

PostCSS

Lightning Talk

Merle Ehm

February 17th, 2017

ERGOVIA

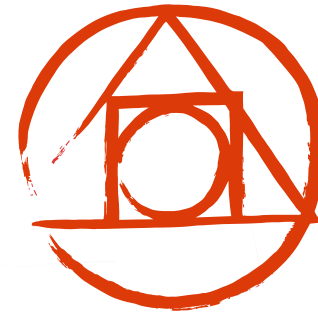
Großer Licht-Tm
Fl(2)12s 63m 24M

Tschöden

Yet another preprocessor for CSS? NO!

PostCSS is a tool for transforming CSS with JavaScript

- Integrates in JavaScript based build tools like Grunt, Gulp, Webpack and so on
- Uses modular architecture
- Offers many useful plugins



PLUGIN: cssnext



“Use tomorrow’s CSS syntax, today.”

- Write future-proof CSS
- No need to learn specific preprocessor syntax like SASS or LESS

```
1  /* CSS INPUT */
2
3  :root {
4  ■  --red: #d33;
5  }
6  a:hover {
7  ■  color: color(var(--red) a(54%));
8  }
9
```

```
15  /* CSS OUTPUT */
16
17  a:hover {
18  ■  color: #dd3333;
19  ■  color: rgba(221, 51, 51, 0.54);
20  }
21
```

PLUGIN: Autoprefixer



“Increase code readability”

- Adds vendor prefixes if needed



```
1  /* CSS INPUT */  
2  
3  :-fullscreen {  
4  }
```

```
8  /* CSS OUTPUT */  
9  
10 :-webkit-full-screen {  
11 }  
12 :-moz-full-screen {  
13 }  
14 :-fullscreen {  
15 }
```

PLUGIN: CSS Modules

CSS MODULES

“The end of global CSS”

- No need to worry about too generic names like “.name {...}”
- But only useful for HTML generation in JavaScript

```
1  /* CSS INPUT from logo.css */
2
3  .name {
4  ■   color: gray;
5  }
```

```
9  /* CSS OUTPUT */
10
11  .logo_name_SVK0g {
12  ■   color: gray;
13  }
```

```
1  /* Use in JavaScript file */
2
3  import styles from "./logo.css";
4
5  element.innerHTML = '<div class="' + styles.name + '"></div>';
```

PLUGIN: stylelint



“Avoid errors in your CSS”

- Lints your CSS and enforces consistent conventions
- Supports latest CSS syntax

`visual.css`

2:12 ✖ Unexpected invalid hex color "#4f"

4:1 ⚠ Expected ".foo.bar" to have a specificity no more than "0,1,0"

6:13 ✖ Unexpected unit "px" for property "margin"

7:17 ✖ Expected single space after ",", in a single-line function

color-no-invalid-hex

selector-max-specificity

declaration-property-unit-blacklist

function-comma-space-after

Großer Licht-Tm
Fl(2)125 63m 34M

Tuchhafen

Setup in Webpack

Webpack

Use `postcss-loader` in `webpack.config.js`:

```
module.exports = {
  module: {
    loaders: [
      {
        test: /\.css$/,
        loaders: [
          'style-loader',
          'css-loader?importLoaders=1',
          'postcss-loader'
        ]
      }
    ]
  }
}
```

Then create `postcss.config.js`:

```
module.exports = {
  plugins: [
    require('precss'),
    require('autoprefixer')
  ]
}
```

Thank you!

Shall we use PostCSS?

