21.4 Exkurs zu CSS-Präprozessoren

Wenn Sie sich näher mit CSS-Frameworks befassen, werden Sie vermutlich des Öfteren über Begriffe wie *Less* oder *Sass* stoßen. Wenn Sie z. B. das Bootstrap-Framework anpassen wollen, müssen Sie die Less-Variablen anpassen. Um es kurz zu machen, die beiden Stylesheet-Sprachen sind kein Hexenwerk und im Grunde nur dazu da, Ihnen das Leben mit CSS zu erleichtern. Beide (Less und Sass) sind *CSS-Präprozessoren*. Damit hier kein falscher Eindruck entsteht: Die Stylesheet-Sprachen Less und Sass werden nicht nur für CSS-Frameworks verwendet. Beide können Sie auch in Ihren eigenen Webprojekten einsetzen.

Wozu soll das gut sein, fragen Sie sich? Mit CSS allein können Sie z. B. keine zentralen Variablen festlegen und diese so an mehreren Stellen wiederverwenden. Auch eine Schachtelung von Regeln kann mit reinem CSS nicht realisiert werden, womit sich eine Menge Codewiederholungen vermeiden ließe. Mit Less und Sass ist das möglich. Ebenfalls interessant ist die Möglichkeit, dass Sie mit ihnen Regeln unter einem Namen zusammenfassen und an mehreren Stellen wiederverwenden können. Dies wird als *Mixins* bezeichnet, und damit lassen sich ebenfalls unnötige Codewiederholungen vermeiden.

Um einen solchen Präprozessor zu verwenden, schreiben Sie eine Datei in der Syntax der von Less oder Sass. Ein Webbrowser kann damit noch nichts anfangen. Dafür sorgt ein CSS-Präprozessor, der die Datei mit der Less- oder Sass-Syntax versteht und in eine normale CSS-Datei umwandelt. Sie sehen das Prinzip vereinfacht in Abbildung 21.25.

In den nächsten Abschnitten erhalten Sie einen kurzen Überblick darüber, wie die Syntax für die Stylesheet-Sprache Less aussieht. Hier wird nur auf Less (und ganz kurz auf Sass) eingegangen, aber es gibt noch eine Reihe weiterer Stylesheet-Sprachen mit einem CSS-Präprozessor wie z. B. Stylus, Turbine, CSS Crush oder PCSS, um nur einige zu nennen.

Sie dürfen in diesem Buch keine umfassende Einführung in die Stylesheet-Sprachen erwarten. Hier werden lediglich ein paar ihrer wichtigsten Features vorgestellt, um zumindest zu

verstehen, wie Sie entsprechende Features in den CSS-Frameworks verwenden können. Wollen Sie ganze Projekte mit einem Präprozessor wie Less oder Sass realisieren, müssen Sie die Dokumentation auf der jeweiligen Projektseite der Stylesheet-Sprache studieren.

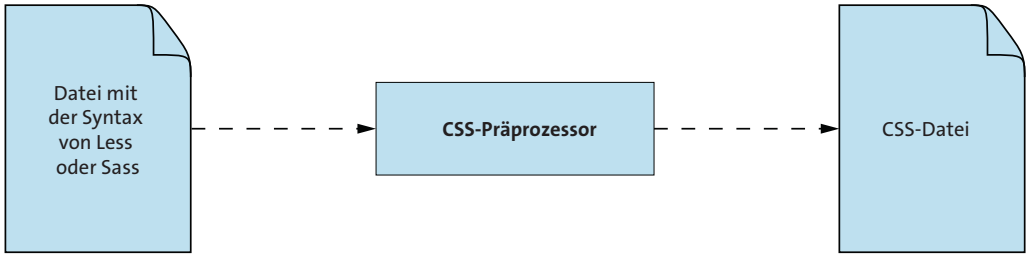

Abbildung 21.25 Von Less oder Sass über den CSS-Präprozessor zur gewöhnlichen CSS-Datei

21.4.1 Die Stylesheet-Sprache Less

Less ist in JavaScript geschrieben und kann daher sowohl server- als auch clientseitig verwendet werden. Um die Stylesheet-Sprache selbst testen zu können, müssen Sie das entsprechende JavaScript in das HTML-Dokument einbinden. Um die folgenden Beispiele im Webbrowser zu testen, können Sie daher *less.js* zunächst von der Webseite *http://lesscss.org* herunterladen und folgendermaßen einbinden:

```
…
<link rel="stylesheet/less" type="text/css" href="styles.less">
<script src="less-1.7.0.min.js"
        type="text/javascript"></script>
…
```

Listing 21.19 /Beispiele/Kapitel021/21_4_1/index.html

Die clientseitige Verwendung von Less ist nur geeignet, um damit während der Entwicklung zu arbeiten. In der Praxis sollten Sie die serverseitige Verwendung vorziehen, weil diese zuverlässiger ist und wesentlich effizienter.

Serverseitige Verwendung von Less

In der Praxis sollten Sie die serverseitige Verwendung von Less bevorzugen. Gewöhnlich wird dazu der Weg über *Node.js* (*http://nodejs.org*) der Standard-Paketverwaltung *npm* (*www.npmjs.com*) genommen, um Less als Kommandozeilen-Tool wie folgt zu installieren:

```
$ npm install less
```

Am häufigsten in Verwendung (auch in den CSS-Frameworks) mit Less finden Sie die Variablen und Mixins. Diese werden hier kurz in der Praxis demonstriert.

Variablen mit Less

Variablen können in Less mit einem @ eingeleitet und dann überall über diesen Variablennamen verwendet werden. Hierzu ein einfaches Beispiel:

```
…
@blau: blue;
@weiss: #FFF;
@rot: rgb(255,0,0);

header, footer {
   background-color: @blau;
   color: @weiss;
}
h1 {
   color: @blau;
}
p {
   color: @rot;
}
…
```

Listing 21.20 /Beispiele/Kapitel021/21_4_1/style.less

Anhand der Variablennamen lässt sich häufig einfacher erkennen, worum es sich bei diesem Wert handelt. Der Hauptvorteil ist eher, dass die Werte, wenn diese mehrfach verwendet werden, an einer zentralen Stelle geändert werden können und nicht irgendwo in der Datei verteilt gesucht werden müssen.

Der Präprozessor wandelt die eben abgedruckten Zeilen der Datei *styles.less* in folgenden CSS-Code um:

```
header, footer {
   background-color: blue;
   color: #FFF;
}
h1 {
   color: blue;
}
p {
   color: rgb(255,0,0);
}
```

Mixins mit Less

Oftmals verwenden Sie ein ganzes Bündel von zusammengehörenden CSS-Anweisungen immer wieder an unterschiedlichen Stellen, was dann wiederum zu häufigen Codewieder-

holungen führt. Mit *Mixins* können Sie das Bündel solcher CSS-Anweisungen mit Eigenschaften zusammenfassen und dann durch einen Bezeichner aufrufen. Innerhalb von Mixins können wiederum Less-Variablen verwendet und, falls nötig, mit Standardwerten versehen werden. Hierzu ein einfaches Beispiel mit Mixins und Less-Variablen:

```
…
@blau: blue;
@weiss: #FFF;
@rot: rgb(255,0,0);

.flaeche(@rahmen: 1px, @farbe1: black, @farbe2: @weiss) {
  border: @rahmen solid @farbe1;
  background-color: @farbe2;
}

.abstand(@polsterung: 5px) {
  padding: @polsterung;
  margin: 10px 0px;
}

header {
  color: @blau;
  .flaeche;
  .abstand;
  text-align: center;
}
footer {
  color: @weiss;
  .flaeche(2px, blue, red);
  .abstand(3%);
  text-align: center;
}
…
```

Listing 21.21 /Beispiele/Kapitel021/21_4_1/styles.less

Der Less-Präprozessor wandelt diese Less-Mixins in folgende CSS-Zeilen um:

```
header {
  color: blue;
  border: 1px black #FFF;
  background-color: #FFF;
  padding: 5px;
```

```
  margin: 10px 0px;
  text-align: center;
}
footer {
  color: #FFF;
  border: 2px blue red;
  background-color: red;
  padding: 3%;
  margin: 10px 0px;
  text-align: center;
}
```

Verschachtelung

Ein weiteres komfortables Konzept ist die Verschachtelung, womit Selektoren quasi ineinander verschachtelt werden können, wodurch weniger geschrieben werden muss und der eine oder andere Fehler vermieden werden kann.

Der folgende Ausschnitt demonstriert eine solche Verschachtelung in der Praxis:

```
…
#lead_article {
  h1 {
    font-style: italic;
    color: black;
  }
  p {
    font-size: 1.1em;
    color: darkgray;
  }
  a {
    text-decoration: none;
    font-weight: bold;
    &:hover { background-color: silver;}
  }
}
…
```

Listing 21.22 /Beispiele/Kapitel021/21_4_1/styles.less

Der Less-Präprozessor wandelt diese Verschachtelung in folgendes CSS um:

```
#lead_article  h1 {
  font-style: italic;
  color: black;
```

```
}

#lead_article  p {
  font-size: 1.1em;
  color: darkgray;
}

#lead_article  p a {
  text-decoration: none;
  font-weight: bold;
}

#lead_article p a:hover { background-color: silver;}
```

Weitere Funktionen mit Less

Less kann noch weitaus mehr, als hier mit den Variablen und Mixins kurz demonstriert wurde. Weitere Funktionen wären z. B. Namenräume, Funktionen für mathematische Berechnungen oder Farbanpassungen und noch vieles mehr. Einen umfassenden Überblick und eine Einführung in alle Features und Funktionen von Less bietet die Projektseite *http://lesscss.org*.

21.4.2 Die Stylesheet-Sprache Sass

Alles eben Beschriebene in Verbindung mit Less gilt auch für Sass. Nur die Syntax von Sass ist etwas anders als die von Less. Während Sie bei Less die Variablen mit `@variable` verwendet haben, müssen Sie bei Sass `$variable` notieren, wie das folgende Sass-Beispiel im Vergleich zu einem Less-Beispiel zeigt:

Sass	Less

`$farbe: blue;`
`#main p {`
`  color: $farbe;`
`}`
	`@farbe: blue;`
	`#main p {`
	`  color: @farbe;`
	`}`

Tabelle 21.5 Variablen von Sass im Vergleich mit Less

Auch die Mixins werden bei Sass etwas anders verwendet als bei Less. Hier müssen Sie ein Mixin mit dem Schlüsselwort `@mixin` definieren und beim Aufruf `@include` verwenden. Auch dazu eine Gegenüberstellung von Sass und Less:

Sass	Less

`@mixin abstand($pad: 5px) {`
`  padding: $pad;`
`  margin: 10px 0px;`
`}`
 	
`#news p {`
`  @include abstand(3%);`
`  text-align: center;`
`}`
	`.abstand(@pad: 5px) {`
	`  padding: @pad;`
	`  margin: 10px 0px;`
	`}`
	
	`#news p {`
	`  .abstand(3%);`
	`  text-align: center;`
	`}`

Tabelle 21.6 Mixins von Sass im Vergleich mit Less

Auch zu Sass ließe sich noch viel mehr schreiben, aber auch hier verweise ich Sie auf die offizielle Webseite des Projekts *http://sass-lang.com*.