

Препроцессоры

Зачем нужны препроцессоры

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

Препроцессор может полностью замещать синтаксические конструкции языка или частично расширять их, то есть добавлять новые конструкции.

Основная задача CSS-препроцессора — это предоставление удобных синтаксических конструкций для разработчика, чтобы упростить и ускорить разработку и поддержку стилей в проектах.

Зачем нужны препроцессоры

CSS препроцессоры преобразуют код, написанный с использованием препроцессорного языка, в чистый и валидный CSS-код, который нацелен на:

- читабельность для человека
- структурированность и логичность
- производительность

CSS препроцессоры являются синтаксическим сахаром (дополнения синтаксиса языка программирования, которые не вносят каких-то существенных изменений или новых возможностей, но делают этот язык более читабельным для человека)

Какие бывают препроцессоры

На данный момент можно выделить 3 самых популярных препроцессора:

- Less (<https://lesscss.org/>)
- Sass (SCSS) (<https://sass-lang.com/>)
- Stylus (<https://stylus-lang.com/>)

Какие бывают препроцессоры

За и против использования препроцессоров:

Плюсы:



- Подойдут для больших проектов
- Расширяют возможности обычного CSS
- Упрощают работу с однотипным кодом
- Понятная и логичная структура

Минусы:



- Нет смысла использовать в маленьких проектах и простых страницах
- На выходе препроцессора получается CSS-код, который сложно прочесть человеку
- Нужно изучать дополнительно помимо CSS
- У каждого препроцессора разных синтаксис

Препроцессор LESS

LESS(Leaner Style Sheets) - компактная таблица стилей. Документация - <https://lesscss.org/>

Может работать на стороне клиента (браузер) и на стороне сервера (например, под управлением Node.js)

Подключение на стороне клиента:

- сохранить CSS-код в файл с расширением .less
- добавить ссылку на less-файл на веб-странице (так же как CSS-файл) с атрибутом `rel="stylesheet/less"`, а затем файл `less.js`, который можно загрузить с www.lesscss.org

```
<link rel="stylesheet/less" type="text/css" href="/styles.less" />  
<script src="/less.js" type="text/javascript"></script>
```

Этот метод работает только в современных браузерах и подходит только для использования на локальном сервере. Для рабочего сайта необходима предварительная компиляция с применением Node.js на веб-сервере.

Препроцессор LESS

Возможности LESS:

- Вложенные правила.

LESS даёт возможность вкладывать определения вместо либо вместе с каскадированием

Код CSS:

```
.header h1{  
    font-size: 20px;  
}  
.header p{  
    font-size: 14px;  
}
```

Код LESS:

```
.header {  
  h1 {  
    font-size: 20px;  
  }  
  p {  
    font-size: 14px;  
  }  
}
```

Препроцессор LESS

Возможности LESS:

- Переменные

Less позволяет использовать переменные. Перед именем переменной ставится @. В качестве знака присваивания используется :

Код LESS:

```
@color: #4D926F;

.header {
    color: @color;
}

h1 {
    color: @color;
}
```

Код CSS:

```
.header {
    color: #4D926F;
}

h1 {
    color: #4D926F;
}
```


Препроцессор LESS

Возможности LESS:

- Примеси

Примеси позволяют переиспользовать один и тот же код в разных местах, тем самым делают код чище, понятнее и упрощают его изменение:

Код LESS:

```
.rounded-corners (@radius: 4px) {  
    border-radius: @radius;  
}  
  
.header {  
    .rounded-corners;  
}  
  
.card {  
    .rounded-corners(10px);  
}
```

Код CSS:

```
.header {  
    border-radius: 4px;  
}  
  
.card {  
    border-radius: 10px;  
}
```

Препроцессор LESS

Возможности LESS:

- Использование функций и операций

Less позволяет использовать операции и функции. Благодаря операциям можно складывать, вычитать, делить и умножать значения свойств и цветов.

Код LESS:

```
@padding: 5px;

.header {
  padding: @padding;
}

.card {
  padding: (@padding * 2);
}
```

Код CSS:

```
.header {
  padding: 5px;
}

.card {
  padding: 10px;
}
```

Препроцессор SASS

SASS - Syntactically Awesome Stylesheets. Документация - <https://sass-lang.com/>

Sass может быть использован тремя способами: как командной строкой, как отдельный модуль Ruby и как плагин.

Sass имеет 2 синтаксиса:

- sass - отличается отсутствием фигурных скобок, вложенные элементы реализованы с помощью отступов
- scss - использует фигурные скобки, как и сам CSS

Препроцессор SASS

Возможности SASS:

- Вложенные правила.

Код CSS:

```
.header h1{  
    font-size: 20px;  
}  
.header p{  
    font-size: 14px;  
}
```

Код Sass:

```
.header  
  h1  
    font-size: 20px  
  
  p  
    font-size: 14px
```

Препроцессор SASS

Возможности SASS:

- Импорт файлов.

Импорт файлов через директиву `@import`. Вместо создания отдельного HTTP-запроса Sass импортирует указанный в директиве файл в тот, где он вызывается, т.е. на выходе получается один CSS-файл, скомпилированный из нескольких фрагментов.

Код CSS:

```
html,
body {
  margin: 0;
  padding: 0;
}
body {
  background-color: #efefef;
}
```

Код Sass:

```
@import reset
body
  background-color: #efefef
```

файл reset.sass

```
html,
body
  margin: 0
  padding: 0
```

Препроцессор SASS

Возможности SASS:

- Примеси

Примеси позволяют переиспользовать один и тот же код в разных местах, тем самым делают код чище, понятнее и упрощают его изменение

Код CSS:

```
.box {  
  -webkit-transform: rotate(30deg);  
  -ms-transform: rotate(30deg);  
  transform: rotate(30deg);  
}
```

Код Sass:

```
=transform($property)  
  -webkit-transform: $property  
  -ms-transform: $property  
  transform: $property  
  
.box  
  +transfom(rotate(30deg))
```

Препроцессор SASS

Возможности SASS:

- Расширение/наследование

Используя директиву `@extend` можно наследовать наборы свойств CSS от одного селектора другому.

Код CSS:

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

Код Sass:

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

Препроцессор SASS

Возможности SASS:

- Вычисления

Sass имеет несколько стандартных математических операторов, таких как +, -, *, / и %.