

RESPONSIVES WEBDESIGN

Ein Problem beim Entwurf von Internetseiten ist die unterschiedliche Bildschirmgröße verschiedener Geräte. Bestimmt ist Ihnen schon aufgefallen, dass sich manche Webseiten an die Bildschirmgröße anpassen: So etwas nennt man **Responsives Webdesign**.

Beispiel:

PC-Monitor



Smartphone



Ausgangssituation:

Ihnen liegt bereits eine Webseite vor, bei der Sie ein Responsives Webdesign „nachrüsten“ sollen. Je nach Bildschirmgröße soll sich das Design der Webseite so anpassen:

Bildschirm-Breite < 40em




Bildschirm-Breite zwischen 40em und 60em



Bildschirm-Breite > 60em



	<p><u>Was genau ist denn jetzt dieses em?</u></p> <p>Früher hat man versucht, die Inhalt einer Webseite anzupassen, indem man abfragt, wie viele Bildpunkte (Pixel) ein Bildschirm darstellen kann. Die Bildschirme (z.B. von Smartphones) sind aber immer hochauflösender geworden (haben also mehr Bildpunkte). Dadurch kann es im Extremfall passieren, dass ein Smartphone sogar mehr Bildpunkte hat als ein älterer Monitor. Die Unterscheidung über die Bildpunkte funktioniert also inzwischen nicht mehr gut.</p> <p><u>Die Lösung:</u></p> <p>Die Einheit em (ausgesprochen wie der Buchstabe M) gibt ungefähr an, wie breit der Buchstabe M auf einem Bildschirm dargestellt wird. Das gibt uns zum Beispiel die Möglichkeit zu sagen: Wenn weniger als 40 mal der Buchstabe M auf einen Monitor passt, dann ist das ein kleiner Bildschirm!</p> <p><i>(Die Einheit em ist also keine absolute Größe! Der Wert von em hängt also von der gewählten Schriftart ab!)</i></p>
---	---

Mobile First!

Um ein Responsives Webdesign zu erreichen, legen wir im **Stylesheet** zuerst fest, wie die Anzeige für die **kleinste Bildschirmgröße** sein soll.

(Diese Reihenfolge ist nicht zwingend, ist aber sehr sinnvoll! Auf Mobilgeräten kann so Datenvolumen gespart werden, wenn z.B. grafische Elemente für größere Bildschirme gar nicht geladen werden.)

```
body{
  display: grid;
  grid-template-columns: 1fr;
  grid-template-areas:
    "head"
    "nav"
    "main"
    "aside"
    "foot";
}
```

Danach nehmen wir die Einstellungen für alle anderen Tags vor (z.B. <h1>), die für die kleinste Bildschirmgröße gelten sollen.

Darstellung für weitere Bildschirmgrößen

Am Ende des Stylesheets legen wir dann fest, was bei anderen Bildschirmauflösungen anders formatiert werden soll. Hier im Beispiel für Bildschirmgrößen **ab 40em**:

```
@media (min-width: 40em)
{
  body{
    grid-template-columns: repeat(2, 1fr);
    grid-template-areas:
      "head head"
      "nav aside"
      "main main"
      "foot foot";
  }

  h1 {
    font-size: 40px;
  }
}
```



Medienabfragen (Media Queries)

Was wir oben tun, nennt man eine **Medienabfrage**. Den Begriff müsste man sich merken können, schließlich fragen wir damit ja das Medium, also die Art des Geräts ab. Der englische Begriff dafür ist Media Query (Mehrzahl: Media Queries).

Warum funktioniert das auf Mobilgeräten noch nicht?

Das funktioniert alles super, wenn man das ganze testet, indem man das Browserfenster eines PCs in der Größe ändert. Auf einem Smartphone wird das Ergebnis in der Regel aber schwer lesbar sein.

Das heißt: Die Schrift wird zu klein sein! Das Problem lösen wir, indem wir in der HTML-Datei festlegen, dass die Schriftgröße so sein soll, wie auf den größeren Displays:

HTML-Seite:

```
<html>
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Hinweis: Unter *Viewport* versteht man die Größe des Browser-Fensters!

Aufgabe

- Öffnen Sie die den Ordner *homepage_10_Responsives_Webdesign-AUFGABE*.
- Realisieren Sie bei den dort bereits entworfenen Webseiten ein Responsives Webdesign wie oben gefordert (3 verschiedene Bildschirmgrößen).
- Außerdem soll sich die Größe der Überschriften (<h1>) an die Bildschirmgröße anpassen:

Bildschirmgröße	Größe von <h1>
< 40em	30px
< 60em	40px
Ab 60em	50px