

CSS-препроцессоры

Переменные

- Философия этого препроцессора - повторное использование CSS синтаксиса там, где это возможно

синтаксис

@название_переменной: значение_переменной;

Объявляем, используя атрибут @

```
1  @background-color: #ffffff;
2  @text-color: #1A237E;
3
4  p{
5      background-color: @background-color;
6      color: @text-color;
7      padding: 15px;
8  }
9
10 ul{
11     background-color: @background-color;
12 }
13
14 li{
15     color: @text-color;
16 }
17
```

LESS

В КОМПИЛЯЦИИ

CSS

```
1  p {
2    background-color: #ffffff;
3    color: #1A237E;
4    padding: 15px;
5  }
6  ul {
7    background-color: #ffffff;
8  }
9  li {
10   color: #1A237E;
11 }
12
```

Операции с переменными (операторы)

Пример	Название	Результат	Числа	Строки
$-\text{@a}$	Отрицание	Смена знака @a	+	ошибка
$\text{@a} + \text{@b}$	Сложение	Сумма @a и @b	+	игнорируется *
$\text{@a} - \text{@b}$	Вычитание	Разность @a и @b	+	игнорируется *
$\text{@a} * \text{@b}$	Умножение	Произведени е @a и @b	+	игнорируется *
$\text{@a} / \text{@b}$	Деление	Частное от деления @a на @b	+	игнорируется *

Допускается производить операции с выражениями, значения которых соответствуют строкам. «*» — игнорируется при условии, что производится операция с явно указанной строкой в выражении не через переменную, а напрямую. Иначе ошибка.

пример

LESS

```
1  @the-border: 1px;
2  @base-color: #111111;
3  @red: #842210;
4
5  #header {
6    color: @base-color*3;
7    border-left: @the-border;
8    border-right: @the-border*2;
9  }
10
11 #footer {
12  color: @base-color+#333333;
13  }
14
```

КОМПИЛЯЦИЯ

```
1 #header {  
2   color: #333333;  
3   border-left: 1px;  
4   border-right: 2px;  
5 }  
6 #footer {  
7   color: #444444;  
8 }  
9
```

CSS

Арифметические операции (строгий режим)

Для включения строгого режима арифметических операций нужно взять выражения в круглые скобки (выражение).

В таком режиме запрещается производить операции со строками.

Пример	Название	Результат	Числа	Строки
$(-@a)$	Отрицание	Смена знака $@a$	+	ошибка
$(@a + @b)$	Сложение	Сумма $@a$ и $@b$	+	ошибка
$(@a - @b)$	Вычитание	Разность $@a$ и $@b$	+	ошибка
$(@a * @b)$	Умножение	Произведени е $@a$ и $@b$	+	ошибка
$(@a / @b)$	Деление	Частное от деления $@a$ на $@b$	+	ошибка


```
@margin: 10px;  
@padding: 10px;
```

```
body {  
  background: @white;  
  color: @black;  
  margin: @margin;  
  padding: @padding;  
}
```

```
body {  
  background: @white;  
  color: @black;  
  margin: @margin;  
  padding: (@padding + 10);  
}
```

Перечисленные в таблицах операции можно производить с числами, строками, цветами и числами с указанной размерностью

@a	@b	@a + @b	@a - @b	@a * @b	@a / @b
1	1	2	0	1	1
1px	2	3px	-1px	2px	0.5px
5%	4	9%	1%	20%	1.25%
2%	3px	5%	-1%	6%	0.6666..66 %
0.33	11%	11.33%	-10.67%	3.6300...03 %	0.0300...02 %
#6699cc	25%	#6b9ed1	#6194c7	#ffffff	#141f29
#666	#333	#999999	#333	#ffffff	#020202

примеси

- Примеси являются одним из способов подключить набор свойств из одного правила в другое правило.

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

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

Свойства из `.bordered` появятся в `#menu a` и в `.post a`. (Отметьте, что в качестве примеси можно использовать и `#id`)

ИЛИ

LESS

```
1 #circle{
2   background-color: #4CAF50;
3   border-radius: 100%;
4 }
5
6 #small-circle{
7   width: 50px;
8   height: 50px;
9   #circle
10 }
11
12 #big-circle{
13   width: 100px;
14   height: 100px;
15   #circle
16 }
```

```
1 #circle {
2   background-color: #4CAF50;
3   border-radius: 100%;
4 }
5 #small-circle {
6   width: 50px;
7   height: 50px;
8   background-color: #4CAF50;
9   border-radius: 100%;
10 }
11 #big-circle {
12   width: 100px;
13   height: 100px;
14   background-color: #4CAF50;
15   border-radius: 100%;
16 }
```

пример

LESS

```
1  @dark-color: #333;
2  @light-color: #f0f0f0;
3  @path-img: "../img";
4  @block: block;
5
6  .@{block} {
7    background: url("@{path-img}/pic.png") no-repeat;
8  }
9  .@{block}-title {
10   background:@dark-color;
11   color: @light-color;
12 }
13 .@{block}-content {
14   padding: 10px;
15 }
16
```

CSS

```
1  .block {
2    background: url("../img/pic.png") no-repeat;
3  }
4  .block-title {
5    background: #333;
6    color: #f0f0f0;
7  }
8  .block-content {
9    padding: 10px;
10 }
11
```

ВЛОЖЕННОСТЬ

```
nav { }  
nav li { }  
nav li a { }  
nav li a:hover { }  
nav li a.active { }  
nav li a.visited { }
```

```
nav {  
  li {  
    a {  
      &:hover { }  
      &:active { }  
      &:visited { }  
    }  
  }  
}
```

- Символ "&" используется перед псевдо-классами в нашей вложенной структуре.

CSS

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

less

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

CSS

```
.column1 {width: 100px;}  
.column2 {width: 120px;}  
.column3 {width: 130px;}
```

LESS

```
.column {  
  &1 {width: 100px;}  
  &2 {width: 120px;}  
  &3 {width: 130px;}  
}
```

LESS

```
ul{
  background-color: #03A9F4;
  padding: 10px;
  list-style: none;

  li{
    background-color: #fff;
    border-radius: 3px;
    margin: 10px 0;
  }
}
```

CSS

```
ul {  
  background-color: #03A9F4;  
  padding: 10px;  
  list-style: none;  
}  
ul li {  
  background-color: #fff;  
  border-radius: 3px;  
  margin: 10px 0;  
}
```

ФУНКЦИИ

- У `less` есть много функций, которые трансформируют цвета, работают со строками и числами.

```
@base: #f04615;
```

```
@width: 0.5;
```

LESS

```
.class {
```

```
  width: percentage(@width); // returns `50%`
```

```
  color: saturate(@base, 5%);
```

```
  background-color: spin(lighten(@base, 25%), 8);
```

```
}
```

ФУНКЦИИ

- С помощью LESS-функции **spin** можно повернуть цветовое колесо на определённый угол относительно заданного цвета и получить **новый цвет**.

spin(цвет, угол_поворота)

color: spin(red, 90); // цвет повернётся от красного на 90° по часовой

border-color: spin(#f0f, -45); // цвет на 45° от #f0f против часовой

LESS-функции lighten и darken

color: lighten(red, 50%); // светлее red на 50%

color: darken(blue, 25%); // темнее blue на 25%

Второе значение задаётся в процентах от 0% до 100%.

При задании **100%** в **lighten** функция возвращает **полностью белый цвет**,

а 100% для darken — полностью чёрный.

То есть эти функции «смешивают» заданный цвет с белым или чёрным.

Работа с цветом

```
@color: #3d82d1;
.left_box {
  background: lighten(@color, 20%);
}
.right_box {
  background: darken(@color, 20%);
}
```

LESS



LESS-функции для работы с цветом: **saturate** и **desaturate**

color: saturate(green, 20%); // green

насыщеннее на 20%

color: desaturate(blue, 50%); // blue бледнее

на 50%

```
base-color: rgb(110, 27, 255);  
alter-color: spin(@base-color, 60);
```

LESS

```
btn {  
  &-primary {  
    background-color: @base-color;  
  
    &-hover, &:hover {  
      background-color: lighten(@base-color, 20%);  
    }  
  
    &-pressed, &:active {  
      background-color: darken(@base-color, 20%);  
    }  
    /* &-правило добавьте тут */  
  }  
}
```

CSS

```
.btn-primary {  
  background-color: #6e1bff;  
  /* &-правило добавъте тук */  
}  
.btn-primary-hover,  
.btn-primary:hover {  
  background-color: #af81ff;  
}  
.btn-primary-pressed,  
.btn-primary:active {  
  background-color: #4200b4;  
}
```

Общие функции

- **color**
- **image-size, image-width, image-height**
- **Convert**
- **default**
- **unit**
- **get-unit**
- **svg-gradient**

Пространства имен и аксессуары

- Объединение нескольких миксинов и переменных под **#bundle**, для последующего повторного использования или распределения
- **#bundle {**
- **.button**
- **{ display: block;**
- **border: 1px solid black;**
- **background-color: grey;**
- **&:hover {**
- **background-color: white**
- **}**
- **}**
- **.tab { ... }**
- **.citation { ... }**
- **}**

Пространства имен и аксессуары

- применение миксин к `.button` классу в нашем `#header`

```
#header a {  
  color: orange;  
#bundle > .button;  
}
```

переменные, объявленные внутри пространства имен будут в области видимости только этого пространства имен и не будут доступны из вне пространства

Область видимости(Scope)

- Область видимости в Less очень похожа, на подобную область видимости в языках программирования.
- Переменные и миксины, вначале **ищутся локально**, и если они не найдены, компилятор будет осуществлять **поиск в родительском пространстве**, и так далее по иерархии

порядок определения переменных имеет значение

```
@var: red;
  #page {
    @var: white;
    #header {
      color: @var; // white
    }
  }
}
```

- В данном примере, если мы удалим определение переменной **@var** внутри **#page**, то свойство **color** у **#header** будет: **red**, поскольку переменная **@var** не будет найдена ни внутри **#header**, ни внутри **#page**, а только в корне всех правил, где она и содержит значение: **red**

Группа в ВК

- https://vk.com/simple_tech_university