

SASS und Compass

Struktur für eure Stylesheets

Guten Tag

- Thomas Moseler
- Frontend-Developer
- www.rufzeichen-online.de

Warum?

```

.page-home .view-portraits-statement-startseite .views-field-field-image-undpauler {
... }

.page-home .view-portraits-statement-startseite .views-field-field-statement {
... }

.page-home .view-portraits-statement-startseite .views-field-field-statement .field-content {
... }

.page-home .view-portraits-statement-startseite .views-field-field-statement p {
... }

.page-home .views-slideshow-controls-bottom .views-slideshow-pager-field-item {
... }

.page-home .views-slideshow-controls-bottom .views-slideshow-pager-field-item.active {
... }

.page-home .views-slideshow-controls-bottom .views-slideshow-pager-field-item.active li {
... }

```

```

.page-home .view-portraits-statement-startseite .views-field-field-image-undpauler {
... }

.page-home .view-portraits-statement-startseite .views-field-field-statement {
... }

.page-home .view-portraits-statement-startseite .views-field-field-statement .field-content {
... }

.page-home .view-portraits-statement-startseite .views-field-field-statement p {
... }

.page-home .views-slideshow-controls-bottom .views-slideshow-pager-field-item {
... }

.page-home .views-slideshow-controls-bottom .views-slideshow-pager-field-item.active {
... }

.page-home .views-slideshow-controls-bottom .views-slideshow-pager-field-item.active li {
... }

```

```
#menu {  
... }  
  
#menu ul {  
... }  
  
#menu ul li {  
... }  
  
#menu ul li a {  
... }  
  
#menu ul li a span {  
... }
```

```

a:hover, a:active {
  color: #77B302;
}

li a.active {
  color: #77B302;
}

h1, h2, h3, h4, h5, h2 a:link, h2 a:visited, h3 a:link, h3 a:visited {
  color: #77B302;
}

#sidebar-right .view-comments-recent .views-row a:hover,
#sidebar-right .view-comments-recent .views-row a:active {
  color: #77B302;
}

.page-home .view-portraits-statement-startseite .views-field-field-
statement {
  color: #77B302;
}

```

Don't repeat yourself!

Probleme von CSS

- Unstrukturiert
- rein additiv
- Bietet keine Programmierlogik
- Durchblick durch ein fremdes Stylesheet mit 1000+ Zeilen?

Was ist SASS?

- Übersetzung: „Syntactically Awesome Stylesheets“
- CSS-Präprozessor
- Wird in CSS kompiliert
- CSS ist eine Untermenge von SASS
→ Man kann also normales CSS in SASS-Dateien hineinschreiben, und Schritt für Schritt SASS-Befehle hinzufügen.
- <http://sass-lang.com/>
- Es gibt auch noch LESS, das ist ähnlich <http://lesscss.org/>
- Ausführlicher Vergleich: <http://css-tricks.com/sass-vs-less/>

Was ist Compass?

- Vor allem eine Bibliothek aufbauend auf SASS
- Es gibt jede Menge vordefinierte Elemente, die man wiederverwenden kann → spart noch mehr Arbeit
- <http://compass-style.org/>
- <http://compass-style.org/reference/compass/css3/>

Warum gleich Compass verwenden statt nur SASS?

- Ich kann die ganzen Compass-Goodies verwenden, muß aber nicht
- Benutzt sich genauso wie SASS, wenn ich die Compass-Bibliothek nicht verwende

Installation

- <http://www.undpaul.de/blog/2012/03/15/sass-und-compass-teil-2-compass-aufsetzen>
- Man braucht Ruby und Ruby Gems
- Es gibt auch PHP-Versionen, die nicht ganz so aktuell sind. Dadurch auch auf Online-Server installierbar. Interessant wenn ihr gewohnt seid direkt auf dem Server zu arbeiten
- Auf Mac: Ruby schon vorhanden
- Auf PC: je nach Geschmack, ich benutze Cygwin
- Noch einfacher: <http://mhs.github.com/scout-app/>
→ braucht allerdings Java, und wenn das nicht im Standard-Pfad installiert ist, wird's auch gleich wieder fummelig. Man muß dann in [ScoutInstallDir]\javascripts\app\process_interaction.js den Pfad zur Java.exe (respektive entsprechender Name aufm Mac) anpassen.

Zwei verschiedene Möglichkeiten der Syntax

- SASS [Demo] .sass

```
ul#main-menu
  li
    padding: 0
    a
      text-decoration: none
      &:hover
        background: #FFA40B
```

- SCSS .scss

```
ul#main-menu {
  li {
    padding: 0;
    a {
      text-decoration: none;
      &:hover {
        background: #FFA40B;
      }
    }
  }
}
```

Aufsetzen des Kompilierens mit Compass

- Compass config [Demo]
- /config/config.rb

```
http_path = "/"  
css_dir = "../css"  
sass_dir = "../sass"  
images_dir = "images"  
javascripts_dir = "javascripts"
```

Kompilieren

- Manuell

```
compass compile config
```

- Automatisch

```
compass watch config
```

Ist das nicht schön

- Zeilennummer der Herkunftszeile in der .scss Datei wird angegeben

```
/* line 24, ../sass/style.scss */
html, body {
  ... }
```

- Error reporting [Demo]

```
error sass/style.scss (Line 35: Undefined mixin 'hello-susy'.)
```

Zwei verschiedene Arten von Kommentaren

- Nur in der .scss Datei sichtbar

```
// Dieser Kommentar ist nur in der Quelldatei sichtbar
html, body {
  ... }
```

- Wird beim Kompilieren in die .css Datei übernommen

```
/* Dies ist ein normaler CSS-Kommentar,
daher wird er auch beim Kompilieren übertragen. */
html, body {
  ... }
```

Zwei verschiedene Arten von Kommentaren

- Nur in der .scss Datei sichtbar

```
// Dieser Kommentar ist nur in der Quelldatei sichtbar
html, body {
  ... }
```

- Wird beim Kompilieren in die .css Datei übernommen

```
/* Dies ist ein normaler CSS-Kommentar,
daher wird er auch beim Kompilieren übertragen. */
html, body {
  ... }
```

Drei Dinge braucht mein CSS

- Variablen
- Verschachtelung
- Mixins

Variablen

- z.B. für Farbwerte

```
$font-base-color: #444;
$primary-color: #22A81F;
```

- Wie wäre es mit Schriftgrößen und dem Font-Stack?

```
$font-base-size: 12px;
$font-size-minus-one: 0.93em;
$font-size-minus-two: 0.83em;
$font-family-copy: verdana, arial, helvetica, sans-serif;
$font-family-headlines: „trebuchet ms“, arial, helvetica, sans-serif;
```

Verschachtelung

- Man verschachtelt mit mehreren Klammern, SASS übersetzt auch das in „flaches“ CSS

```
#menu {  
  ul {  
    padding: 0;  
    li {  
      display: inline;  
      margin-right: 5px;  
      a {  
        display: inline-block;  
        background: $light-grey;  
        border: 1px solid #ddd;  
        padding: 3px 6px;  
      }  
    }  
  }  
}
```

Verschachtelung

- Keine Wiederholungen
- Liest sich wie HTML

Mixins

- ermöglichen komplexere Funktionen und das Ersetzen von größeren Code-Blöcken
- Vom Aufbau wie eine Funktion in PHP: man kann Argumente in das Mixin hineingeben
- Wird eingebunden mit `@include [Name-des-Mixins]`, wobei Name-des-Mixins kein Dollarzeichen oder ähnliches braucht

```
@mixin tommy-border-radius ($radius: 20px) {  
    border-radius: $radius;  
    webkit-border-radius: $radius;  
    moz-border-radius: $radius;  
}  
  
( ... )  
  
@include tommy-border-radius;
```

Mixins aus Compass

- Genau hier kommt Compass ins Spiel: z.B. Mixins für border-radius, box-shadow oder viele andere CSS3-Effekte sind schon vorhanden [Demo]

```

@import compass/css3;
@import compass/typography;

( ... )

@include border-radius;

#menu ul {
  @include horizontal-list(5px, left);
}

```

Anspieltips:

- Sprites: erzeugt aus einzelnen Bildern Sprites. Sehr abgefahren.

Farbpalette definieren

```

(...)
<div class="orange-150"></div>
<div class="orange"></div>
<div class="orange-50"></div>
(...)

.orange {
  background: $main-orange;
}
.orange-50 {
  background: lighten($main-orange, 25%);
}
.orange-25 {
  background: lighten($main-orange, 37.5%);
}
.orange-150 {
  background: darken($main-orange, 12%);
}
.orange-200 {
  background: darken($main-orange, 20%);
}

```

Das abstrahieren wir noch ein bisschen mehr...

Farbpalette definieren

```
<div class="orange darken-200"></div>  
<div class="orange darken-150"></div>  
<div class="orange"></div>  
<div class="orange lighten-50"></div>  
<div class="orange lighten-25"></div>
```

Die Klasse für die Grundfarbe wird von der Klasse für die Farbabstufung getrennt. So können wir auch eine andere Farbe einsetzen und dafür die Farbabstufungen erzeugen.

Abstufungen unabhängig von Farbe

```

$lighten-50: 25%;
$lighten-25: 37.5%;
$darken-150: 12%;
$darken-200: 20%;

.orange {
  $palette-color: $main-orange; // enter the color for this palette here
  background: $palette-color;
  &.lighten-50 {
    background: lighten($palette-color, $lighten-50);
  }
  &.lighten-25 {
    background: lighten($palette-color, $lighten-25);
  }
  &.darken-150 {
    background: darken($palette-color, $darken-150);
  }
  &.darken-200 {
    background: darken($palette-color, $darken-200);
  }
}

```

Das &-Zeichen innerhalb des Nestings bewirkt, daß die zwei Klassen als .orange.orange-50 direkt hintereinander geschrieben werden: sie befinden sich am gleichen Element.

Abstufungen unabhängig von Farbe

Das ist mir aber immer noch nicht effizient genug. Definieren wir ein Mixin:

```

@mixin color-palette ($palette-color: #888) {
  $lighten-50: 25%;
  $lighten-25: 37.5%;
  $darken-150: 12%;
  $darken-200: 20%;

  background: $palette-color;
  &.lighten-50 {
    background: lighten($palette-color, $lighten-50);
  }
  &.lighten-25 {
    background: lighten($palette-color, $lighten-25);
  }
  &.darken-150 {
    background: darken($palette-color, $darken-150);
  }
  &.darken-200 {
    background: darken($palette-color, $darken-200);
  }
}

```

Das sieht doch schon ganz anders aus...

Nun können wir mit wenigen Zeilen Code für verschiedene Farben unsere Abstufungen definieren.

```
/**** Farben ***/  
$main-orange: #E06003;  
$medium-blue: #3496ED;  
  
/*** Paletten-Definitionen ***/  
.orange {  
  @include color-palette ($main-orange);  
}  
.medium-blue {  
  @include color-palette ($medium-blue);  
}  
.grey {  
  @include color-palette;  
}
```

Das eröffnet Möglichkeiten

Diese dynamische Definition von Farbenermöglichst ein ganz anderes Arbeiten. Vor allem in der Konzeptionsphase möchte man die Farben vielleicht ändern oder andere Kombinationen ausprobieren.

[Demo]

Was noch alles geht

- If-Statements
- Berechnungen
- und, und, und...

Weiterführende Links

- Sehr schönes Einführungsvideo: <http://vimeo.com/24278115>
- Tutorial auf der Projektseite von SASS <http://sass-lang.com/tutorial.html>
- Compass <http://compass-style.org/reference/compass/>