SASS

css on steroids

WAAROM ZOU JE SASS GEBRUIKEN?

- *weer een nieuwe trend?*
- *Ik schrijf al 15 jaar CSS en dat werkt prima*
- *Ik heb zeker geen extra hulp nodig*
- *Mijn workflow is al complex genoeg*

Het is soms erg lastig om wéér iets nieuws te leren

ELEVATOR PITCH

“Heb je ooit al eens een zoek-en-vervang moeten doen om een kleur aan te passen?”



WAAROM IS CSS ZO LASTIG?

- *Er zijn erg veel properties*
- *Cascade is lastig, gewicht selectors, volgorde.. etc*
- *Er zijn veel selectors*
- *Er zijn veel Quirks, backwards-compatible, browser-prefixing, unsupported features*

CSS IS EENVOUDIG

keep it simple

- *De syntax is bewust eenvoudig*
- *Lage leercurve voor designers*
- *Gevolg: veel herhaling van properties & value's*
- *Gevolg: 1 aanpassing is veel werk*
- *Hacks maken het nog lastig, wie kan een clearfix uit zijn hoofd schrijven?*

DRY

don’t repeat yourself

EENVOUDIG OF EFFICIENT?

variabelen voorkomen herhaling

- *CSS heeft traditioneel (geen) variabelen (x = 10)*
- *Of conditional logic (if / else)*
- *Of functies*
- *Of includes, constants..*

WAT IS SASS?

Met Sass schrijf je bijna-CSS, maar met al deze extra's. Deze zet je om naar puur 'vanilla CSS', dat iedere browser begrijpt. Sass is de laag tussen jou en de CSS.



SASS WORDT GECOMPILED

van SASS naar CSS

- *Via CLI: Grunt, Gulp of een andere build-tool*
- *Via GUI: CodeKit, Koala etc.*



SCSS OF SASS?

sass is een extension van css3

Sass is een extension van CSS3,

en 'Sassy CSS' ofwel 'SCSS'-dialect is een superset van CSS3

(This means that every valid CSS3 stylesheet is valid SCSS as well)

Enkel de extensie wijzigen; de overgang is dus pijnloos.

VOORBEELD

scss omgezet naar css met CodeKit:

sass_voorbeeld_2.scss

```scss

$pink: #ea4c89;

p {
    font-size: 12px;
    color: $pink;
}

p strong {
    text-transform: uppercase;
}
```

Line 1, Column 1 Tab Size: 4 SCSS

sass_voorbeeld_2.css

```css
p {
  font-size: 12px;
  color: #ea4c89; }

p strong {
  text-transform: uppercase; }

```

Line 7, Column 1 Tab Size: 4 CSS

SIMPEL

je renamed je style.css naar style.scss
en begint variabelen te gebruiken etc.

SASS

de niet-scss-syntax is anders:
geen curly-braces, geen semi-colon, wel inspringen

```
1  $font-stack:    Helvetica, sans-serif
2  $primary-color: #333
3
4  body
5    font: 100% $font-stack
6    color: $primary-color
```

NADELEN?

Breaking up with Sass: it's not you, it's me

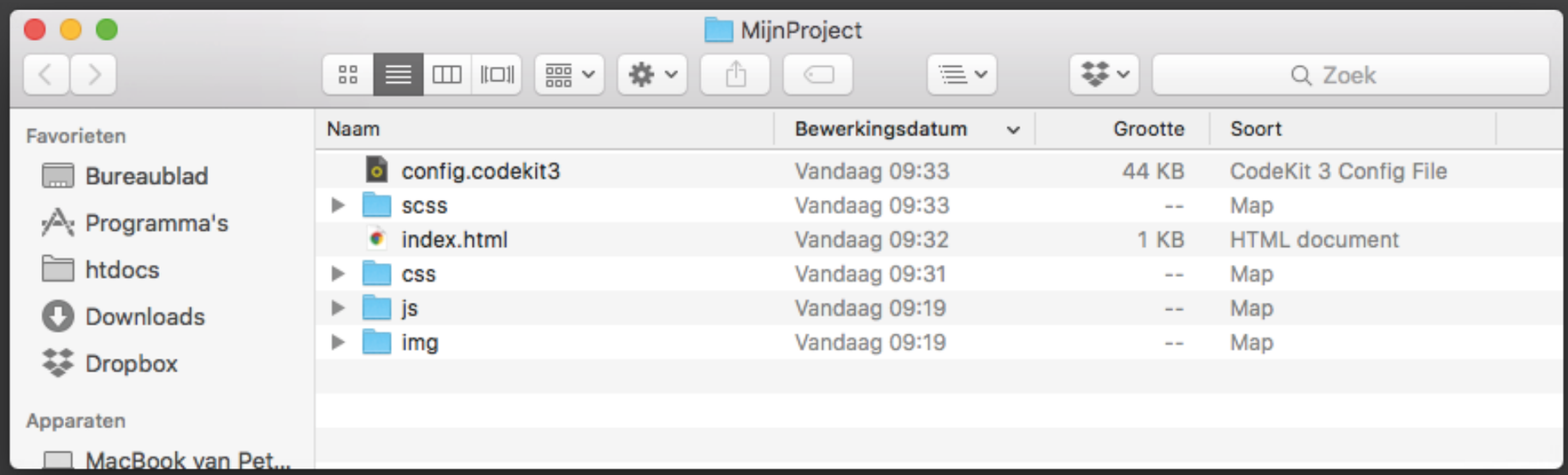
- *Angst voor het onbekende*
- *Weer 'iets nieuws' leren*
- *Je haat command-line interfaces*
- *Je project-map wordt weer iets drukker*

Deze redenen mogen u niet tegenhouden.

ORDE

in de chaos: organiseer je map

- *Je voorkomt .css-, sourcemap-, scss- en partial-bestanden die doorelkaar staan.*



GUI TOOLS:

CodeKit (OSX):
https://codekitapp.com/

Koala (OSX, Windows, Linux):
http://koala-app.com

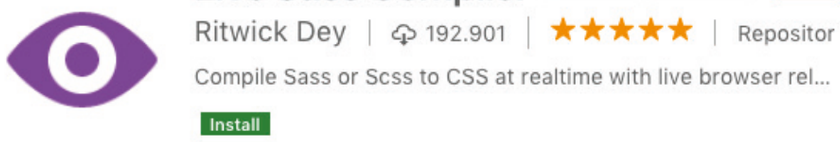
ANDERE VOORDELEN

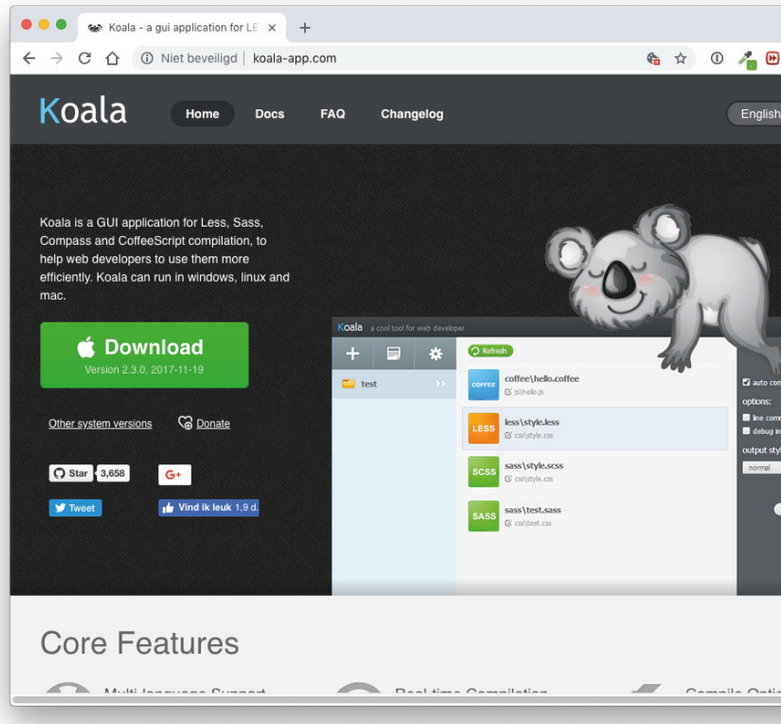
variabelen voorkomen herhaling

- *Compile Sass, Less, Stylus, CSS, CoffeeScript, Pug, Slim, Haml, TypeScript, JavaScript, ES6, Markdown, SVG, PNG, GIF and JPEG right out of the box.*
- *Compress CSS, Autoprefix, foutcontrole*
- *Gebruik Bless.js om automatisch IE te ondersteunen. (blesscss.com)*
- *Refresh automatisch je browser als je iets opslaat*
- *Automatisch Sourcemaps aanmaken*
- *Afbeeldingen, SVG automatisch verkleinen*
- *JS syntax en error checking*

BLIJF VAN JE CSS!

Vanaf dit moment pas je NIETS meer aan in je CSS; alles gebeurt nu in je SASS-bestanden.



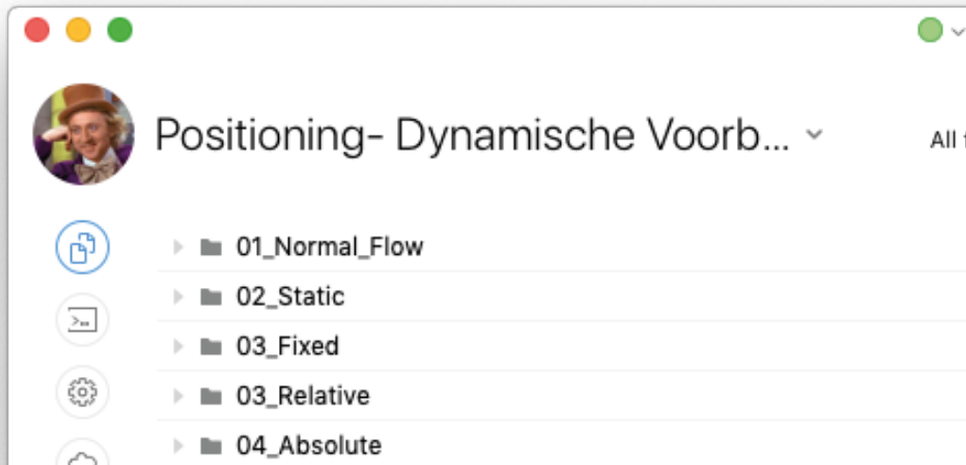


OEFENING 1

experimenteer met het opzetten van een SCSS systeem

---

1) Gebruik een online tool als: codepen, leuk voor sandbox-experimenten.
2) Gebruik een plugin in je Visual Studio Code: (Live Sass Compiler)
3) Gebruik een GUI Tool: CodeKit, Koala of Prepross.io.
4) Gebruik een CLI tools: Grunt, Gulp, Webpack oid.



VARIABELEN

1x definieren op 1000 plaatsen aanpassen.

WAARVOOR GEBRUIKEN:

Colors, Font Stacks, margin & padding

(= consistency in design), of zelfs je style-guide.

VARIABELEN

voorbeelden met keywords, kleuren, boolean, strings...:

```scss
/* Numbers met of zonder units */
$rounded: 4px;
$line-height: 1.5;

/* Color */
$base: purple;
$border: rgba(0, 255, 0, 0.6);
$shadow: #333;

/* Boolean */
$rounded: false;
$shadow: true;
```

– Strings - can be set with or without quotes:

SCSS

```scss
$header: 'Helvetica Neue';
$callout: Arial;
$message: "Loading...";
```

– Lists

SCSS

```scss
$colors: red, green, blue, yellow;
$margin: 40px 0 20px 100px;
```

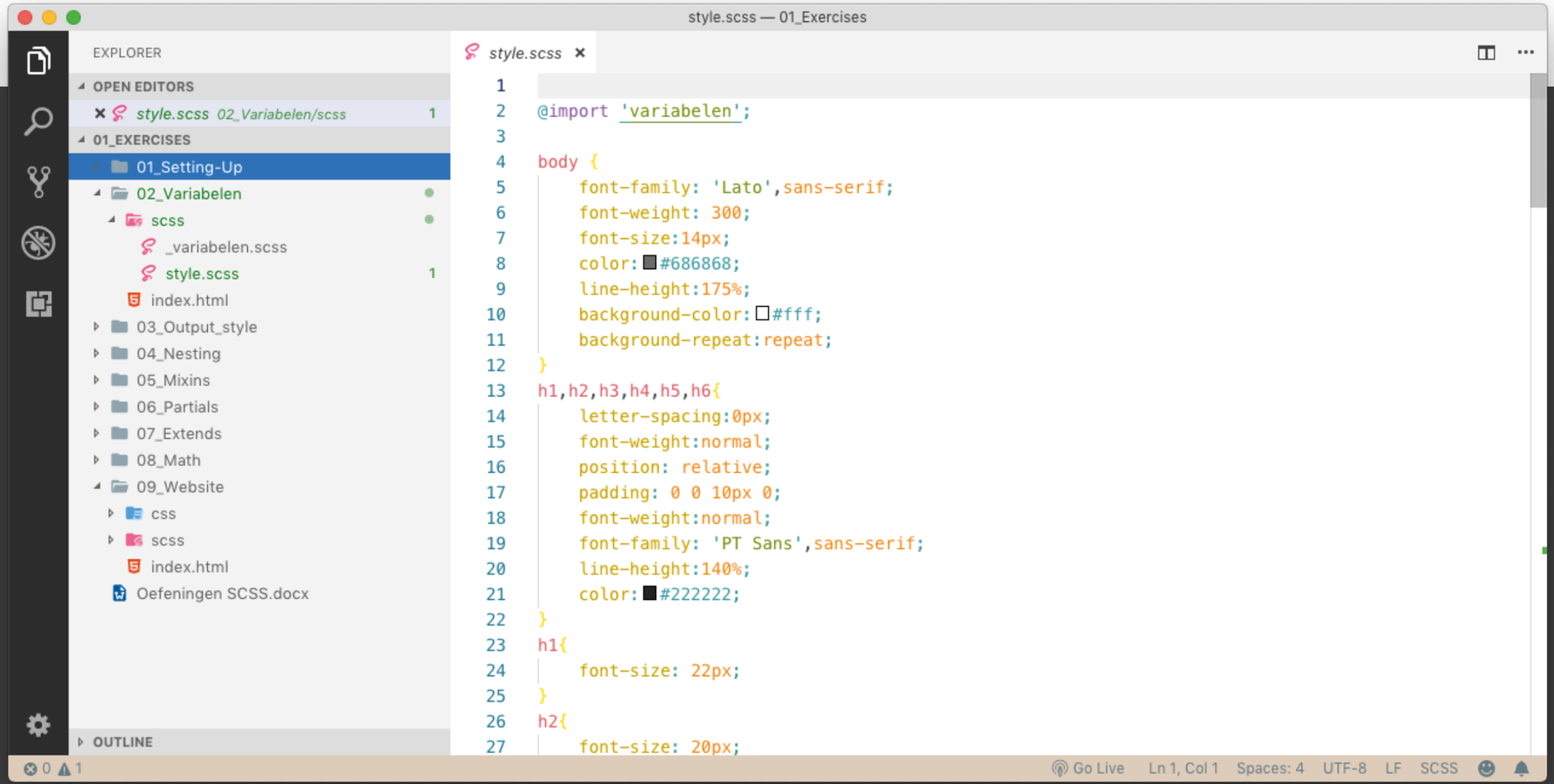
– Null

SCSS

```scss
$shadow: null;
```

OEFENING 2

clean up 'style.css' with DRY in mind.



GEEN CSS MEER?

niemand leest mijn comments meer?

Al die jaren schreef ik perfect uitgelijnde CSS,
met geweldig commentaar, wie gaat deze nog lezen?

En ik wil mijn CSS outputten volgens mijn geweldige layout met commentaar

VERSCHILLENDE OUTPUT

kies je stijl (in koala)

Nested output:

```
ol {
  margin: 10px 0;
  padding: 10px 0; }
  ol li {
    font-size: 2em;
    line-height: 1.4; }
    ol li p {
      color: #333; }
```

Expanded:

```
ol {
  margin: 10px 0;
  padding: 10px 0;
}

ol li {
  font-size: 2em;
  line-height: 1.4;
}

ol li p {
  color: #333;
}
```

Compact:

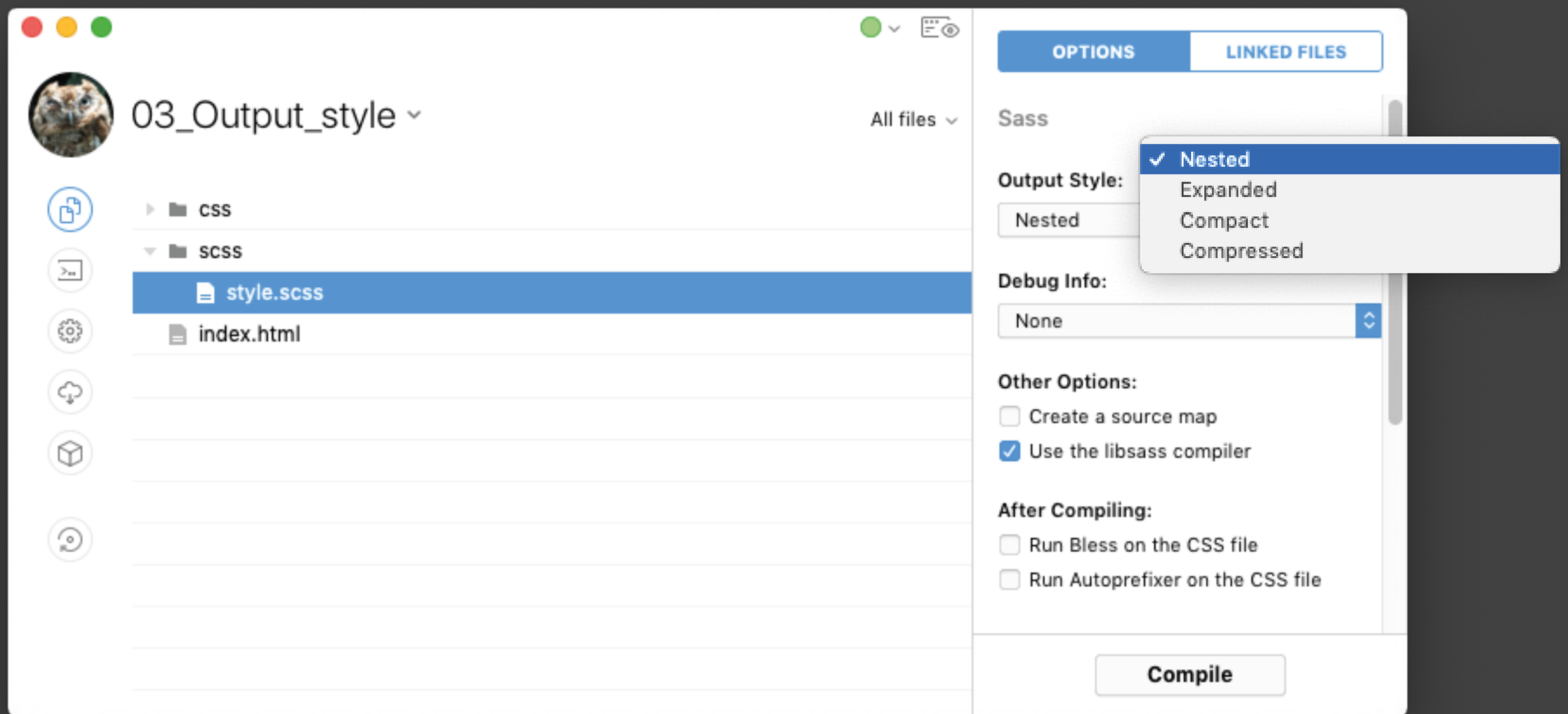
```
ol { margin: 10px 0; padding: 10px 0; }
ol li { font-size: 2em; line-height: 1.4; }
ol li p { color: #333; }
```

Compressed:

```
ol { margin: 10px 0; padding: 10px 0; }
```

OEFENING 3

Change output to 'compressed', and other options..



NESTING

groepeer selectors

BLIJF BINNEN DE HAKEN

nesting: door binnen de eerste selector te blijven, krijg je automatisch specifiekere selectors.

.scss

```scss
/* Nesting */

nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

.css

```css
nav ul {
  margin: 0;
  padding: 0;
  list-style: none; }

nav li {
  display: inline-block; }

nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none; }
```

NESTING

ook mogelijk met attributes

```
<header role="banner">
      <div id="logo">
        <img src="logo.png" alt="Sasquatch Records" />
      </div>
      <h1>Sasquatch Records</h1>
      ...
</header>
```

```
/* Tweede voorbeeld nesting */

header[role="banner"] {
          margin: 20px 0 30px 0;
          border-bottom: 4px solid #333;
          #logo {
            float: left;
            margin: 0 20px 0 0;
            img {
              display: block;
              opacity: .95;
} }
          h1 {
            padding: 15px 0;
            font-size: 54px;
            line-height: 1;
            font-family: Jubilat, Georgia
            font-weight: bold;
} }
```

```
/* Tweede voorbeeld nesting */
header[role="banner"] {
  margin: 20px 0 30px 0;
  border-bottom: 4px solid #333; }
  header[role="banner"] #logo {
    float: left;
    margin: 0 20px 0 0; }
    header[role="banner"] #logo img
      display: block;
      opacity: .95; }
  header[role="banner"] h1 {
    padding: 15px 0;
    font-size: 54px;
    line-height: 1;
    font-family: Jubilat, Georgia,
    font-weight: bold; }
```

GROEPEER PROPERTIES

minder typewerk

Properties met dezelfde namespace (font-size, font-style, font-weight), mag je combineren in een groep:

```
/* Nesting van Properties */

header[role="banner"] h1 { pa
font: {
    size: 54px;
    family: Jubilat, Georgia,
    weight: bold;
}
  line-height: 1;
}
```

```
.mybackground{
    background: {
            color: #ea4c89
            size: 16px 16p
            image: url(sas
            repeat: no-rep
            position: top
    }
}
```

NESTING

hergebruiken basis-selector, zeer nuttig:

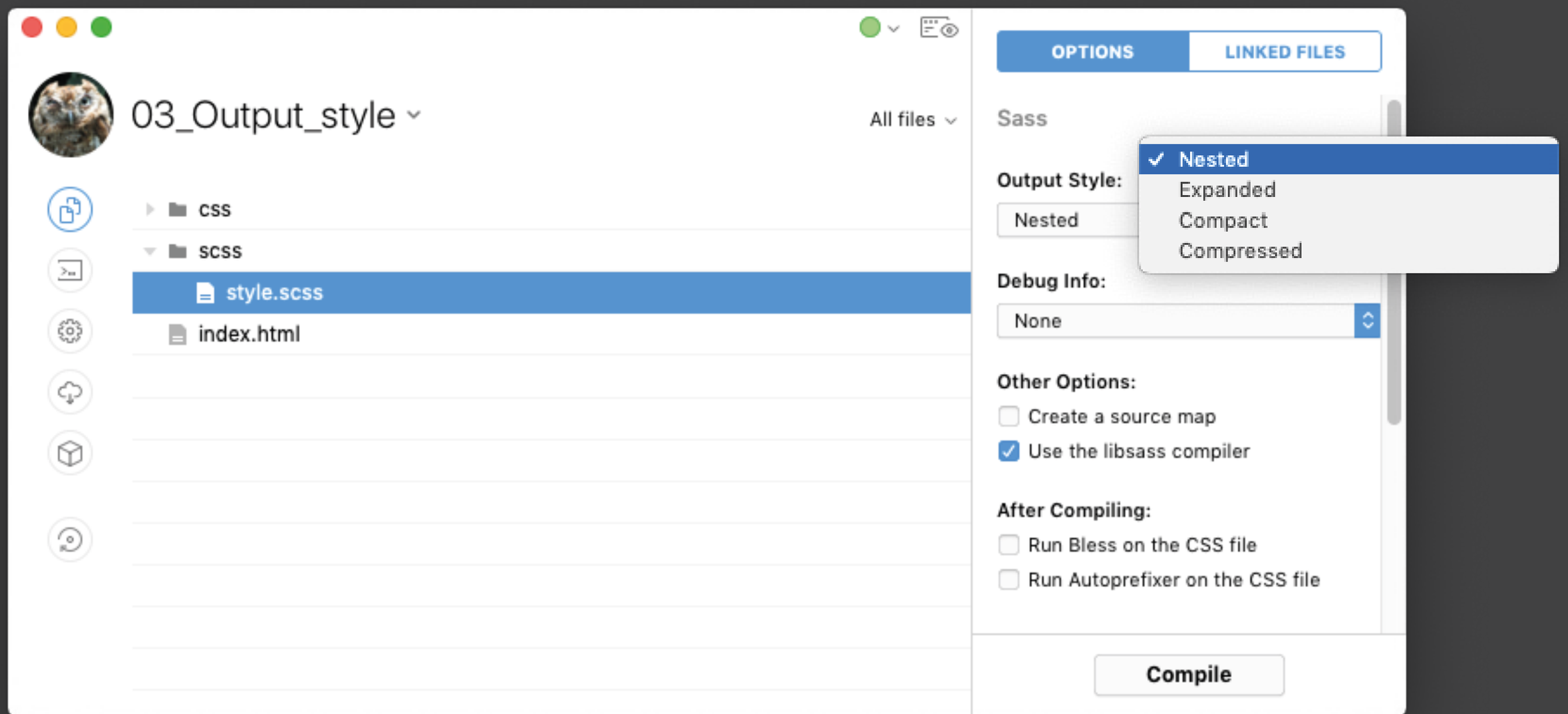
Binnen het nesten kunnen we de ampersand (&) gebruiken om de basis selector aan te roepen:

```
/* Reference base selector */

a {
    font-weight: bold; text-decoration: non
    color: red;
    border-bottom: 2px solid red;
    &:hover {
        color: maroon;
        border-color: maroon;
    }
}
```

```
/* Reference base selector */
a {
  font-weight: bold;
  text-decoration: none;
  color: red;
  border-bottom: 2px solid red; }
  a:hover {
    color: maroon;
    border-color: maroon; }
```

OEFENING 4 - NESTING

Change output to 'compressed', and other options..



STYLEGUIDE

een sass-bestand zou kunnen dienen als een styleguide:
font-sizes, colors, font-styles:

```scss
/* Lichter en donkerder */

$color-main: #333;          // black
$color-light: #999;         // grey
$color-accent: #ea4c89;  // pink

section.secondary {
    background: darken($color-accent, 30%);
}

section.secondary {
    background: lighten($color-accent, 30%);
}
```



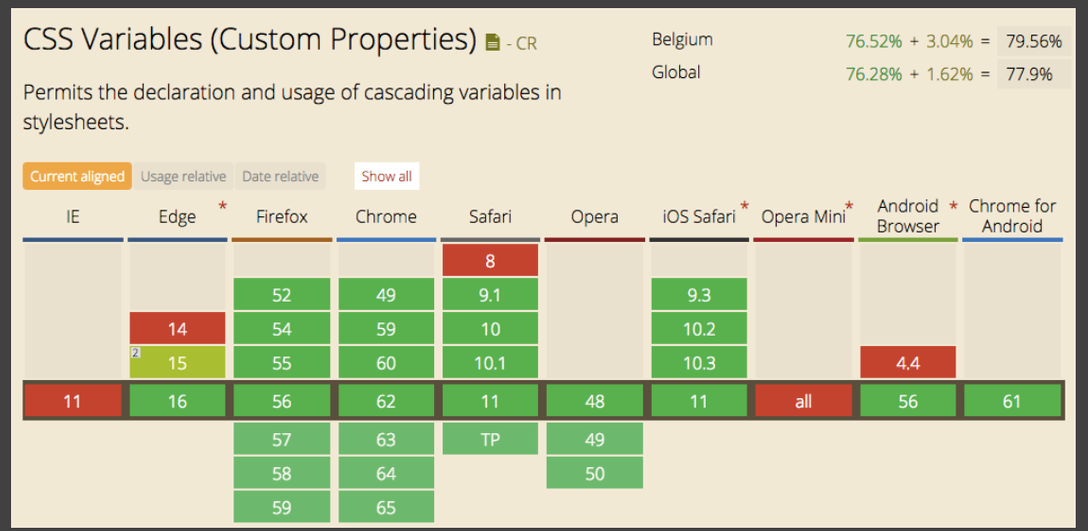
NU SASS LATER CSS

back to the future

Veel zaken die nu puur ‘sass’ zijn, worden onderdeel van CSS zelf.
CSS-variabelen worden meer en meer ondersteunt native door browsers. Echter de syntax is iets anders:

SCSS: $mijn-kleur

CSS-variabele: :root



Zeker lezen:
https://css-tricks.com/difference-between-types-of-css-variables/

MIXINS

sample with style

Mixins zijn stukken CSS, groepen properties of selectors die je kan hergebruiken.

Je kan ze ook argumenten geven, zoals bij een functie

MIXINS

hergebruik je code

Veel elementen hebben dezelfde eigenschappen.

Bijvoorbeeld:
H1 - H6: zelfde lettertype, kleur, marges, gewicht...

Als je een groep stijlkenmerken wil hergebruiken,
dan gebruik je mixins.

```scss
/* Mixins */

@mixin title-style {
        margin: 0 0 20px 0;
        font-family: $font-stack;
        font-size: 20px;
        font-weight: bold;
        text-transform: uppercase;
        color: $color-accent;
}

h1, h2, h3, h4, h5, h6{
    @include title-style;
}
```

MIXINS

meerdere argumenten, scheiden met komma's:

```
/* Multiple arguments */

@mixin title-style($color, $background) {
    margin: 0 0 20px 0;
    font-family: $font-stack;
    font-size: 20px;
    font-weight: bold;
    text-transform: uppercase;
    color: $color;
    background: $background;
}
```

CSS3 SYNTAX

vendor prefixes, juiste volgorde, syntax: lastig!

Niet met Sass:

```
/* CSS3 features gemakkelijker oproepen */

@mixin rounded($radius) {
      -webkit-border-radius: $radius;
         -moz-border-radius: $radius;
              border-radius: $radius;
}

div.rounded{
    @rounded(5px);
}

/* Another example of mixin */
@mixin shadow($x, $y, $blur, $color) {
          -webkit-box-shadow: $x $y $blur $color;
             -moz-box-shadow: $x $y $blur $color;
                  box-shadow: $x $y $blur $color;
}
```

CSS3 SYNTAX

the hellhole of gradients:

```
.gradient{
/* Fallback for sad browsers */
background-color: #d42a78;

/* Mozilla Firefox */
background-image: -moz-linear-gradient(#ff70b1, #d42a78);

/* Opera */
background-image: -o-linear-gradient(#ff70b1, #d42a78);

/* WebKit (Safari/Chrome 10) */
background-image: -webkit-gradient(linear, left top, left bottom, color-stop(0, #ff70b1

/* WebKit (Chrome 11+) */
background-image: -webkit-linear-gradient(#ff70b1, #d42a78);

/* IE10 */
background-image: -ms-linear-gradient(#ff70b1, #d42a78);

/* W3C */
background-image: linear-gradient(#ff70b1, #d42a78);
}
```

CSS3 SYNTAX

the hellhole of gradients: niet met mixins:

```
/* Gradient mixin */
@mixin linear-gradient($from, $to) {
    background-color: $to;
    background-image: -moz-linear-gradient($from, $to);
    background-image: -o-linear-gradient($from, $to);
    background-image: -webkit-gradient(linear, left top, l
    background-image: -webkit-linear-gradient($from, $to);
    background-image: -ms-linear-gradient($from, $to);
    background-image: linear-gradient($from, $to);
}

a {
    padding: 3px 8px;
    color: #fff;
    @include rounded(4px);
    @include linear-gradient(#ff70b1, #d42a78);
}
```

RECYCLE

verzamel al je mixins in een bestand

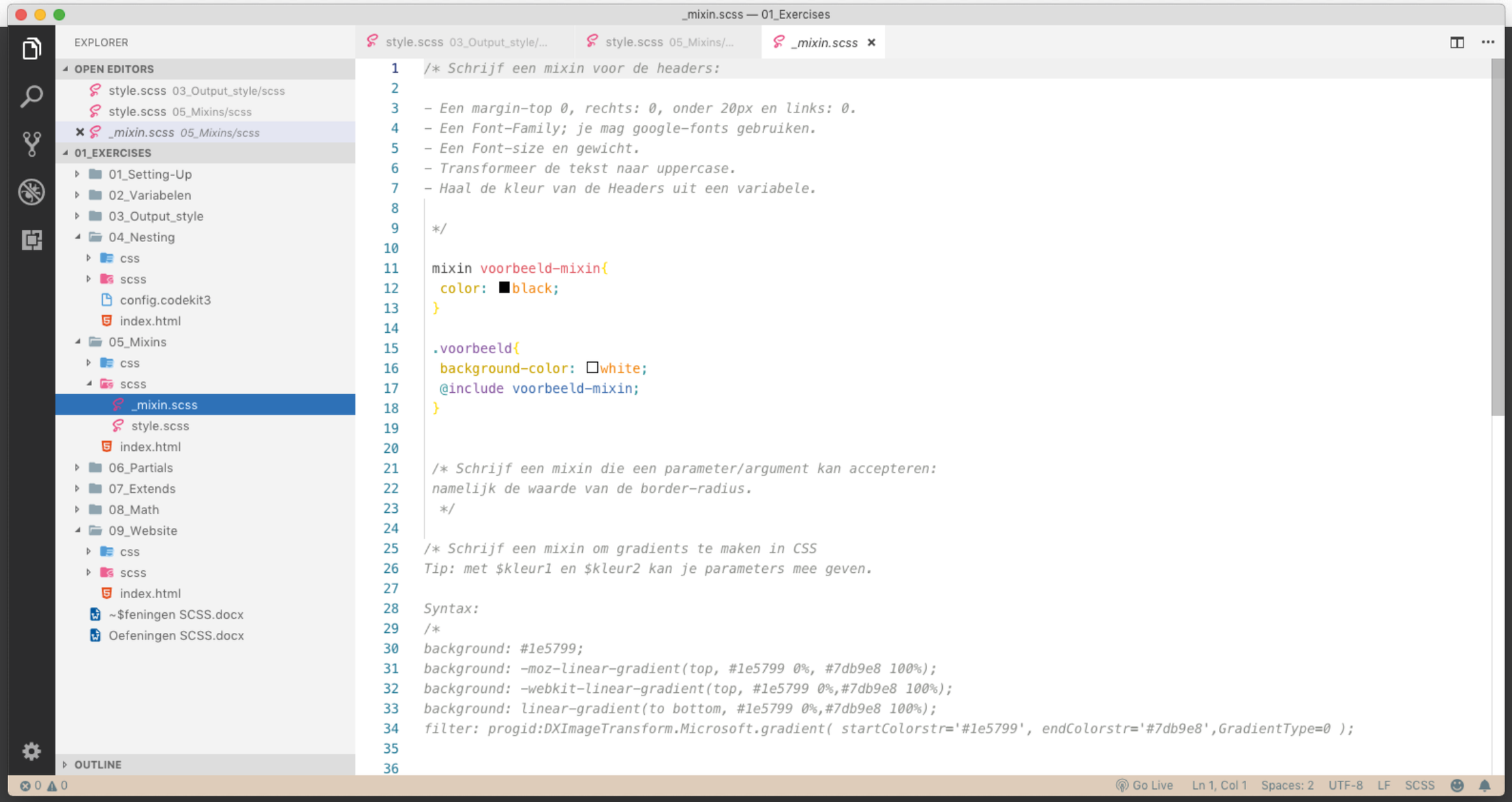
Gebruik @import om de mixins.scss te importeren

Verzamel al je code voor rounded corners, gradients, self-clearing floats... in een enkele mixins.scss

```scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

// Import other files
  @import "mixins.scss";

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

OEFENING 5

schrijf een aantal mixins voor gradients etc.
tip: google eens op 'bourbon'
of kijk eens op codepen naar mixins



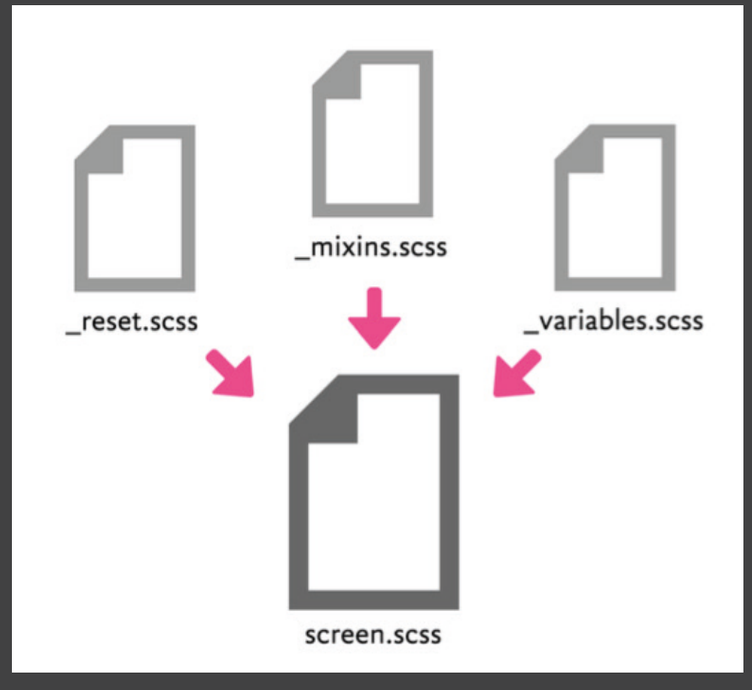
PARTIALS

voorkom dat mixin.scss gecompiled wordt naar mixin.css

PARTIALS

prefix je bestandsnaam met een underscore

- Mixin-bestanden kunnen met een “_” geprefixed worden
- Ze noemen nu ‘partials’
- Daardoor niet omgezet naar .CSS



PARTIALS

mijn eigen bibliotheek met handigheidjes…

Voordat je zelf op zondagmiddag het ultieme ‘partial’-bestand gaat maken:

Het bestaat al…

Compass (compass-style.org), Bourbon (Bourbon.io)…

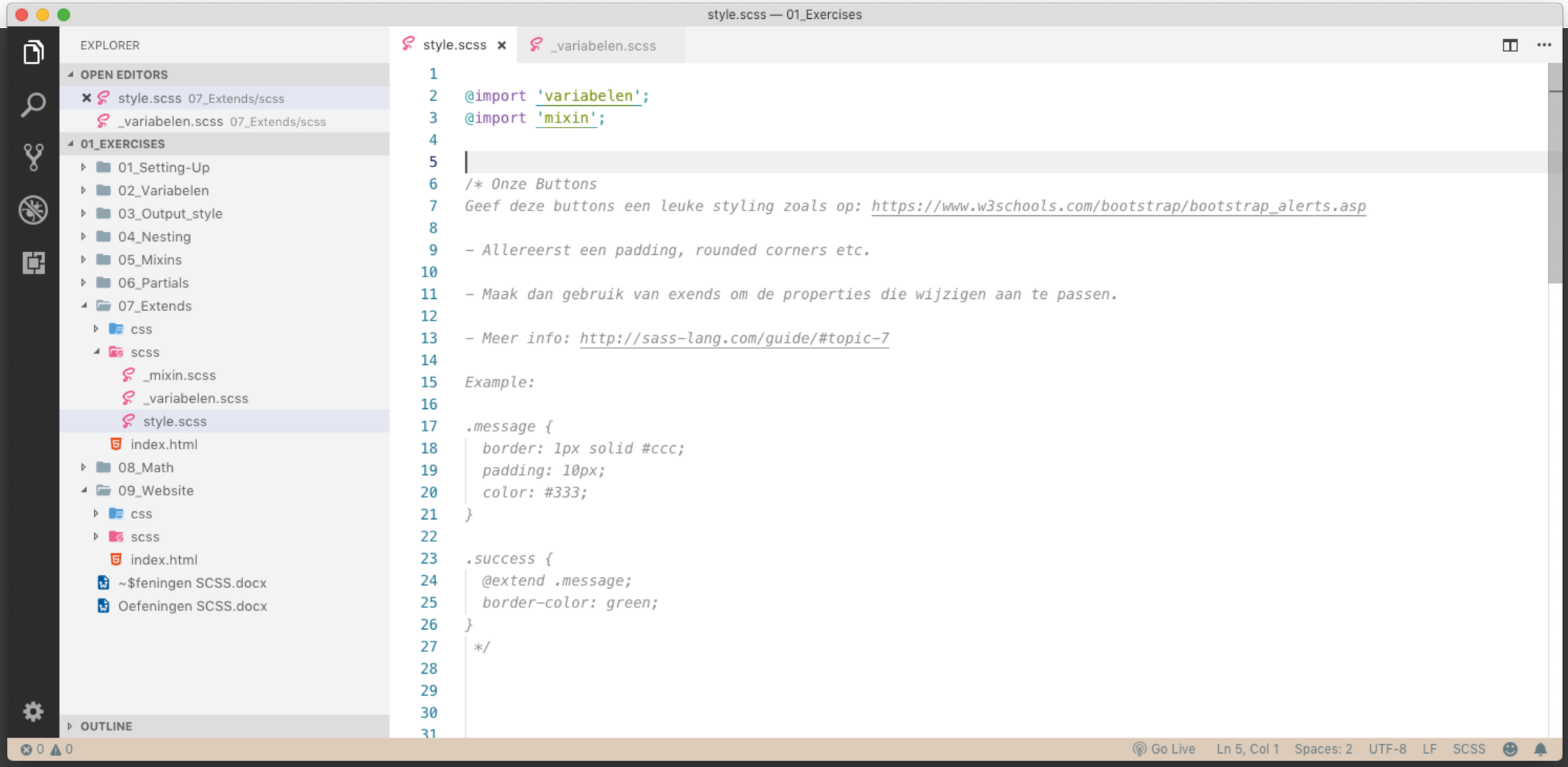
EXTEND'S

werk verder op bestaande code-fragmenten:

```
/* Extends */
.box {
    padding: 2em;
    color: black; background-color: w
}

.warning-box {
    @extend .box;
        border: 2px dotted red;
}

.success-box {
    @extend .box;
        border: 2px dotted chartreuse
}

.info-box {
    @extend .box;
    border: 2px dotted blue;
}
```

OEFENING 7

Pas de buttons aan



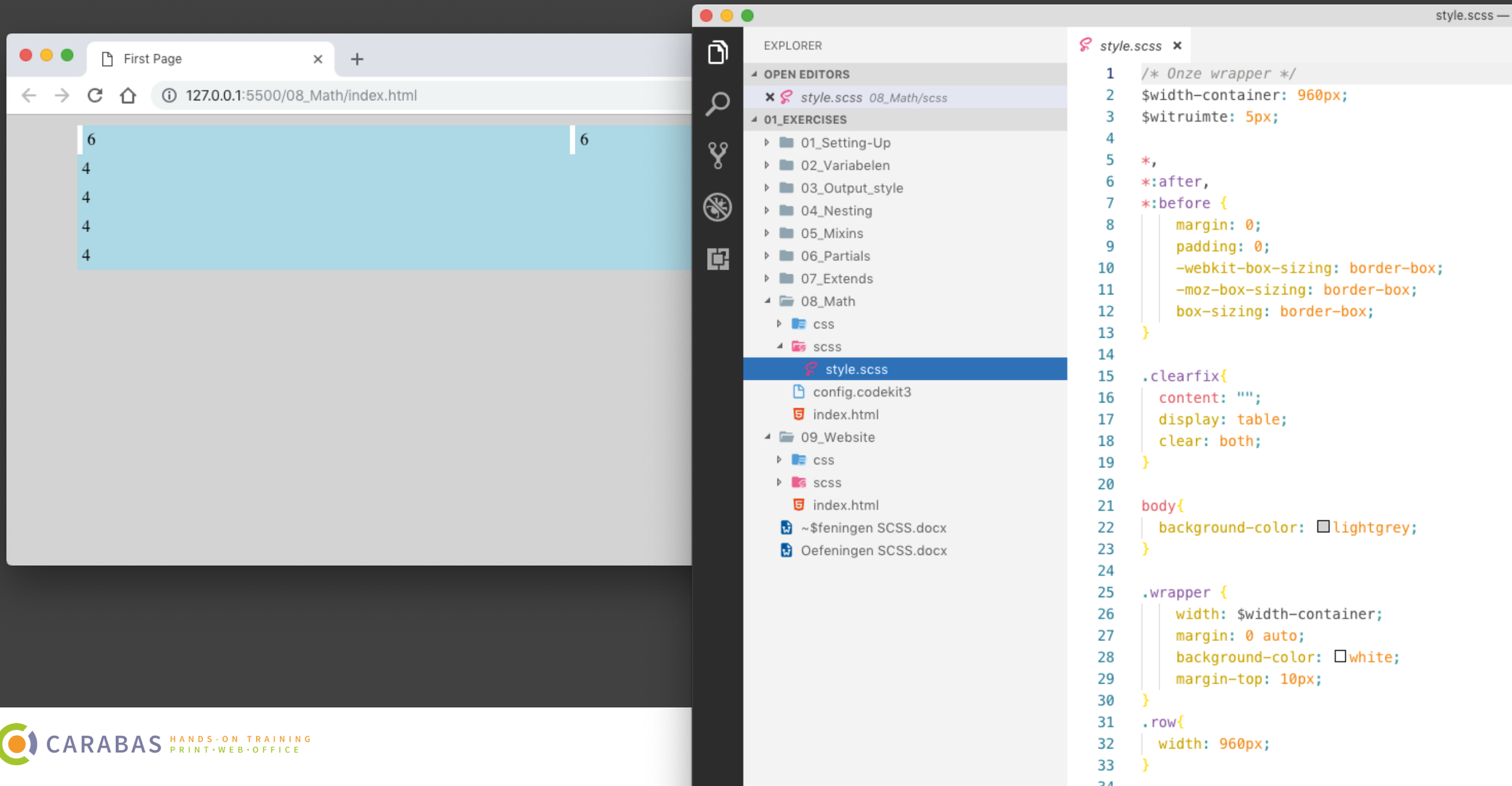
MATH

doe berekeningen in je code:

```
/* MATH */
.container { width: 100%; }


article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}

aside[role="complementary"] {
  float: right;
  width: 300px / 960px * 100%;
}
```

OEFENING 8

Math: gebruik de $width-container om de breedte van de kolommen te berekenen, combineer met mediaquery als je wil..





CONCLUSIE

Ik hoop dat ik je wat enthousiast heb kunnen maken.
Zelf gebruik ik maar 20% SASS, enkel om mijn CSS iets beheersbaarder te maken. En dat mag.

Er is nog veel meer informatie en functies die je kan doen met sass.

Dit waren mijn bronnen:
- https://abookapart.com/products/sass-for-web-designers
- https://www.slideshare.net/ShahoToofani1/sass-and-compass-workshop
- Brad Hussey op Youtube:

