

Les 11: Preprocessors - Frameworks

**HO
GENT**

Overzicht

- CSS preprocessors en CSS frameworks
- CSS custom properties (CSS variables)
- calc()

CSS in grotere projecten

- CSS was oorspronkelijk niet ontworpen voor grote projecten:
 - (oorspronkelijk) geen variabelen
 - CSS selectors met 'global scope'.
 - ...
- Om het schrijven van CSS-code voor grotere projecten meer beheersbaar en/of sneller te maken heeft men daarom allerlei methodologieën (bijv. BEM), tools (bijv. SASS) en libraries (bijv. Bootstrap) ontwikkeld.

CSS preprocessors

- Volgens de website <https://stateofcss.com/> is SASS momenteel de meest gebruikte CSS preprocessor (zie volgende slide).
- Enkele andere bekende CSS preprocessors zijn:
 - LESS
 - PostCSS

Pre-/Post-processors

Utilities that augment CSS.



Rankings

Export

Share

Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



CSS frameworks

- Volgens de website <https://stateofcss.com/> is Bootstrap nog steeds het meest gebruikte [CSS Framework](#) (zie volgende slide).
- Het meest trendy CSS framework is momenteel [tailwindcss](#).
- Enkele andere bekende CSS frameworks:
 - Material Design (<https://material.io/>)
 - Foundation (<https://get.foundation/>)

CSS Frameworks

Libraries that give you pre-made components and styles.



Rankings

Export

Share

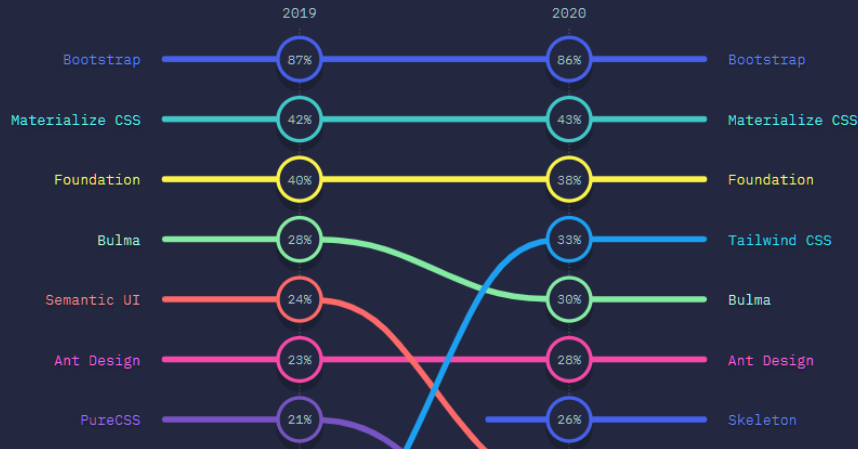
Satisfaction

Interest

Usage

Awareness

Satisfaction, interest, usage, and awareness ratio rankings.



SASS en BOOTSTRAP

- Deze laatste les bevat twee workshops. De eerste workshop **01_sass_workshop.pdf** bespreekt kort de CSS preprocessor [SASS](#) en in de tweede workshop **02_bootstrap_workshop.pdf** krijg je een inleiding in het werken met het CSS framework [Bootstrap](#).

CSS variables

CSS custom properties (variables)

- CSS custom properties worden ook wel CSS variables genoemd.
- Custom property names moeten beginnen met --, bijvoorbeeld `--primary-color`.
- Voor uitgebreide info zie [MDN – Using CSS custom properties \(variables\)](#).

CSS custom properties (variables)

- Het definiëren van een custom property gebeurt zoals bij andere CSS properties in een CSS rule. Voorbeeld :

```
:root {  
  --themecolor1: rgb(10, 10, 10);  
}
```

Meestal definieert men de properties op de [:root](#) pseudo-class.

- Om de waarde te gebruiken in een declaratie gebruik je de [var\(\)](#)-functie.

```
background-color: var(--themecolor1);
```

01-css-variables-calc

```
:root {  
  --themecolor1: rgb(10, 10, 10);  
  --themecolor2: rgb(45, 98, 113);  
  --themecolor3: white;  
  --font: Calibri, sans-serif;  
}  
  
...  
body {  
  background-color: var(--themecolor2);  
  font-family: var(--font);  
  font-size: 1.1rem;  
}
```

Nog een voorbeeld:

```
:root {  
  /* palet colors */  
  --themecolor1: rgb(10,10,10);  
  --themecolor2: rgb(45, 98, 113);  
  --themecolor3: white;  
  /* colors */  
  --body-bgcolor: var(--themecolor2);  
  --container-bgcolor: var(--themecolor3);  
  --heading1-color: var(--themecolor2);  
  --nav-link-color: var(--themecolor1);  
}  
...  
body {  
  background-color: var(--body-bgcolor);  
}
```

Bij de definitie van variabelen kan je ook de waarde van andere variabelen gebruiken.

calc()

**HO
GENT**

Rekenen met calc()

- Met de calc()-functie kan je berekenen uitvoeren in een declaratie:

```
width: calc(100% - 30px);
```

```
width: calc(var(--variable) + 20px);
```

- Je kan de standaardoperatoren +, -, * en / gebruiken, alsook ronde haakjes en je kan verschillende eenheden combineren.
- **Let op** de + en de - operator moeten voorafgegaan en gevolgd worden door 'whitespace'.

01-css-variables-calc

- We wensen dat de webpagina de volledige viewport vult in de hoogte:

```
.container {  
  ...  
  min-height: calc(100vh - 8px * 2);  
  max-width: 1000px;  
  margin: auto;  
}
```

