
SCSS

PREPROCESSOR

Препроцессоры Sass, LESS, Stylus

Дополнительный синтаксис для сокращения количества написанного кода

Требуется программа для компиляции в CSS

Есть и postprocessor, граница не всегда четкая



SASS vs нативного CSS

Вложенность

Функции

Операторы

Миксины

Управление порядком выполнения

Валидный документ CSS будет также валидным SCSS

Вложение



```
nav{
  display: flex;
  ul{
    list-style: none;
    li{
      color: white;
    }
  }
}
```



```
nav {
  display: flex;
}
nav ul {
  list-style: none;
}
nav ul li {
  color: white;
}
```

Переменные



```
$primary-color: blue;
```

```
html{  
  color: rgba($primary-color, 0.2);  
}
```



```
html {  
  color: rgba(0, 0, 255, 0.2);  
}
```

Амперсанд - &



```
a{  
  color: yellow;  
  &:hover{  
    color:yellowgreen;  
  }  
}
```



```
a {  
  color: yellow;  
}  
a:hover {  
  color: yellowgreen;  
}
```

& и БЭМ

```
.navbar{  
  display: flex;  
  &__link{  
    background-color: yellowgreen;  
  }  
}
```

```
.navbar {  
  display: flex;  
}  
.navbar__link {  
  background-color: yellowgreen;  
}
```

Mixins

```

@import url('http://fonts.googleapis.com/css?family=Open+Sans');

@mixin special {
  color: #fff;
  background-color: #befc6d;
  border: 1px dotted #59950c;
}
a.nav {
  @include special;
}
a.nav:hover {
  @include special;
  border: 1px yellow solid;
}
```

```

a.nav {
  color: #fff;
  background-color: #befc6d;
  border: 1px dotted #59950c;
}

a.nav:hover {
  color: #fff;
  background-color: #befc6d;
  border: 1px dotted #59950c;
  border: 1px yellow solid;
}
```


Вычисления



```
div{  
  width: (200px / 2);  
}
```



```
$i: 20px;  
  
div{  
  width: (200px / $i);  
}
```

@extend

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

```
.success, .error {  
  border: 1px solid #ccc;  
  padding: 10px;  
  color: #333;  
}  
  
.error {  
  border-color: red;  
}  
  
.success {  
  border-color: green;  
}
```

import -> @use



```
$color: red;
```

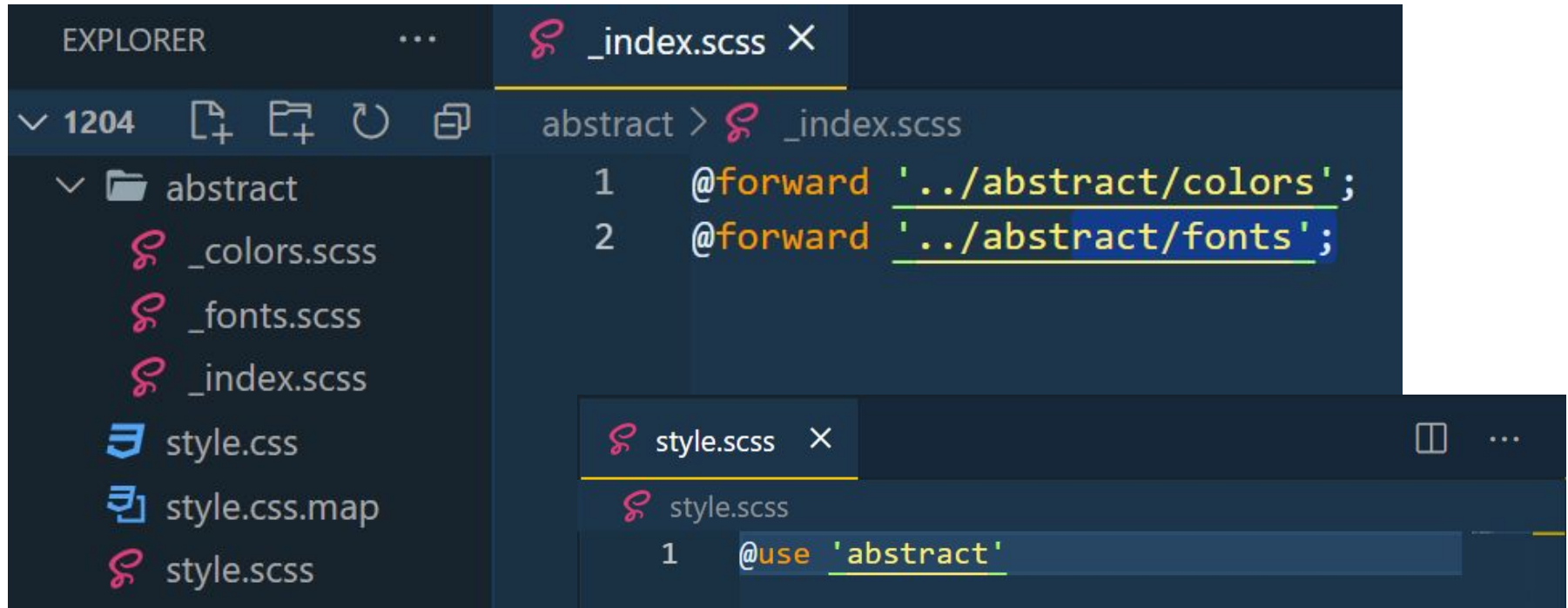


```
@use 'abstract/colors' as c;  
  
html{  
  color: c.$color;  
}
```



```
html {  
  color: red;  
}
```

@forward



Встроенные модули

```
• • •  
  
@use 'sass:map';  
  
$font-weights: (  
  "regular": 400,  
  "medium": 500,  
  "bold": 700);  
  
html{  
  font-weight: map.get($font-weights, "medium");  
}
```

```
• • •  
  
html {  
  font-weight: 500;  
}
```

partials

_base.scss

_layout.scss

_components.scss

_ в начале имени указывает на то, что это только часть кода и его не нужно генерировать.

С помощью @use подключаем в другие файлы

Пример структуры проекта

_base.scss

_layout.scss

_components.scss

main.scss

Base: все сбросы, переменные, миксины и другие служебные классы.

Layout: все связанное с макетом: контейнеры, сетки.

Components: все, что можно использовать повторно, например кнопки, навигационные панели, карты и т.д.

Main: только включает в себя вышеуказанные файлы.

Чем компилировать

SASS Live Compiler от Glenn Marks для VS Code

Scout (open source для Mac, Windows, and Linux) <https://scout-app.io/>

Prepros (лицензия напоминает WinRar) <https://prepros.io/>

Источники

Документация на английском - <https://sass-lang.com/>

Документация на русском - <https://sass-scss.ru/>

Переводная статья - <https://nuancesprog.ru/p/3378/>

Статья о структуре проекта на английском -
<https://itnext.io/structuring-your-sass-projects-c8d41fa55ed4>