WebCademy.ru
ОБУЧЕНИЕ ВЕБ-РАЗРАБОТКЕ

Препроцессоры

Less, Scss, PUG

авто-обновление верстки проекта

УСКОРЕНИЕ

НАПИСАНИЯ CSS КОДА

css препроцессор

LESS

Возможности

- Переменные
- Подмешивания
- Вложенные правила
- Вложенные директивы
- Операции
- Функции
- Пространство имен
- Комментарии
- Импорт

Переменные в LESS

.less

```
@blue: #5B83AD;
#header {
  color: @blue;
}
```

.css

```
#header {
  color: #5B83AD;
}
```

Mixins - подмешивания

.less

```
.bordered {
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}

#menu a {
  color: #111;
  .bordered;
}

.post a {
  color: red;
  .bordered;
}
```

.css

```
.bordered {
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}

#menu a {
  color: #111;
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}

.post a {
  color: red;
  border-top: dotted 1px black;
  border-bottom: solid 2px black;
}
```

Nested rules – вложенные правила

.less

```
#header {
  color: black;

  .navigation {
    font-size: 12px;
  }

  .logo {
    width: 300px;
  }

}
```

.css

```
#header {
  color: black;
}

#header .navigation {
  font-size: 12px;
}

#header .logo {
  width: 300px;
}
```

css препроцессор

SCSS

Зачем?

- Увеличение производительности
- Использование циклов и условий
- Использование массивов и объектов
- Создание и наследование шаблонов страниц
- Совершение простых математических вычислений

Вложенность

SCSS позволяет вкладывать CSS селекторы иерархично. Следовательно, вложенность делает ваш код более читабельным и лаконичным.

.scss

```scss
.parent {
    .first-children {
        background: #000;
    }

    .second-children {
        background: #fff;
    }
}
```

.css

```css
.parent .first-children {
    background: #000;
}

.parent .second-children {
    background: #fff;
}
```

Переменные и типы данных

Переменные– это самое простое, что поддерживается SassScript. Для их задания используется символ «$». Переменные доступны только в пределах того уровня вложенности селекторов, на котором они определены:

.scss

```scss
// объявим переменную главного цвета
$mainColorWhite: #f8f8ff;

// используем нашу переменную внутри селектора
.variable-example {
    color: $mainColorWhite;
}
```

.css

```css
.variable-example {
    color: #f8f8ff;
}
```

Импорт (@import)

Данная директива позволяет импортировать sass файлы, которые могут быть в дальнейшем объединены в один css файл. Переменные и миксины, объявленные в импортированном файле, могут использоваться в главном файле.

.scss

```
// импортирует весь код указанных файлов
@import'_auth.scss';
@import'_first.scss'';

// можно не указывать расширение - Sass так же определит нужные файлы
@import'_auth';
@import'_first';
```

Медиа (@media)

@media работает практически также, как и в привычном CSS: если директива вложена в css-правило, то при компиляции она будет поднята наверх таблицы стилей, а все селекторы, в которых была директива, переместятся внутрь.

.scss

```scss
$widthOfDesktop: 1800px;
$mainMedia: screen;

@media #{$mainMedia} {
  .media-example {
    @media (width: $widthOfDesktop) {
      margin-top: 20px;
    }
  }
}
```

.css

```css
@media screen and (width: 1800px) {
  .media-example {
    margin-top: 20px;
  }
}
```

Наследование (@extend)

Директива «@extend» позволяет наследовать наборы свойств от одного селектора к другому, что позволяет держать Scss -файлы «чистоте».

.scss

```
.example-extend {
    border: 1px solid #000;
    color: #000;
}
.good-message {
    @extend .example-extend;
    border-color: green;
}
.error-message {
    @extend .example-extend;
    border-color: red;
}
```

.css

```
.example-extend, .good-message, .error-message {
    border: 1px solid #000;
    color: #000;
}
.good-message {
    border-color: green;
}
.error-message {
    border-color: red;
}
```

Цикл (@while)

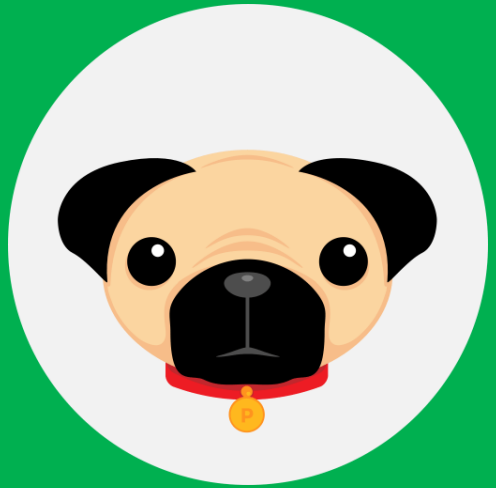
.scss

```scss
$i: 100;
@while $i > 0 {

    .item-#{$i} {
        background: adjust-hue(#00BFFF, $i);
        height: #{$i}px;
        width : $i * 2px;
    }

    $i: $i - 25;
}
```

.css

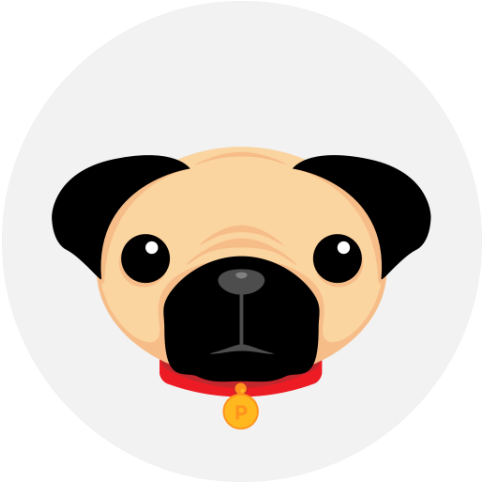
```css
.item-100 {
    background: #ea00ff;
    height: 100px;
    width: 200px;
}
.item-75 {
    background: #8000ff;
    height: 75px;
    width: 150px;
}
.item-50 {
    background: #1600ff;
    height: 50px;
    width: 100px;
}
.item-25 {
    background: #0055ff;
    height: 25px;
    width: 50px;
}
```

html шаблонизатор
PUG



Зачем?

- Вынесение общих блоков в отдельные файлы
- Использование циклов и условий
- Использование массивов и объектов
- Создание и наследование шаблонов страниц
- Определение собственных функций (примесей)
- Ремонтопригодность
- Исчезает возможность не закрыть тег



Прощай, закрывающий тег!

Что бы указать тег в jade, необходимо просто написать его имя.

Если тег не указан то считается что это div.

.pug

```
span.text


.block
```

.html

```
<span class="text"></span>


<div class="block"></div>
```

Содержимое

Передать содержимое блока можно несколькими способами.

.pug

```
// обычно делают так
.block Текст внутри блока

// а можно еще так
.block
  | Текст внурти блока

// или так
.block.
    Текст внутри блока
```

.html

```
<div class="block">
        Текст внутри
</div>
```

Вложенность блоков

Что бы указать вложенность блоков, нужно перенести блок на новую строку и “отбить” табуляцией.

.pug

```
.block
    .inner-block Текст внутреннего блока

.sibling-block
    | Текст соседнего блока
```

.html

```
<div class="block">
    <div class="inner-block">
        Внутренний блок
    </div>
</div>

<div class="sibling-block">
    Соседний блок
</div>
```

Атрибуты

Атрибуты передаются в скобках перед определением класса.

.pug

```
img(src="", alt="второй атрибут").img-class
```

.html

```
<img class="img-class" src="" alt="второй атрибут"/>
```

Include файлов

Pug позволяет подключать сторонние файлы при помощи директивы include (расширение можно не указывать). Например: макетов много - хедер один.

.jade

```
include ../path/to/file
```

Include и Вынесение общих блоков

Pug позволяет подключать сторонние файлы при помощи директивы include (расширение можно не указывать). Например: макетов много - хедер один.

header.pug

```
head
    title My Site
    script(src='/javascripts/jquery.js')
    script(src='/javascripts/app.js')
```

footer. pug

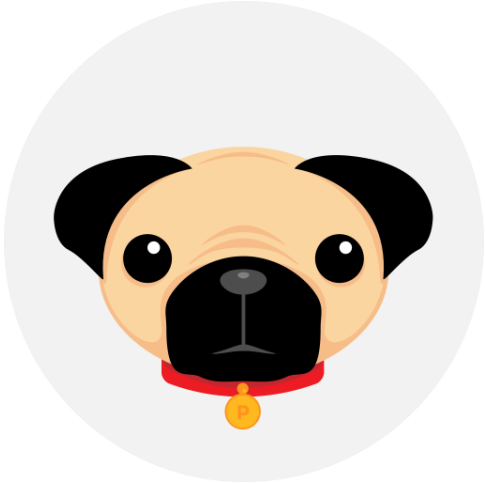
```
#footer
    p Copyright (c) foobar
```

index. pug

```
doctype html
Html
    include ./includes/head.jade
    body
    h1 My Site
    p Welcome to my super lame site.
    include ./includes/footer.jade jade
```

Что еще умеет PUG

- Миксины (мини шаблоны для кнопок, форм, карточек..)
- Циклы
- Условия
- Блоки в шаблонах
- Базовые шаблоны
- Переменные
- Массивы и объекты с данными



Ссылки

LESS

- http://lesscss.org/
- http://lesscss.org/less-preview/

SASS

- https://sass-scss.ru/
- https://www.sassmeister.com/

PUG

- https://pugjs.org/api/getting-started.html
- https://pughtml.com/
- http://html2jade.org/

Практика