eine maximale Bildschirmbreite von 480px haben.



Einbindung von Stylesheets

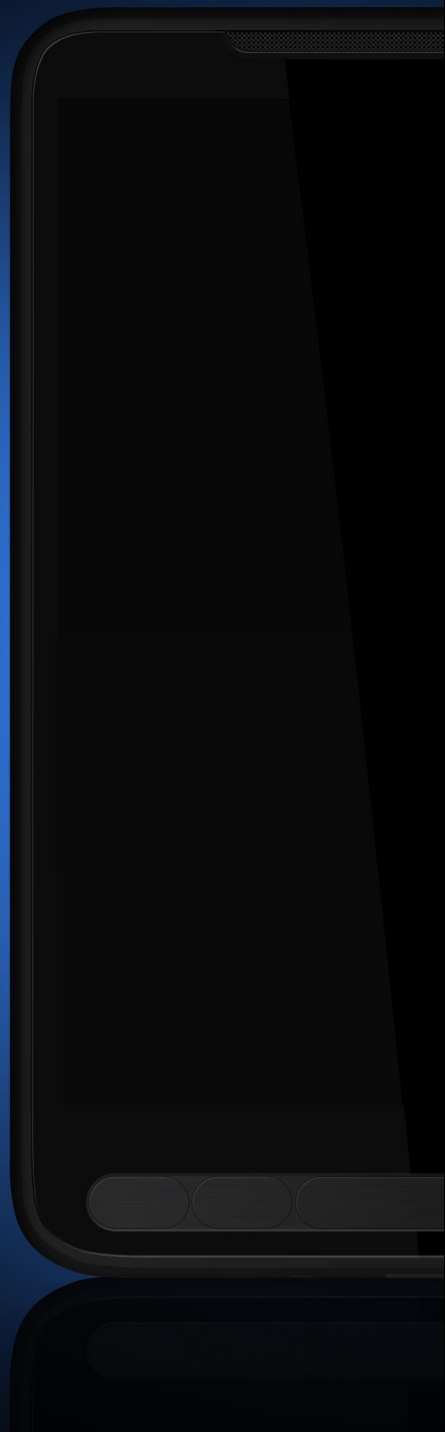
Problemfall Internet Explorer

Da ältere Internet Explorer (sprich Version 6 und 7) diesen Media-Typ noch nicht verstehen, muss dieses Stylesheet vor diesen Browsern "versteckt" werden. Dies kann man mit "Conditional Comments" machen:

```
<!--[if !IE]> -->
<link rel="stylesheet" type="text/css"
media="only screen and (max-device-width: 480px)" href="mobile.css">
<!--<![endif]-->
```

Relative Angaben statt absolute

Alle Größen innerhalb der Stylesheets bestenfalls in *em* angeben anstatt in *px*, so dass alle Maßangaben automatisch mit der Skalierung des Body-Tags angepasst werden.



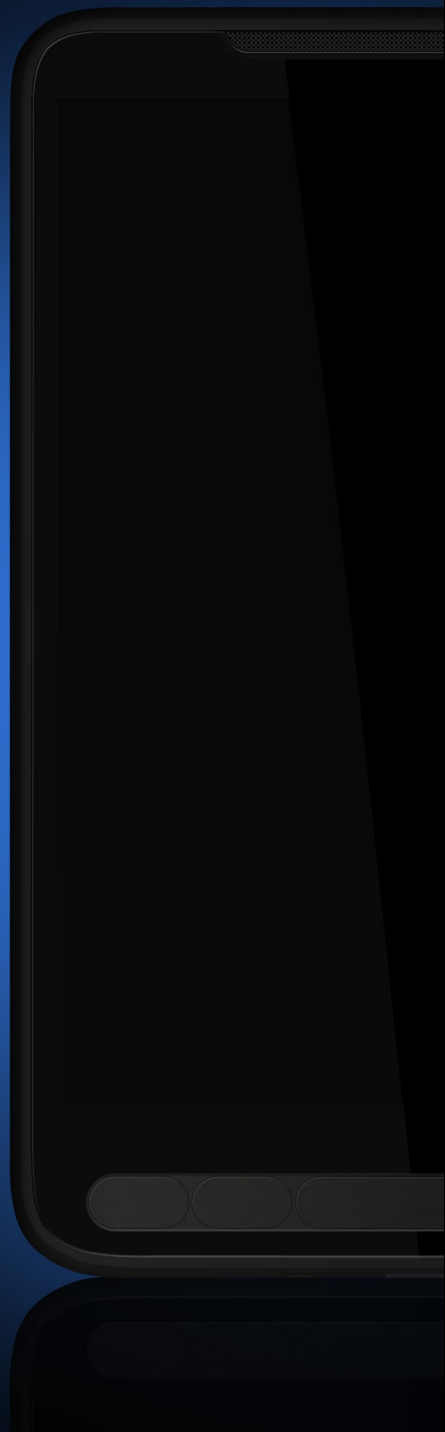
Einbindung von Stylesheets

Alternative Einbindung

Anstelle eines link-Tags mit der Adresse einer eigenen Stylesheet-Datei kann innerhalb der CSS-Datei eine @media-Regel notiert werden.

```
@media screen and (min-device-width: 481px)
{
   #box { float: left; }
}

@media screen and (max-device-width: 480px)
{
    #box { position: absolute; }
}
```

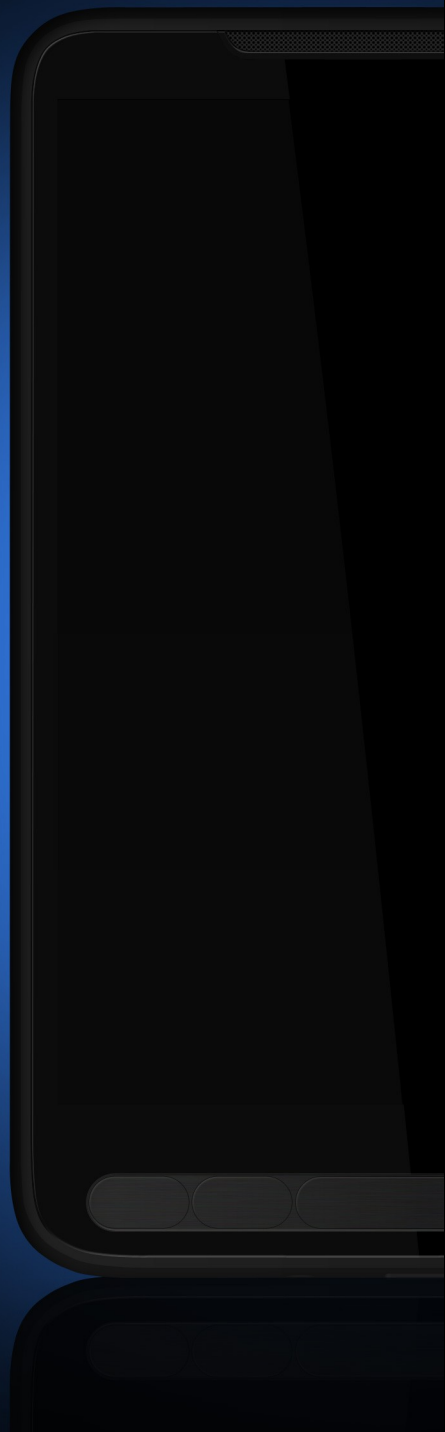


Die komplexere Lösung

spezielle Version für mobile Geräte

z.B. *http://m.sueddeutsche.de*

```
<?php
   $browser = get_browser(null, true);
   if ($browser["ismobiledevice"] == true) {
      echo "mobile Inhalte";
   } else {
      echo "normale Inhalte";
   }
?>
```



Vorteile

Layout kann von Grund auf entwickelt werden

User kann auch auf nicht-mobile Webseite

Nachteile

Eigene HTML-Dateien notwendig

Viewport

Auflösung ≠ Desktopgröße ≠ Browserfenstergröße ≠ Anzeigebereich (Viewport)

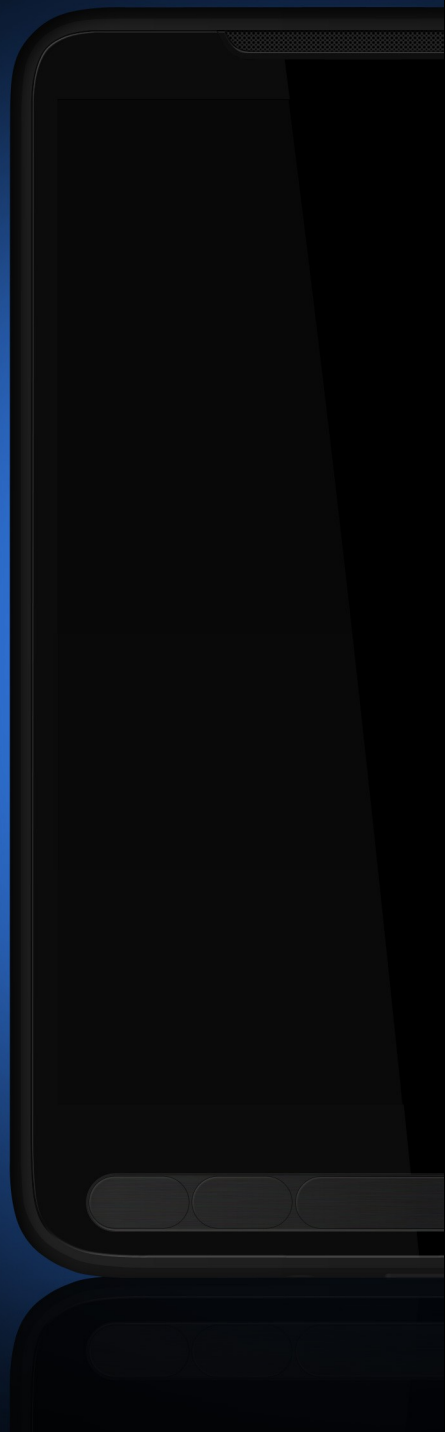
Es gibt noch eine zusätzliche Angabe im Head-Element der HTML-Datei, die man vornehmen kann, um die Darstellung am Mobiltelefon festzulegen:

```
<meta name="viewport" content="width=480" />
```

Viewport für Android und Opera Mobile

Um aber auch für Android und Opera Mobile eine bessere Anfangsskalierung zu bekommen, ist es nötig folgende Angabe zu verwenden — so skaliert der Viewport auf die Gerätebreite und bleibt dabei ungezoomt:

```
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1" />
```

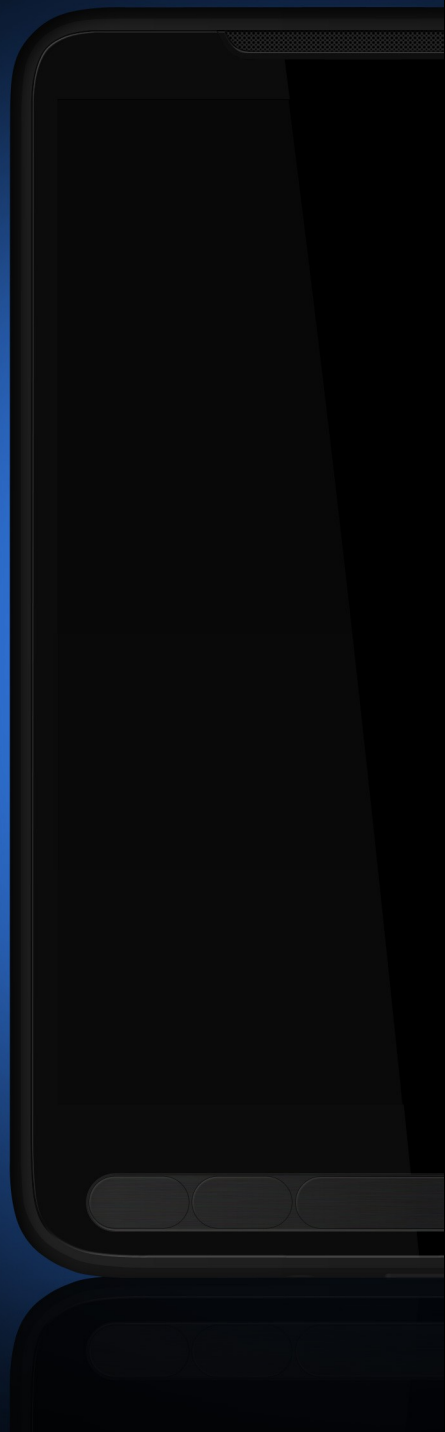


Viewport

Viewport für Android und Opera Mobile

Danach noch die automatische Textskalierung des Mobiltelefons deaktivieren,
da man die Textgrößen im Stylesheet ja konkret festlegen sollte:

```
/* {{{ global */html { -webkit-text-size-adjust: none;}/* }}} */
```



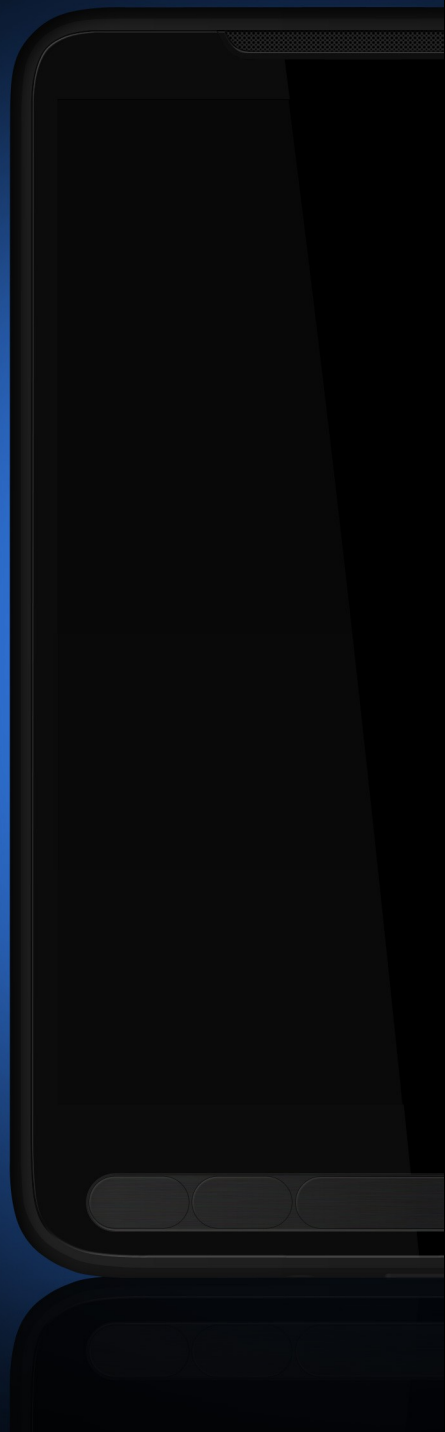
Favicons vs. Homescreen

Apple-Touch-Icon

Spezielle Icons, die auf dem Homescreen
des Geräts angezeigt werden, sind mit folgender Zeile möglich:

```
<link rel="apple-touch-icon" href="/icon.png"/>
```

Das Icon sollte in der Größe 60 × 60 Pixel und als PNG abgepeichert werden.



Weitere Technologien

CSS Mobile

Eine speziell auf den mobilen Einsatz angepasste CSS-Sprache.

SVG Tiny

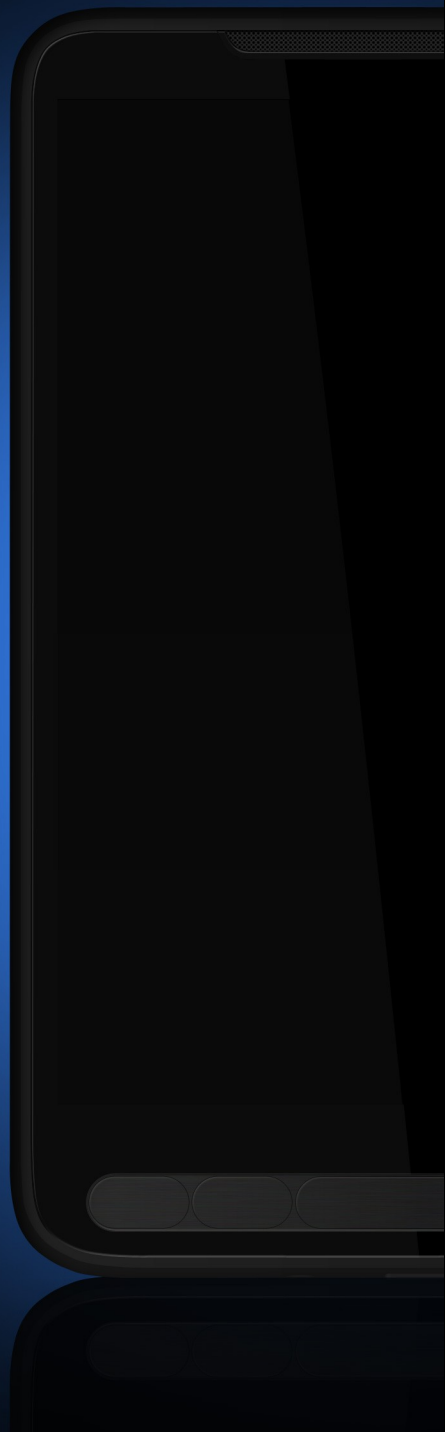
Ein skalierbares Vektorgrafik-Format,
angepasst an die Möglichkeiten von mobilen Endgeräten.

XHTML For Mobile

Definiert ein XHTML-Format für Mobilgeräte.

Adaptive Images http://adaptive-images.com/

Wurde entwickelt, um die Dateigröße von Bildern in responsiven Webdesigns zu verringern. Mit Hilfe der PHP- und Javascript-Lösung kann man erreichen, dass von allen Bilder auf einer Webseite automatisch verschiedene kleinere Bildformate erstellt werden. Diese, in Format und Dateigröße reduzierten Bilder, werden dann genutzt, wenn ein Besucher deiner Webseite z.B. auf einem Smartphone aufruft.



Testing

Test am Gerät

Die beste Lösung zum Testen, ist es, direkt am Gerät zu testen – nur dort können z.B. Gesten getestet werden. Da aber oftmals nicht nur ein Gerät Zielgruppe ist, sondern oft mehrere, mit unterschiedlichen Auflösungen, ist dies schwer umsetzbar.

MobileOK http://validator.w3.org/mobile/

Ein vom W3C-Konsortium entworfener Test, der Webseiten auf deren Mobiltauglichkeit prüft. Anschließend bekommt man erläuterte und gut dokumentierte Ergebnisse.

Smartphone-Emulatoren

Um sich bereits einen ersten Eindruck beim Designen und Coden zu verschaffen, ist es hilfreich, sich Emulatoren zu bedienen. Diese können zumindest den Bildausschnitt und teilweise GUI-Elemente simulieren bzw. darstellen.