

# Переход с Sass на PostCSS

# Обо мне

Станислав Матвеев, ведущий инженер, ООО  
«Кадис». Пишу на PHP, JavaScript.

# SASS

SASS - Syntactically Awesome StyleSheets.

SASS - это расширение CSS, которое позволяет писать более читаемый и поддерживаемый CSS-код, за счёт использования дополнительных возможностей.  
Полностью совместим с CSS.

# SASS. Переменные

Для обозначения переменных в SASS используется символ \$:

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;  
body {  
    font: 100% $font-stack;  
    color: $primary-color;  
}
```

# SASS. Вложенность

Позволяет писать вложенные CSS-селекторы более удобным и читаемым способом:

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;}  
  li { display: inline-block; }  
  a {  
    display: block;  
    text-decoration: none;}  
}
```

# SASS. Импорт

Позволяет составлять результирующий стилевой файл из нескольких:

```
// _reset.scss
Html, body,
ul,
ol {
  margin: 0;
  padding: 0;}
// base.scss
@import 'reset';
body {
  background-color: #efefef;}
```

# SASS. МИКСИНЫ

Миксины позволяют сделать группу CSS правил, которые можно использовать несколько раз.

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius;  
}  
.box { @include border-radius(10px); }
```

# PostCSS

PostCSS — это инструмент для трансформации CSS-стилей с помощью Javascript плагинов.

Эти плагины достаточно разнообразны по возможностям: линтинг кода, использование будущих спецификаций CSS и многое другое.



# PreCSS

Препроцессор с SASS-подобным синтаксисом

<https://jonathantneal.github.io/precss/>

# Autoprefixer

Плагин, который автоматически добавляет вендорные префиксы к CSS-свойствам.

# CssNext

PostCSS CssNext плагин позволяет использовать синтаксис будущих спецификаций CSS не беспокоясь о совместимости

# CssNext. Кастомные СВОЙСТВА

CssNext поддерживает кастомные свойства:

```
:root {  
  --mainColor: red;  
}  
  
a {  
  color: var(--mainColor);  
}
```

# CSSNext. Наборы КАСТОМНЫХ СВОЙСТВ

```
root {  
  --danger-theme: {  
    color: white;  
    background-color: red;  
  };  
}  
.danger {  
  @apply --danger-theme;  
}
```

# CSSNext. Вложенность

Позволяет писать вложенные селекторы более читаемым способом:

```
a {  
    & span {  
        color: white;  
    }  
    @nest span & {  
        color: blue;  
    }  
}
```

# Lost

Система сеток. Строи сетку на основе свойства calc()

<https://github.com/peterramsing/lost>

# Stylelint

Плагин для проверки CSS-кода соответствию некоторым правилам

<https://github.com/stylelint/stylelint/blob/master/docs/user-guide/postcss-plugin.md>



# Конфигурация SASS

<https://bitbucket.org/saidekdevs/webpack4-sass/src>

# Конфигурация PostCSS

<https://bitbucket.org/saidekdevs/webpack4-postcss/src>

# CSSNano

Плагин для минификации CSS

<https://github.com/ben-eb/cssnano>

# Делаем свой плагин

<https://dockyard.com/blog/2018/02/01/writing-your-first-postcss-plugin>

# ССЫЛКИ

<https://sass-lang.com/> - SASS

<http://postcss.org/> - PostCSS

<http://cssnext.io/> - CSS Next

<https://webdesign.tutsplus.com/series/postcss-deep-dive--cms-889>

<https://www.postcss.parts/>

**СПАСИБО!**