

Языки препроцессоров

Компиляция и Транспланация

- Компиляция – трансляция программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке
- Транспланация - трансляция программы, составленной на исходном языке высокого уровня, в эквивалентную программу на языке такого же уровня

Что такое Sass?

- Sass (Syntactically Awesome Stylesheet)
- CSS PreProcessor
- Let you use features that do not exist in CSS (functions, cycles)
- .sass и scss файлы транспилируются в CSS

Как компилировать Scss

- <https://www.sassmeister.com/> - онлайн компилятор
- <https://www.cssportal.com/scss-to-css/> - онлайн компилятор
- Можно работать с Scss в codepen
- Плагин на VS Code
- Консольный тул

✓.scss file

SCSS Sass

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

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

CSS

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

SCSS Sass

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

body
  font: 100% $font-stack
  color: $primary-color
```

CSS

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

.sass file

Переменные

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

Вложение

Sass

```
nav {  
  ul {  
    margin: 0;  
    padding: 0;  
    list-style: none;  
  }  
  
  li { display: inline-block; }  
  
  a {  
    display: block;  
    padding: 6px 12px;  
    text-decoration: none;  
  }  
}
```

CSS

```
nav ul {  
  margin: 0;  
  padding: 0;  
  list-style: none;  
}  
nav li {  
  display: inline-block;  
}  
nav a {  
  display: block;  
  padding: 6px 12px;  
  text-decoration: none;  
}
```

Модули - разделение файлов

- Для того чтобы файл не транспланировался, его имя должно начинаться с _

Sass

```
// _base.scss
$font-stack: Helvetica, sans-serif;
$primary-color: #333;

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

```
// styles.scss
@use 'base';

.inverse {
  background-color: base.$primary-color;
  color: white;
}
```

CSS

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

.inverse {
  background-color: #333;
  color: white;
}
```


Для чего могут быть полезны модули

- Разделение независимых компонент в разные файлы (компоненты для кнопок, меню и тд)
- Разделение стилей для разных страниц (`_about.scss`, `_home.scss`)
- Задача переменных в одном файле

@import vs @use

- @import также позволяет собирать CSS файл из модулей
- @import делает переменные, функции, миксины глобальными (засоряет пространство имен)
- => Лучше всегда использовать @use

МИКСИНЫ

- Создание миксина - `@mixin mixin_name(parameter : default_value) { ... }`
- Подключение миксина с помощью `@include`

```
@mixin theme($theme: DarkGray) {  
  background: $theme;  
  box-shadow: 0 0 1px rgba($theme, .25);  
  color: #fff;  
}  
  
.info {  
  @include theme;  
}  
  
.alert {  
  @include theme($theme: DarkRed);  
}  
  
.success {  
  @include theme($theme: DarkGreen);  
}
```

Наследование

```
%message-shared {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.message {  
  @extend %message-shared;  
}  
  
.success {  
  @extend %message-shared;  
  border-color: green;  
}  
  
.error {  
  @extend %message-shared;  
  border-color: red;  
}  
  
.warning {  
  @extend %message-shared;  
  border-color: yellow;  
}
```

Математические операции

- <https://sass-lang.com/documentation/operators/equality>

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 600px / 960px * 100%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 300px / 960px * 100%;  
}
```

```
.container {  
  width: 100%;  
}  
  
article[role="main"] {  
  float: left;  
  width: 62.5%;  
}  
  
aside[role="complementary"] {  
  float: right;  
  width: 31.25%;  
}
```

Условные операторы

```
@mixin triangle($size, $color, $direction) {  
  height: 0;  
  width: 0;  
  
  border-color: transparent;  
  border-style: solid;  
  border-width: $size / 2;  
  
  @if $direction == up {  
    border-bottom-color: $color;  
  } @else if $direction == right {  
    border-left-color: $color;  
  } @else if $direction == down {  
    border-top-color: $color;  
  } @else if $direction == left {  
    border-right-color: $color;  
  } @else {  
    @error "Unknown direction #{$direction}.";  
  }  
}  
  
.next {  
  @include triangle(5px, black, right);  
}
```

```
.next {  
  height: 0;  
  width: 0;  
  border-color: transparent;  
  border-style: solid;  
  border-width: 2.5px;  
  border-left-color: black;  
}
```

Мини-проект

- <https://codepen.io/bradtraversy/full/ExjmGdY>

Бонус: Работа с Figma

- <https://figmatemplate.com/cloud-budget-business-landing-page-figma/>