HTML, CSS & JS basis

Les 8:

CSS preprocessor SASS

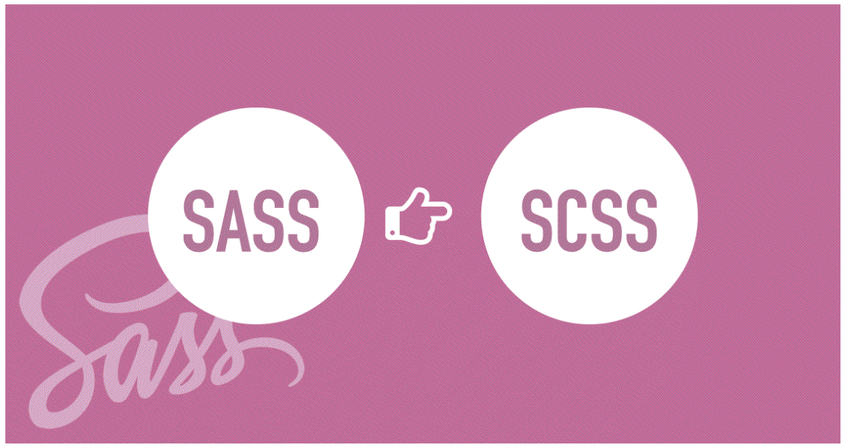
Onderwerpen les:

- SASS gebruiken in VSC
- SASS variables
- Mixins
- Nesting
- Partials and use

SASS

Wat is SASS?

SASS is een CSS extension / preprocessor taal dat wordt geïnterpreteerd of ‘compiled’ naar CSS. SASS maakt het schrijven van goede CSS makkelijker voor front-end developers. Je kunt namelijk variabelen, mixins (herbruikbare styling), nesting en veel meer gebruiken.

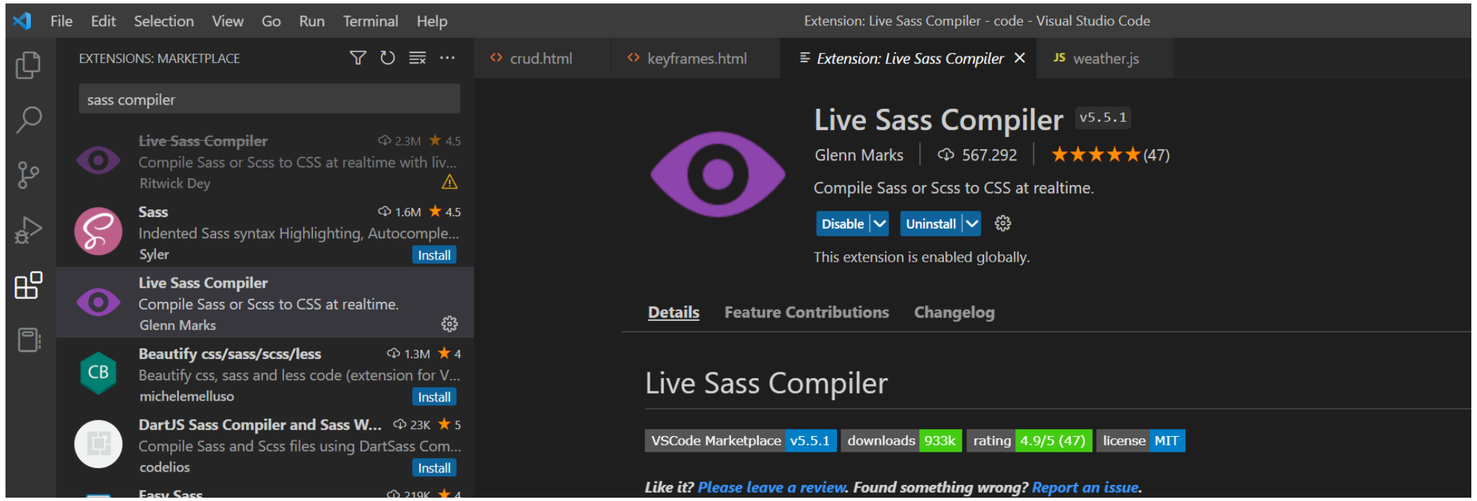


https://sass-lang.com/

SASS

SASS toevoegen aan VSC

Download de live sass compiler in VSC.



SASS

SASS toevoegen aan VSC

1 Maak een HTML bestand

2 Maak een style.scss bestand

3 Typ styling en sla op. (dit genereert automatisch een style.css bestand)

4 Link vanuit het HTML bestand naar het style.css bestand.

SASS

SASS variables

Op websites zul je bepaalde kleuren meerdere keren typen in je CSS. Misschien een tekst kleur of een brand kleur die je op buttons, linkjes en andere plekken hergebruikt. Zodra je besluit om deze kleur aan te passen dan moet je dit vaak op meerdere plekken aanpassen. In SASS kun je een variabele aanmaken op 1 plek deze hergebruiken. Zodra je de waarde van de variabele aanpast, verandert de waarde overal waar je de variabele hebt hergebruikt.

```
$dark-color: #212121;

p {
    color: $dark-color;
}

.a-div {
    background-color: $dark-color;
}
```

https://sass-lang.com/documentation/variables

SASS

@mixin

Een mixin is een herbruikbaar stukje code dat je kan toevoegen aan een CSS statement. Voorbeeld: Je maakt een mixin waarin je het element een flexbox maakt en de items erin wil centeren. Deze kun je hergebruiken voor alle parent element die het zelfde gedrag moeten vertonen.

```
@mixin flexCenter {
    display: flex;
    justify-content: center;
    align-items: center;
}

header {
    @include flexCenter();
    color: $dark-color;
}
```

https://sass-lang.com/documentation/at-rules/mixin

SASS

@mixin($variable)

Een mixin kan ook variabelen ontvangen zodat we de mixin kunnen aanpassen zoals wij dat willen. Misschien hebben we een terugkerend stukje code nodig, maar is de width, height, color of iets anders soms net even anders. Het variabele gedeelte van de code kunnen we meegeven aan de mixin.

```
@mixin flexCenter($align) {
    display: flex;
    justify-content: center;
    align-items: $align;
}

header {
    @include flexCenter(center);
    color: $dark-color;
}
```

https://sass-lang.com/documentation/at-rules/mixin

SASS

nesting

In CSS kun je elementen selecteren die in andere elementen zitten. Bijvoorbeeld een p die in een header element zit kun je selecteren als (header p) of (header > p). Een link die in een list item van een ordered list zit kun je selecteren als (ol li a). Met SASS kun je de CSS nesten net als HTML elementen in een HTML bestand.

```
section {
    background-color: red;
    height: 100vh;
    h2 {
        color: white;
    }
    div {
        background-color: bisque;
        h3 {
            color: $color-dark;
        }
    }
}
```

https://sass-lang.com/documentation/style-rules#nesting

SASS

nesting pseudo

Nesting werkt ook met pseudo elements (::before, ::after) en pseudo classes (:hover, :active etc). Hier-voor gebruik je het & symbool.

```
button {
    color: white;
    padding: 12px 32px;
    background-color: $base-color;
    &:hover {
        padding: 12px 44px;
        background-color: red;
    }
}
```

https://sass-lang.com/documentation/style-rules#nesting

SASS

nesting Modifier

In een website heb je vaak elementen die een variant zijn van andere elementen. Hiervoorgebruiken we dan een combo class. In BEM termen noemen we dit een modifier.

In SASS kunnen we een stukje code extenden en de modifier genest bij de originele code typen.

```
.message {
    display: inline-block;
    background-color: aliceblue;
    &--alert {
        font-weight: 700;
        background-color: black;
        color: white;
    }
}
```

https://sass-lang.com/documentation/style-rules#nesting

SASS

Modulair werken

Bij grotere projecten zul je vele regels scss schrijven, dan is het handig om je CSS op te delen in meerdere bestanden (partials). Bijvoorbeeld een bestand voor basis code (*, html, body, section, container etc), een bestand voor tekst styling, een bestand voor navigatie, een bestand voor kleuren etc.

Een partial krijgt als bestandnaam een _ (underscore). bijvoorbeeld: _base.scss

Deze code kan je inladen met @use ‘base’.

https://sass-lang.com/guide#topic-4

```scss
// in _colors.scss

$base-color: #c6538c;
$color-dark: #212121;
```

```scss
// style.scss

@use 'colors';
```