



HTML и CSS. Уровень 3.

Продвинутые методологии и инструменты верстки



Модуль 2. Препроцессоры

Препроцессоры

Препроцессор — это инструмент, преобразующий код из одного синтаксиса в другой.

Обычно, на вход препроцессора поступает код, написанный с использованием синтаксических конструкций, понятных этому препроцессору.

Pug (Jade)

(NodeJS препроцессор для HTML)

Установка Pug

- Установить Node.js <https://nodejs.org/en/download/>
- Установить Pug-cli (через командную строку)

```
npm install pug-cli -g
```

```

doctype html
html(lang="en")
  head
    title= pageTitle
    script(type='text/javascript').
      if (foo) {
        bar(1 + 5)
      }
  body
    h1 Jade - node template engine
    #container.col
      if youAreUsingJade
        p You are amazing
      else
        p Get on it!
    p.
      Jade is a terse and simple
      templating language with a
      strong focus on performance
      and powerful features.

```

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Jade</title>
    <script type="text/javascript">
      if (foo) {
        bar(1 + 5)
      }
    </script>
  </head>
  <body>
    <h1>Jade - node template engine</h1>
    <div id="container" class="col">
      <p>You are amazing</p>
      <p>
        Jade is a terse and simple
        templating language with a
        strong focus on performance
        and powerful features.
      </p>
    </div>
  </body>
</html>

```

Тэги в Pug

```
.div  
  p Hello!  
  p World!
```

Тэги в html

```
<div>  
  <p>Hello!</p>  
  <p>World!</p>  
</div>
```


Атрибуты в Pug

```
div(class="movie-card", id="oceans-11")  
  img(src="/img/oceans-11.png", class="movie-poster")
```

Атрибуты в html

```
<div class="movie-card" id="oceans-11">  
    
</div>
```

Переменные в Pug

- var title = "On Dogs: Man's Best Friend";
- var author = "enlore";
- var theGreat = "escape!";

h1= title

p Written with love by #{author}

p This will be safe: #{theGreat}

Переменные в html

<h1>On Dogs: Man's Best Friend</h1>

<p>Written with love by enlore</p>

<p>This will be safe: escape!</p>

Миксины в Pug

```
mixin pet(name)  
  li.pet= name  
ul  
//Включение миксина  
+pet('cat')  
+pet('dog')  
+pet('pig')
```

Миксины в html

```
<ul>  
  <li class="pet">cat</li>  
  <li class="pet">dog</li>  
  <li class="pet">pig</li>  
</ul>
```

Подключение файлов

```
include includes/head.pug
```

Преобразование pug в html

В командной строке при нахождении в дирректории с файлом

```
pug filename.pug
```

<https://pugjs.org>

Stylus

(препроцессор для CSS)

Установка Stylus

➤ Установить Node.js <https://nodejs.org/en/download/>

➤ Установить Stylus (через командную строку)

```
npm install stylus -g
```



```
body
  font 14px/1.5 Helvetica, arial, sans-serif
  button
    button.button
      input[type='button']
      input[type='submit']
        border-radius 5px
```

```
body {
  font: 14px/1.5 Helvetica, arial, sans-serif;
}
body button,
body button.button,
body input[type='button'],
body input[type='submit'] {
  border-radius: 5px;
}
```

Переменные в Stylus

```
mycolor = #5B83AD  
newcolor = mycolor + #111;
```

```
header  
  color newcolor
```

Переменные в CSS

```
header {  
  color: #6c94be;  
}
```

Вложенность в Stylus

```
header
  color black
  .navigation
    font-size 12px
  .logo
    width 300px
```

Вложенность в CSS

```
header {
  color: black;
}
header .navigation {
  font-size: 12px;
}
header .logo {
  width: 300px;
}
```

Миксины в CSS

Миксины — это функции, позволяющие многократно использовать набор CSS-свойств во всей таблице стилей. Вместо того, чтобы писать один и тот же код несколько раз, мы можем написать его один раз в миксине и затем использовать

Миксины в Stylus

```
bordered(color)  
  border-top: dotted 1px color;  
  border-bottom: solid 2px color;
```

```
color = #111
```

```
//Включение миксина
```

```
#menu a  
  color color  
  bordered(color)
```

Миксины в CSS

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

Миксины как функции

```
.mixin() {  
  @width: 100%;  
  @height: 200px;  
}
```

```
.caller {  
  .mixin();  
  width: @width;  
  height: @height;  
}
```

Подключение файлов

```
@import "file.styl"
```

Преобразование Stylus в css

В командной строке при нахождении в дирректории с файлом

```
stylus -w file.styl -o file.css
```


<http://stylus-lang.com>

LESS

(препроцессор для CSS)

Установка LESS

(в командной строке)

```
$ npm install -g less
```

Переменные в LESS

```
@nice-blue: #5B83AD;
```

```
@light-blue: @nice-blue + #111;
```

```
#header {  
  color: @light-blue;  
}
```

Вложенность в LESS

```
#header {  
  color: black;  
  .navigation {  
    font-size: 12px;  
  }  
  .logo {  
    width: 300px;  
  }  
}
```

Миксины в LESS

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

//Включение миксина

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

Подключение файлов

```
@import file.less
```

Преобразование LESS в css

В командной строке при нахождении в дирректории с файлом

```
$ lessc filename.less filename css
```


<http://lesscss.org>

SCSS (SASS)

(препроцессор для CSS от Ruby)

Установка SASS

(в командной строке - требуется Ruby)

```
$ gem install sass
```

или...

SCSS или SASS?

.scss (Sassy CSS) — использует фигурные скобки, как и сам CSS

.sass (Syntactically Awesome Stylesheets) — отличается отсутствием фигурных скобок, в нём вложенные элементы реализованы с помощью отступов

Синтаксис SCSS похоже на LESS, кроме нескольких отличий

Переменные

Переменные в SCSS:

```
$font-stack: Helvetica, sans-serif;  
$primary-color: #333;
```

```
body {  
  font: 100% $font-stack;  
  color: $primary-color;  
}
```

Миксины

Миксины в SCSS:

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius;  
}
```

```
.box { @include border-radius(10px); }
```

Подключение файлов

```
@import 'file'
```


<http://sass-lang.com>
<https://sass-scss.ru>

