

CSS Preprocessors

SASS



Wat is SASS

- CSS Preprocessor
 - scriptingtaal
 - Voegt functionaliteit toe aan je stylesheet markup
- Verschillende CSS preprocessors bestaan:
 - **SaSS**: Syntactically Awesome Style Sheets
 - Less
 - Stylus
- SaSS compileert naar CSS

Waarom SaSS

- CSS heeft een achterhaalde opbouw
 - Specifieke prefixes
 - Vervangen van kleurcodes in een project
 - Dezelfde stijlen komen voor in verschillende CSS blokken
 - Complexe en lange CSS selectoren worden gebruikt doorheen het project

Waarom SASS

- Sass lost deze problemen op
 - Maakt CSS schaalbaar, herbruikbaar en slim
 - 2 conventies:
 - .SASS
 - .SCSS

- SASS live testen in de browser:

<https://www.sassmeister.com/>

.SCSS

- SCSS volgt de algemene CSS bracket syntax

```
body #main .value {  
    color: #eee;  
  
    a.link{  
        color: #101010;  
        text-decoration: none;  
    }  
}
```

.SASS

- SASS werkt aan de hand van indentation

```
body #main .value
  color: #eee

a.link
  color: #101010
  text-decoration: none
```

SASS in Angular

- SASS is eenvoudig te integreren in nieuwe Angular projecten

```
C:\Users\20003125\workspace>ng new voorbeeldsass
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use?
  CSS
> SCSS [ http://sass-lang.com ]
  SASS [ http://sass-lang.com ]
  LESS [ http://lesscss.org ]
  Stylus [ http://stylus-lang.com ]
```

- Of is bestaande projecten

```
ng config schematics.@schematics/angular:component.styleext scss
```

SaSS functionaliteiten

- Gebruik van variabelen
- Nesting
- Inheritance
- Mixins
- Partials
- Operators

SASS variabelen

- Variabelen maken het mogelijk om een waarde aan een placeholder te koppelen

```
$varname: waarde  
selector {  
    property: $varname;  
}
```

- Werkt met hex-values, strings, kleuren, getallen, booleans,...
- Variabelen zijn scoped
 - Enkel beschikbaar in de blok waar ze gedeclareerd worden

SASS Variabelen

```
$font-stack:    Helvetica, sans-serif;  
$primary-color: #333;  
  
body {  
    font: 100% $font-stack;  
    color: $primary-color;  
}
```

SASS Nesting

- Nesting maakt lange css selectoren onnodig
- Veel vaak terugkomende niveaus zijn maar één maal gedefinieerd
- Duidelijke link in structuur van HTML en SASS

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

SASS Nesting

```
1 ▾ body #main .value {  
2   color: #eee;  
3  
4 ▾   a.link {  
5     color: #101010;  
6     text-decoration: none;  
7   }  
8  
9 ▾   span.highlight {  
10     font-weight: bolder;  
11   }  
12 }
```

```
1 ▾ body #main .value {  
2   color: #eee;  
3 }  
4 ▾ body #main .value a.link {  
5   color: #101010;  
6   text-decoration: none;  
7 }  
8 ▾ body #main .value span.highlight {  
9   font-weight: bolder;  
10 }  
11
```

SASS Inheritance

- Een van de handigste functies van SASS
- @Extend geeft ons de mogelijkheid om selectoren uit te breiden
 - Laat overerven van blokken css properties

```
%equal-heights {  
  display: flex;  
  flex-wrap: wrap;  
}  
  
.message {  
  @extend %equal-heights;  
}
```

- Verhoogt herbruikbaarheid van onze CSS code

SASS Inheritance

```
1 ▾ %message-shared {  
2   border: 1px solid #ccc;  
3   padding: 10px;  
4   color: #333;  
5 }  
6 |  
7 ▾ .message {  
8   @extend %message-shared;  
9 }  
10  
11 ▾ .success {  
12   @extend %message-shared;  
13   border-color: green;  
14 }  
15  
16 ▾ .error {  
17   @extend %message-shared;  
18   border-color: red;  
19 }  
20  
21 ▾ .warning {  
22   @extend %message-shared;  
23   border-color: yellow;  
24 }
```

```
1 ▾ .message, .success, .error, .warning {  
2   border: 1px solid #ccc;  
3   padding: 10px;  
4   color: #333;  
5 }  
6  
7 ▾ .success {  
8   border-color: green;  
9 }  
10  
11 ▾ .error {  
12   border-color: red;  
13 }  
14  
15 ▾ .warning {  
16   border-color: yellow;  
17 }  
18
```

SASS Mixins

- Vaak extra css regels voorzien om properties te ondersteunen op verschillende browsers
- Vaak bundel van css regels die in verschillende selectoren gebruikt wordt
- In SASS kan dit in een mixin voorzien worden om vervolgens doorheen het project te gebruiken
- Mixins kunnen ook argumenten meekrijgen in de vorm van een variabele

SASS Mixins

```
1 ▾ @mixin overlay() {  
2   bottom: 0;  
3   left: 0;  
4   position: absolute;  
5   right: 0;  
6   top: 0;  
7 }  
8  
9 ▾ .modal-background {  
10  @include overlay();  
11  background: black;  
12  opacity: 0.9;  
13 }
```

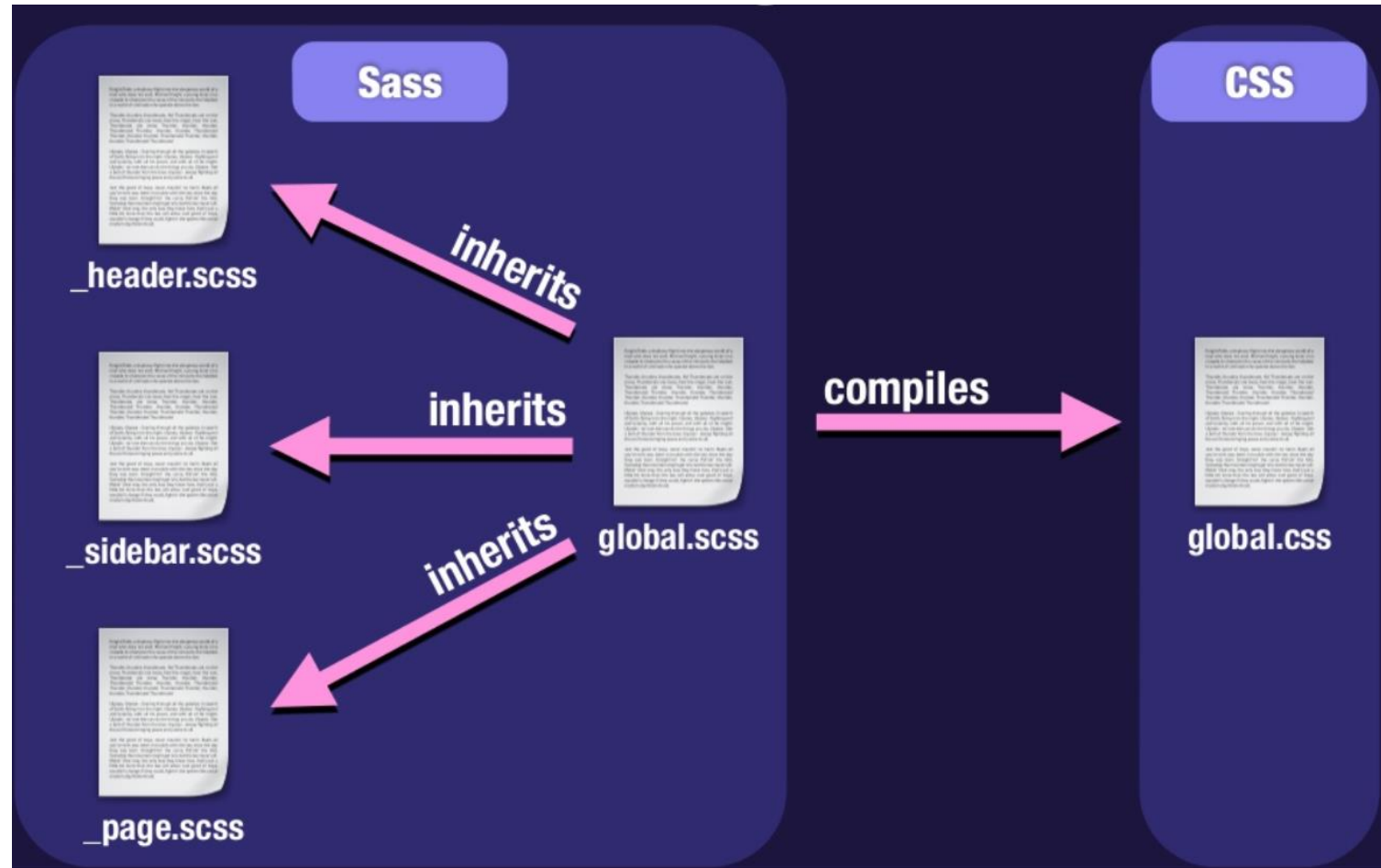
```
1 ▾ .modal-background {  
2   bottom: 0;  
3   left: 0;  
4   position: absolute;  
5   right: 0;  
6   top: 0;  
7   background: black;  
8   opacity: 0.9;  
9 }  
10
```


Partials

- CSS meestal in één bestand
- Partials maakt het mogelijk om CSS op te splitsen in meerdere bestanden (partial files)
- Partials starten altijd met een underscore (_)
- In de global.scss file importeren:

```
@import("header","sidebar","page");
```

Partials



Operatoren

- In SASS kunnen wiskundige operatoren zoals + - / * en % gebruikt worden

```
1 ▾ .container {  
2   width: 100%;  
3 }  
4  
5 ▾ article[role="main"] {  
6   float: left;  
7   width: 600px / 960px * 100%;  
8 }  
9  
10 ▾ aside[role="complementary"] {  
11   float: right;  
12   width: 300px / 960px * 100%;  
13 }
```

```
1 ▾ .container {  
2   width: 100%;  
3 }  
4  
5 ▾ article[role="main"] {  
6   float: left;  
7   width: 62.5%;  
8 }  
9  
10 ▾ aside[role="complementary"] {  
11   float: right;  
12   width: 31.25%;  
13 }
```