

CSS statt LESS und SASS

Das Styling einer Webseite über Cascaded Stylesheets (CSS) ist der beste Weg, um eine gute Trennung zwischen der Struktur und dem Aussehen zu erhalten. Die Sprache CSS wird durch das W3C beständig weiterentwickelt, um die Nutzung angenehmer zu machen und die Gestaltungsspielräume zu vergrößern.

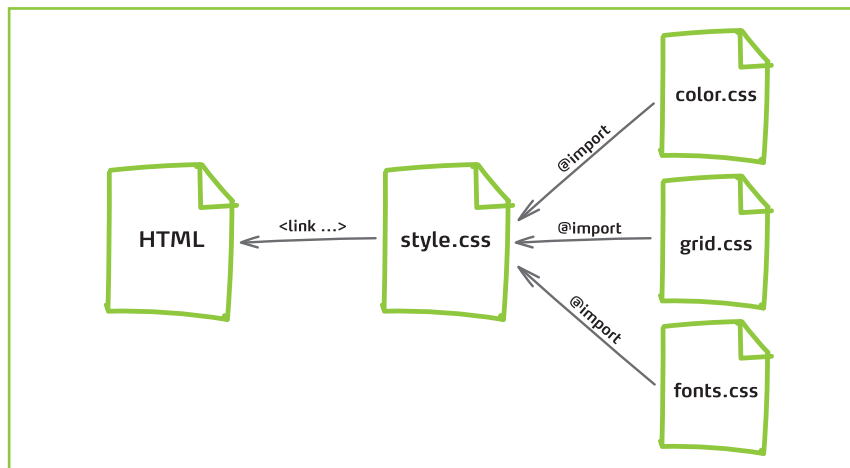
Neuere Stylesheet-Sprachen, wie zum Beispiel SASS oder LESS, können die Arbeit mit CSS vereinfachen, da sie die Sprache entwicklerfreundlich erweitern. Nachteil dieser Sprachen ist, dass sie meist noch vor der Verwendung in CSS-Code kompiliert werden müssen. Um diese Lücke zu schließen, hat das W3C (World Wide Web Consortium) begonnen, den CSS-Standard so zu erweitern, dass einige der Features in CSS direkt einfließen.

Berechnung von Werten

Die Spezifikation "CSS Values and Units Module Level 3" [1] definiert eine Vielzahl neuer Einheiten. Zum Beispiel die Viewport-spezifischen Einheiten "vw" und "vh". Diese Einheiten sind dynamisch und bedeuten 1 Prozent der aktuellen Viewport Höhe bzw. Breite. Elemente können damit zum Beispiel in Abhängigkeit zur Browsergröße skaliert werden.

Abbildung 1:

Importieren von CSS-Dateien



Neben den neuen Einheiten wurde aber auch eine neue Funktion vorgestellt: `calc()`. Damit können Werte für CSS-Eigenschaften über Einheitsgrenzen hinweg berechnet werden. Es ist also möglich, relative und absolute Werte miteinander zu verrechnen:

```
div {  
  width: calc(100% - 50px);  
}
```

Für die Berechnung können Terme in beliebiger Länge genutzt werden. Als Rechenoperationen stehen Addition (+), Subtraktion (-), Multiplikation (*) und Division (/) zur Verfügung. Berechnet werden können allerdings nur Zahlenwerte, keine Farben, wie das mit LESS/SASS möglich ist. Die Verwendung von `calc()` ist in allen verbreiteten Browsern möglich [2].

Endlich CSS importieren

Mit der W3C Spezifikation "CSS Cascading and Inheritance Level 3" [3] wird die Regel `@import` eingeführt. Wenig überraschend können damit CSS-Dateien in andere Stylesheet-Dokumente eingebunden werden (siehe Abbildung 1).

Dies ermöglicht zum einen die bessere Strukturierung von CSS-Regeln. Zum anderen können medienspezifische Stylesheets nur bei Bedarf geladen werden. Die Verwendung von `@import` ist einfach:

```
@import "themes/my-theme.css";
@import url("mobile.css") handheld and
  (orientation: landscape);
```

Fakultativ ist die Nutzung von `url()` zur Adressierung der Zielfile und ändert nichts an der gezeigten Verwendung.

Durch das Erweitern der Import-Regel mit einer Media-Query ist die Auswertung der enthaltenen Regeln an Bedingungen geknüpft. Konkret heißt das: die CSS-Regeln in `mobile.css` werden nur dann angewendet, wenn das aufrufende Gerät vom Type "handheld" ist und im Querformat genutzt wird. Das gleiche Prinzip wird auch von Media-Queries verwendet. Der Import mit der CSS-Regel `@import` kann von allen verbreiteten Browsern angewendet werden [4].

Mit Variablen, Werte zentral definieren

Lange Zeit konnten in CSS keine Variablen verwendet werden. Dieser Umstand führt dazu, dass sich z. B. die Farbcodes von verwendeten Farben an vielen Stellen doppeln. Das kann zwar durch geschicktes Definieren von Regeln vermindert werden, aber in der Praxis ist es kaum ganz zu vermeiden. Mit Variablen kann eine Farbe zentral definiert und vor allem geändert werden. Der entsprechende Entwurf vom W3C heißt "CSS Custom Properties for Cascading Variables Module Level 1" [5]. Die Custom Properties aus dem Titel des Dokuments sind der Schlüssel zur Verwendung von Variablen. Wie der Name schon sagt, wird eine eigene CSS-Eigenschaft mit einem Wert definiert. Der Wert dieser vom Entwickler

definierten CSS-Property kann dann mithilfe des Variablenmoduls an weitere CSS-Eigenschaften übergeben werden:

```
:root {
  --main-color: #8db72c;
}
#content {
  background-color: var(--main-color);
}
```

Für die Verwendung von Custom Properties gelten folgende Regeln:

- Der Name der Eigenschaft muss mit zwei Minuszeichen beginnen.
- Die Namen sind case sensitive. `--color` und `--color` sind also unterschiedliche Eigenschaften.
- Zyklische Abhängigkeiten zwischen Custom Properties sind verboten.

Nachfolgend sieht man, dass die Custom Properties tatsächlich Variablen sind und keine Konstanten:

```
#content { --color: green; }
#first { --color: red; }
.bg{ background-color: var(--color); }
<div id="content">
  <div id="first" class="bg">
    <!--roter Hintergrund -->
  </div>
  <div id="second" class="bg">
    <!--grüner Hintergrund -->
  </div>
</div>
```

Die Property `--color` wird für das Element mit der ID `content` auf `green` gesetzt. Damit ist diese Property innerhalb aller Elemente dieser Kaskade

nutzbar. Für das Element mit der ID `first` wird der Wert von `--color` ersetzt, so dass der Hintergrund dieses `div`-Elements rot ist und nicht grün.

Die `var`-Funktion, welche genutzt wird, um den Wert der Custom Property auszulesen, kann um einen zweiten Parameter erweitert werden (z. B. `var(--color, blue)`). Dieser zweite Parameter ist der Fallback-Wert, der genutzt wird, wenn die Custom Property nicht definiert ist.

Die Kombination von Variablen und Berechnung von Werten mit `calc()` ist zwar in der Spezifikation vorgesehen, funktioniert in der Praxis aber noch sehr mäßig. Je nach verwendeter Rechenart muss die Einheit in der Variable definiert (Addition, Subtraktion) oder darf nicht angegeben werden (Multiplikation). Die Division funktioniert mit keiner Variante und in keinem der Browser, die getestet wurden. Die CSS Variablen werden im Moment sowieso leider noch nicht von jedem Browser unterstützt. Die Microsoft Browser bieten erst ab Edge 15 eine partielle Unterstützung. Alle anderen Browser großer Hersteller haben dieses Feature prinzipiell schon eingebaut [6].

CSS zieht nach

Mit den im Artikel gezeigten Spezifikationen kann CSS ein bisschen zu LESS und SASS aufschließen. Auch wenn der Browser-Support noch nicht ganz zufriedenstellend ist, wird hier in die richtige Richtung gegangen. Durch eine bessere Strukturierung und das Definieren von Variablen wird es einfacher CSS zu pflegen. Leider setzen noch nicht alle großen Browser diese wichtigen Spezifikationen vollständig um. Ich bin jedoch zuversichtlich, dass demnächst die uneingeschränkte Verwendung von `calc()`, `@import` und `var()` möglich sein wird.

Die Ankündigungen der Browser-Hersteller deuten darauf hin, dass SASS und LESS demnächst nicht mehr zwingend notwendig sein werden, um mit CSS zu arbeiten.

■ Peter Herklotz

Links

- [1] <https://www.w3.org/TR/css3-values>
- [2] <http://caniuse.com/#feat=calc>
- [3] <http://www.w3.org/TR/css-cascade-3/>
- [4] https://developer.mozilla.org/de/docs/Web/CSS/@import#Browser_Kompatibilität
- [5] <https://www.w3.org/TR/css-variables-1/>
- [6] <http://caniuse.com/#feat=css-variables>



Peter Herklotz ist Berater bei der buschmais GbR. Seine fachlichen Schwerpunkte liegen in der Konzeption und Entwicklung neuer, sowie Erweiterung bestehender Java-Applikationen.