WORKFLOW # 3.2

Sass, Compass, Sprite Sheets

LoftSchool
от мыслителя к создателю

SASS / SCSS

LoftSchool
от мыслителя к создателю

Дак SASS или SCSS ?

SASS (Syntactically Awesome Style Sheets) - это язык который при помощи компилятора написанного на Ruby, транислируется в CSS.

SCSS - css-подобный диалект SASS.

sass

```
.class
  color: red
  border: 1px solid blue
```

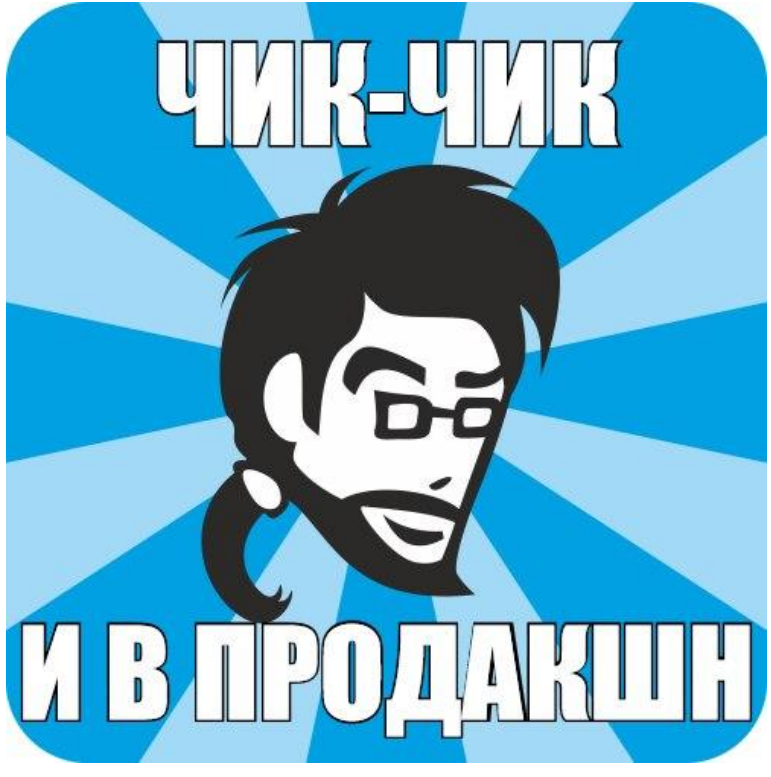
scss

```
.class {
  color: red;
  border: 1px solid blue;
}
```

LoftSchool
от мыслителя к создателю

Ускорение процесса разработки

- вложенности селекторов
- составные селекторы
- арифметические операции
- переменные
- функции
- и много прочих вкусностей...



но для начала нужно SASS установить

Так как **SASS** написан на **RUBY,** его нужно сначала установить

Установка RUBY:

- **MAC** - возрадуйтесь! Ruby у вас уже установлен!
- **Linux (Ubuntu)** — sudo apt-get update && apt-get upgrade && apt-get install ruby
- **Windows** — скачать исполняемый файл Ruby Installer (http://rubyinstaller.org/) и установить его

```
sudo gem install sass
```

Ruby Gems - пакетный менеджер на ruby.

LoftSchool
от мыслителя к создателю

Компилируем SCSS

```
sass path/to/scssFIle.scss dir/with/cssFile.css
```

```
sass --watch scssDir : cssFolder
```

Можно и через **gulp**, как обычно, но об этом немного позже...

LoftSchool
от мыслителя к создателю

Вложенности селекторов

Если написать дополнительный селектор внутри основных фигурных скобок, то это селектор становится **зависимым!**

scss

```scss
.class {
  color: red;

  .dependent-class {
    background: blue;
  }
}
```

css

```css
.class {
 color: red;
}

.class .dependent-class {
   background: blue;
}
```

LoftSchool
от мыслителя к создателю

Составные селекторы

Операток - **&** - конкатенирует строку написанную после него с первоначальным селектором.

scss

```scss
.class {
  font-size: 12px;

  &__inner{
    color: red;
  }
  &:hover{
    font-size: 16px;
  }
  & + .class {
    font-size: 20px;
  }
}
```

css

```css
.class {
 font-size: 12px;
}
.class__inner {
 color: red;
}
.class:hover {
 font-size: 16px;
}
.class + .class {
 font-size: 20px;
}
```

LoftSchool
от мыслителя к создателю

Арифметические операции

SASS умеет производить операции над числами, но если будут складываться **разные** единицы измерения то будет ошибка.

scss

```scss
.class {
  padding: 0 20px;
  width: 400px - (20 * 2);
}


// будет ошибка
.error {
  padding: 20px + 30em;
}
```

css

```css
.class {
 padding: 0 20px;
 width: 360px;
}
```

LoftSchool
от мыслителя к создателю

Переменные

Переменные в **SASS** могут хранить что угодно - строки, числа, цвета, списки (массивы) и т.д. Имя переменной может начинаться либо с латиницы либо с подчеркивания.

scss

```scss
$color: #fff;
$width: 200px;
.class {
  color:  $color;
  width: $width - 50;
}

$content-text: 'контент';
.class {
  &:after {
     content: $content-text;
  }
}
```

css

```css
.class {
 color: #fff;
 width: 150px;
}

.class:after {
 content: "контент";
}
```

LoftSchool
от мыслителя к создателю

Используем функции

SASS содержит в себе огромное количество встроенных функций (работы с цветами, с массивами, со строками и пр.)

scss

```scss
.class {
  background: lighten(#000, 10%);
}
```

css

```css
.class {
 background: #1a1a1a;
}
```

LoftSchool
от мыслителя к создателю

и создаем

Для создания функций есть специальные конструкции **@function** и **@return** для возвращения результата.

scss

```
@function funcName($param) {
  $value : $param;

  @return $value;
}
```

LoftSchool
от мыслителя к создателю

дебаггинг и консольлоггинг

Что бы вывести в консоль значение переменной или другую информацию, существует директива **@debug**

scss

```scss
$num: 2;

.class {
  $int : $num + 3;
  @debug $int; // 5
}
```

LoftSchool
от мыслителя к создателю

Повторное использование кода

- примеси
- циклы
- наследование
- “плейсхолдеры”
- условия

Примеси (mixins)

Примеси созданы для генерации свойств в зависимости от входных параметров

scss

```scss
@mixin transition ($value){
  -webkit-transition: $value;
  -moz-transition: $value;
  -ms-transition: $value;
  -o-transition: $value;
  transition: $value;
}
.class{
  opacity : .8;
  @include transition(opacity  .3s);
}
```

css

```css
.class {
 opacity: .8;
 -webkit-transition: opacity 0.3s;
 -moz-transition: opacity 0.3s;
 -ms-transition: opacity 0.3s;
 -o-transition: opacity 0.3s;
 transition: opacity 0.3s;
}
```

LoftSchool
от мыслителя к создателю

Бесконечное число параметров

Иногда в свойство CSS можно передавать несколько параметров что бы в примесях это не вызывало ошибок, нужно применить оператор (...)

scss

```scss
@mixin transition ($value...) {
  -webkit-transition: $value;
  -moz-transition: $value;
  -ms-transition: $value;
  -o-transition: $value;
  transition: $value;
}

.class {
  @include transition(opacity  .3s, color .3s);
}
```

LoftSchool
от мыслителя к создателю

Циклы

Перебирать массивы можно при помощи конструкции **@each in**

scss

```scss
$drinks: beer, rum, absinthe;

@each $drink in $drinks {
  .my__lovely-#{$drink} {
    background: url('/img/for-#{$drink}');
  }
};
```

css

```css
.my__lovely-beer {
 background: url('/img/for-beer');
}

.my__lovely-rum {
 background: url('/img/for-rum');
}

.my__lovely-absinthe {
 background: url('/img/for-absinthe');
}
```

LoftSchool
от мыслителя к создателю

Условия

Конечно же есть **@if, @else** и **@else if**

scss

```scss
$bg: backgrounded, colored;

@each $type in $bg {
  .colored-block__#{$type} {
    @if $type == colored {
      background: red;
    } @else {
      background: url('/img/pic.jpg');
    }
  }
};
```

css

```css
.colored-block__backgrounded {
  background: url("/img/pic.jpg");
}

.colored-block__colored {
  background: red;
}
```

LoftSchool
от мыслителя к создателю

Наследование

Если нам нужно расширить или унаследовать свойства класса, мы можем применить директиву **@extend**

scss

```scss
.class {
  color: red;
  width: 200px;
  height: 300px;
}

.same-class {
  @extend .class;
  margin-left: 200px;
}
```

css

```css
.class, .same-class {
  color: red;
  width: 200px;
  height: 300px;
}

.same-class {
  margin-left: 200px;
}
```

LoftSchool
от мыслителя к создателю

Плейсхолдеры

Что бы не дублировать функциональные классы, можно использовать специальную конструкцию

scss

```scss
%inlineblock {
  display: inline-block;
  vertical-align: top;
}

.one {
  @extend %inlineblock;
}

.two {
  @extend %inlineblock;
}
```

scss

```scss
.one, .two {
  display: inline-block;
  vertical-align: top;
}
```

LoftSchool
от мыслителя к создателю

less VS sass
FIGHT!

LoftSchool
от мыслителя к создателю

тернарный оператор

SCSS

```scss
$width: 200;

.input  {
        text-align: if ($width > 200, left, center);
}
```

LoftSchool
от мыслителя к создателю

синтаксис циклов

SCSS

```scss
$list: one, two, three;

@each $item in $list {
        .section__elem_#{$item} {
                // styles
        }
}
```

LESS

```less
@list: one, two, tree;
@list-length: length(@list);

.loop(@list-length) when (@list-length > 0) {
  .loop((@list-length - 1));
  @name: extract(@list, @list-length);

  .inner__class-@{name} {
    // styles
  }
}

div {
  .loop(@list-length);
}
```

LoftSchool
от мыслителя к создателю

продвинутая интерполяция

SCSS

```scss
$sides: top, right, bottom, left;

@each $side in $sides {
  .triangle__#{$side} {
    border: 10px solid transparent;
    border-#{$side}-color: 10px solid green;
  }
}
```

LoftSchool
от мыслителя к создателю

сложное формирование имен и свойств

SCSS

```scss
$width: 100;

@for $i from 1 through 5 {
        $newWidth: 100 * $i;

        .triangle__#{if($newWidth > 200, left, right)} {
            border-#{if($newWidth > 200, top, bottom)}-color : red;
        }
}
```

LoftSchool
от мыслителя к создателю

выход за пределы вложенности... и сознания)

SCSS

```scss
.class {
    .inner {
            color: red;

            @at-root {
                .outer-class {
                  border-radius: 30px;
                }
            }
    }
}
```

CSS

```css
.class .inner {
        color: red;
}
.outer-class {
        border-radius: 30px;
}
```

LoftSchool
от мыслителя к создателю

у SASS есть COMPASS

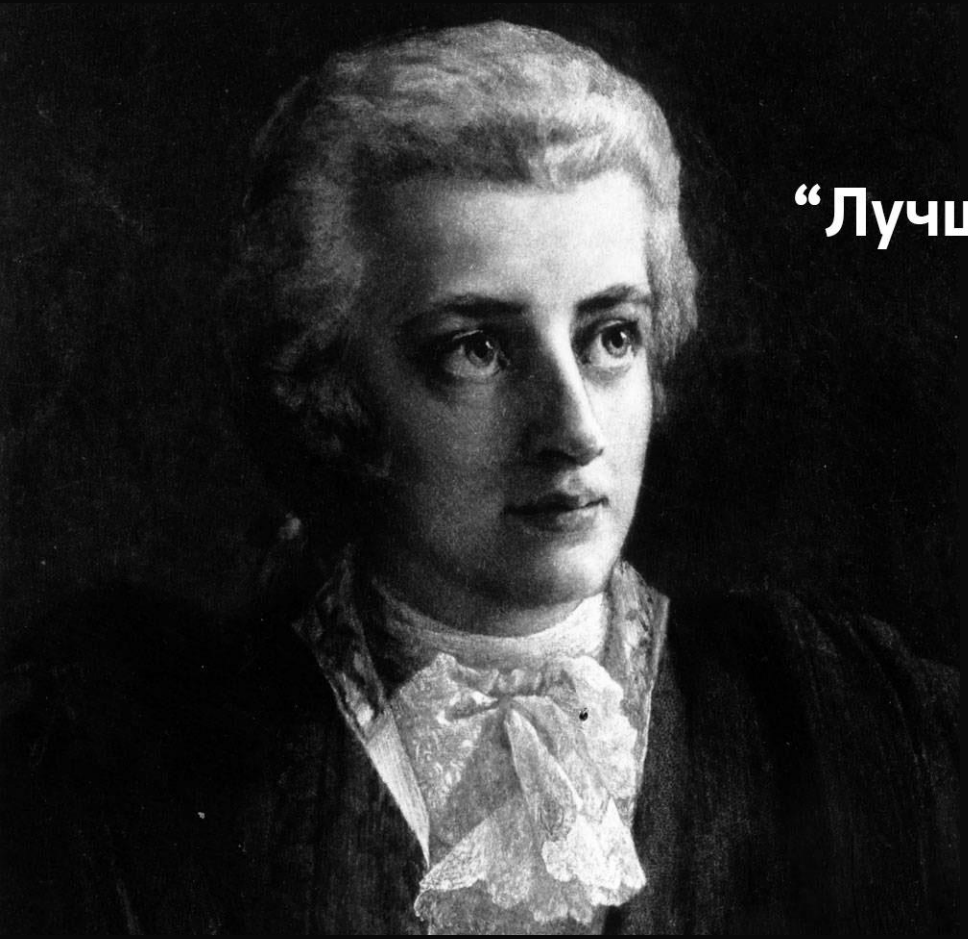
COMPASS

Раз уже у нас установлен ruby, можно ставить любые гемы какие захотим, почему бы не начать с Compass?

```
gem install compass
```

- Набор примесей, функций и дополнительных возможностей
- Имеет возможность расширения при помощи сторонних библиотек.
- Удобный инструмент для работы со спрайтами.

LoftSchool
от мыслителя к создателю



“Лучше один раз увидеть,
чем сто раз услышать”

- Вольфганг Амадей Моцарт