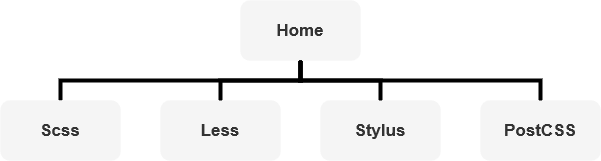
Sophisticated CSS

Sitemap



Home

|  |  |
| --- | --- |
| **Doel** | Informeren over CSS processors |
| **Primaire CTA** | Door laten klikken naar verdere informatie |
| **Informatie** | * Pitch voor gebruik van preprocessors * Overzicht van bekende preprocessors |

**Structuur**

* Header
  + Kort voorbeeld verschil ruwe CSS
* Uitleg van pre-/postprocessors
  + Doel
  + Hoe
  + Flow van pre → post
  + Wanneer
* Blokken met preprocessors:
  + Sass
  + Less
  + Stylus
* Blokken met postprocessors:
  + PostCSS
* Risico’s en discussie
  + Vendor lock-in
  + Meer afstand tussen code en ontwikkelaar
* Links naar aanvullende bronnen

Sass

|  |  |
| --- | --- |
| **Purpose** | Display information about sass |
| **Primaire CTA(s)** | -Read about other preprocessor |
| **Secundaire CTA(s)** | -Download Sass |
| **Information** | -General Information about sass  -How to use sass  -Note features of Sass  -Advantages / Disadvantages of sass |

**Structure**

* **Header**
  + Syntactically Awesome Style Sheets - Sass
  + Quote from site
* **Introduction**
  + Extension to css
  + Most popular
  + Makes large css easier to maintain
  + Will take scss file and turn it into a css file
  + // (-build on ruby)
* **Features**
  + Variables with $
  + Better Nesting
  + Partial files: \_partialfile.scss -> @use
  + Modularity with Partial files
  + Mixins
  + Extend/Inheritance
  + Operators
* **Pro’s & Con’s** 
  + **Advantages**
    - Many new features
    - Easy use
    - Simplifies code
    - Large community
  + **Disadvantages**
    - Many applications are paid
    - chance to become reliant on framework
* **Blokken die linken naar andere plekken op de website**
* **Start using sass**
  + **Link naar de website**
  + **Blok met installatie**
    - can be downloaded as application or for console
    - sass must be downloaded:<https://sass-lang.com/>
    - All code written in .scss files
    - CMD: sass input.scss output.css builds file
    - --watch flag will update .css automatically

Less

|  |  |
| --- | --- |
| **Purpose** | Display information about less |
| **Primaire CTA(s)** | -Read about other preprocessor |
| **Secundaire CTA(s)** | -Download Less |
| **Information** | -General Information about less  -Note features of less  -Advantages / Disadvantages of less  -How to use less |

**Structure**

* **Header**
  + Leaner Style Sheet - Less
  + Quote
* **Introduction**
  + Extension to css
  + Backwards compatible
  + Uses less.js file to convert
  + takes .less files and turns it into .css
* **Features**
  + Variables
  + Mixins (Namespaces / Accessors)
  + better Nesting
  + Operators
  + Escaping (\*)
  + Functions
  + Maps (dictionaries)
  + Comments
  + Importing (\*)
  + Scope (\*)

(\*) = Not as important, maybe do not talk about on site

* **Comparing less**
  + **Advantages**
    - Feature Rich
    - Backwards compatible
    - Third party applications
  + **Disadvantages**
    - Complex
    - Uses scripts
* **How to Use Less**
  + **Link naar de website**
  + **Blok met installatie**
    - Install on cmd using npm: npm install less -g
    - convert with lessc [flag] <source> <destination>
    - Convert can happen in browser
      * link .less stylesheet
      * use less.js script
      * many options

# Stylus

|  |  |
| --- | --- |
| **Doel** | * Display information about Stylus |
| **Primaire CTA(s)** | * Read about other preprocessorrs |
| **Secundaire CTA(s)** | * Download Stylus |
| **Informatie** | * General information about Stylus * Note Features of stylus * Advantages / Disadvantages of Stylus * How to use Stylus |

**Structuur**

* **Header**
  + Stylus
  + “EXPRESSIVE, DYNAMIC, ROBUST CSS”
  + Quote from Stylus: “Stylus is an innovative stylesheet language that compiles down to CSS. Inspired by SASS, Stylus is built with node.js and capable of running in the browser.”
* **Introduction**
  + Design influenced by sas.
  + Built with node.js
  + Running in browser
  + .styl to .css
  + “The stylus was launched in 2010 by former Node JS developer TJ Holowaychuk, nearly 4 years after the release of Sass and 1 year after the release of LESS. The stylus is written Node JS and fits perfectly with JS stack. The stylus was heavily influenced by the logical prowess of the Sass and simplicity of LESS. Even though Stylus is still popular with Node JS developers, it hasn’t managed to carve out a sizeable share for itself.”
* **korte “recensie” :**
  + “Everything in Stylus is expressive and to make it so, Stylus uses a plethora of operators, keywords, and sugary syntax to make you feel right at home.” - Manan Tank
* **Features**
  + Better nesting
  + Inheritance
  + Optional semicolons, braces
  + Mixins
  + Variables
  + Operators
  + Comments
  + Keyword arguments “Kwargs”
  + Importing (\*)
  + Escaping (\*)
* **Pro’s & Con’s**
  + **Advantages**
    - extremely permissive syntax
    - supports multiple syntaxes, mix and match
    - extremely powerful built-in functions, capable of handling heavy computing
    - extensive logical prowess of Sass with the simplicity of LESS
  + **Disadvantages**
    - extremely permissive syntax ;)
    - Not as well known
* **Blokken to other preprocessors**
* **How to use Stylus**
  + **Link naar de website**
  + **Blok met installatie**

# PostCSS

|  |  |
| --- | --- |
| **Doel** | * Display information about PostCSS |
| **Primaire CTA(s)** | * Read about other preprocessorrs |
| **Secundaire CTA(s)** | * Download PostCSS |
| **informatie** | * General information about PostCSS * Note Features of PostCSS * Advantages / Disadvantages of PostCSS * How to use PostCSS |

**Structuur**

* **Header**
  + PostCSS
  + “A tool for transforming CSS with JavaScript”
  + Quote from PostCSS: “PostCSS is a tool for transforming styles with JS plugins. These plugins can lint your CSS, support variables and mixins, transpile future CSS syntax, inline images, and more. PostCSS is used by industry leaders including Wikipedia, Twitter, Alibaba, and JetBrains.”
* **Introduction:**
  + what PostCSS really is: a postprocessor, a preprocessor, or something else, which, generally speaking, doesn’t have a particular significance
  + “Developed by Andrey Sitnik, creator of Autoprefixer, PostCSS was launched as a method to use JavaScript for CSS processing. PostCSS on its own is simply an API, which, when used with its vast ecosystem of plugins, offers powerful abilities. To provide helpful debugging, source maps are generated, and an abstract syntax tree (AST) exists to help us understand where and how code is being transformed.”
  + “Because PostCSS uses the Node.js framework, the abilities of the language and tools can be easily modified and customized as needed. Other tools such as Sass and LESS limit the abilities of the system to the methods available when the compiler was written.”
  + Due to its use of an API, PostCSS allows us to create custom plugins and tools for any features they may need. This modular platform design makes the tool neutral, which in turn focuses features on project requirements. PostCSS is agnostic to language format and allows for the syntax style of different preprocessors, like Sass and LESS, if needed.
  + PostCSS is a framework to develop CSS tools It can be used to develop a template language such as Sass and LESS.
  + “PostCSS is not a preprocessor per se; it doesn’t transform CSS. As a matter of fact, it doesn’t do much by itself at all. What it does is provide a CSS parser and a framework for creating plugins that can analyse, lint, handle assets, optimise, create fallbacks, and otherwise transform parsed CSS. PostCSS parses CSS into an abstract syntax tree (AST), passes it through a series of plugins, and then concatenates back into a string.”
  + “PostCSS is a JavaScript tool which will read your CSS with special additional syntax and will process it and return normal CSS code.”
  + Korte (relevante) geschiedenis (waarom ontstaan, nog geupdate etc)
* **korte “recensie”:**
  + “I’m not compelled by PostCSS because I don’t love having to cobble together the processing features that I want. That also has the bad side effect of making the process of writing CSS different across projects. Plus, I don’t like the idea of preprocessing away modern features, some of which can’t really be preprocessed (e.g. custom properties cannot be preprocessed). But I did love Autoprefixer when we really needed that, which is built on PostCSS.” - Chris Coyier on Oct 19, 2020
  + “When I use PostCSS more and more, I sometimes think that I could stop using preprocessors completely. For now I like using Stylus and PostCSS in one stack. It’s really cool. I’m not sure but maybe in the near future I'll drop Stylus.” - Julian Ćwirko on Jan 01, 2016
* **Features**
  + Create own custum plug-ins for any features you might need
  + Various plug-ins with various features
  + Can convert CSS to other versions of itself, use features of future CSS versions that are not yet supported by all browsers.
* **Pro’s & Con’s**
  + **Advantages**
    - Able to use it with preprocessors
    - Very modular, install only the things you really need, flexible
    - It is integrated in many well known frameworks
    - fast(?)
    - Doesn't force designers to learn a new syntax
    - works in the browser directly without the need to compile it beforehand
  + **Disadvantages**
    - Not very powerful in itself, you need to use or create plug-ins
    - duplication of functionality tussen plug-ins
    - problems with syntax highlighting in the popular IDE (2016?)
    - Have to look for plug-ins with all functions you want
    - Plug-ins don’t always work together properly when based on different PostCss versions
    - Run plug-ins in certain order
* **Blokken to other preprocessors**
* **Start using PostCSS**
  + **Link naar de website**
  + **Blok met installatie**