

Practical PostCSS

Hello, World!

What is a CSS Preprocessor?

The basics

Add desirable functionality to CSS

- variables
- visual hierarchy (nesting)
- partials
- importing
- grouped property declarations (mixins)
- inheritance (extend)
- math



What is a CSS Preprocessor?

The basics

Add desirable functionality to CSS4

- variables
- visual hierarchy (nesting)
- partials
- importing
- grouped property declarations (mixins)
- inheritance (extend)
- math



What is PostCSS?

The basics

- Out of box, only the tools you need to transform CSS
 - No CSS functionality or modifiers
- Completely plugin driven
- Written entirely in JavaScript
- Fast!
 - 1.2x vs libsass
 - 2.7x vs less

How does PostCSS stack up?

Tool comparison

Sass, Less, SCSS

- Consistent feature set
- libsass, C++
- Maturity, adoption

PostCSS

- Customization, flexibility
- JavaScript
- Faster ...negligible?

How do I use it?

Bringing PostCSS into your workflow

1. Integrate with Gulp, Grunt, webpack, Broccoli, Brunch, ENB, Fly, Stylus, Duo, Connect / Express
2. Add plugins

But what if...

You can probably still use PostCSS

- postcss-js

React inline styles, Radium, Free Style, CSS-in-JS

- html-postcss

<style> tags and inline style attributes

- postcss-cli

Command line interface

Emulating Sass

Portability win

Can we reproduce Sass functionality using PostCSS?

- variables
- nesting
- partials
- importing
- mixins
- extend
- math

Emulating Sass

Portability win

Can we reproduce Sass functionality using PostCSS?

- | | |
|-------------|------------------------|
| - variables | postcss-simple-vars |
| - nesting | postcss-nested |
| - partials | postcss-partial-import |
| - importing | postcss-partial-import |
| - mixins | postcss-sassy-mixins |
| - extend | postcss-simple-extend |
| - math | postcss-calc |

Emulating Sass

Portability win

PreCSS is a collection of plugins that allows you to use Sass-like markup.

Variables, nesting, partials, conditionals, loops, mixins, extends, imports, property lookup, root and “other goodies”

PostCSS Plugins

Gracefully degrading

- autoprefixer

Vendor prefixes from CanIUse based on support matrix

- oldie

Internet Explorer fallbacks

PostCSS Plugins

Looking forward

cssnext allows us to use the latest CSS syntax today.
Future proof - no more waiting on browser support.

var(), color(), hwb(), gray(), #rrggbbaa, font-variant, rem
units, :not(), improved calc(), custom media queries...

Uses autoprefixer and provides fallbacks

PostCSS Plugins

Tooling

- cssnano

CSS minification

- stylelint

Modern CSS linting based on rules

- postcss-colorblind

Simulate colorblindness

Writing a Plugin

Do one thing, and do it well

```
module.exports = postcss.plugin('postcss-demo', function(opts) {  
  opts = opts || {};  
  
  return function(css) {  
    css.eachInside(function(node) {  
      if (node.type === 'decl' && node.prop === 'color') {  
        node.value = (opts.color || 'green');  
      }  
    })  
  };  
});
```

So... should I use it?

Perhaps!

"But I can't shake the concern that dropping a tool as proven and popular as Sass — one of the few frontend tools we can truly, indefinitely rely on — is like setting myself adrift, and maybe unnecessarily."

~ @davidtheclark

CodePen recently added support for PostCSS

Used by Google, Twitter, Alibaba, Shopify

Resources

Check these out

- PostCSS.parts
- [PostCSS Github](https://github.com/postcss/postcss)
- [Sass Documentation](http://sass-lang.com/documentation)
- <http://davidtheclark.com/its-time-for-everyone-to-learn-about-postcss/>
- <http://davidtheclark.com/excited-about-postcss/>
- <http://benfrain.com/breaking-up-with-sass-postcss/>
- <http://studiorgb.uk/from-sass-to-postcss/>
- <http://www.sitepoint.com/build-css-preprocessor-postcss/>

Thank you!

bit.ly/sblaurock-talks