Preprocessors: LESS, SASS, Stylus

CSS-препроцессор конвертирует код, написанный на языке препроцессора, в css. Наиболее полулярные CSS – препроцессоры: Sass, LESS, и Stylus.

Препроцессоры делают код более гибким и легким в поддержке, уменьшают количество кода

Возможные проблемы с препроцессорами:

- все те, кто работают с сайтом, должны использовать одинаковый препроцессор. (также нельзя разрабатывать без препроцессора вовсе, если кто-то его использует)

***Syntax***

***Sass & LESS***

Sass использует файлы с расширением .scss

LESS использует расширение .less

Stylus использует расширение .styl

***Variables***

После объявления переменные могут использоваться во всем файле стилей

Sass переменные предшествуются символом $

|  |  |
| --- | --- |
|  | $mainColor: #0982c1;  body {    color: $mainColor;  } |

LESS переменные предшествуются символом @

|  |  |
| --- | --- |
|  | @mainColor: #0982c1;    body {    color: @mainColor;  } |

Stylus

Stylus переменные не предшествуются ничем, но могут предшествоваться символом $.

|  |  |
| --- | --- |
|  | mainColor = #0982c1  $borderStyle = dotted    body    color mainColor    border 1px $borderStyle mainColor |

***Nesting***

Если нужно ссылаться на элементы с одним и тем же родителем, можно это сделать с помощью Nesting, при такой записи символ & будет ссылаться на родительский элемент.

Все 3 препроцессорных языка имеют одинаковый синтаксис

Sass, LESS, & Stylus

|  |  |
| --- | --- |
|  | section {    margin: 10px;      nav {      height: 25px;        a {        color: #0982C1;          &amp;:hover {          text-decoration: underline;        }      }    }  } |

***Mixins***

Mixins – функции, которые позволяют переиспользовать свойства в файле стилей. В него можно передавать параметры и таким образом конфигурировать свойства разных объектов.

***Sass***

/\* Sass mixin error with (optional) argument $borderWidth which defaults to 2px if not specified \*/

@mixin error($borderWidth: 2px) {

  border: $borderWidth solid #F00;

}

.generic-error {

   @include error(); /\* Applies styles from mixin error \*/

}

.login-error {

@include error(5px); /\* Applies styles from mixin error with argument $borderWidth equal to 5px\*/

}

LESS

/\* LESS mixin error with (optional) argument @borderWidth which defaults to 2px if not specified \*/

.error(@borderWidth: 2px) {

  border: @borderWidth solid #F00;

}

.generic-error {

.error(); /\* Applies styles from mixin error \*/

}

.login-error {

  : 20px;

  .error(5px); /\* Applies styles from mixin error with argument @borderWidth equal to 5px \*/

}

Stylus

/\* Stylus mixin error with (optional) argument borderWidth which defaults to 2px if not specified \*/

error(borderWidth= 2px) {

  border: borderWidth solid #F00;

}

.generic-error {

  error(); /\* Applies styles from mixin error \*/

}

.login-error {

  error(5px); /\* Applies styles from mixin error with argument borderWidth equal to 5px \*/

}

***Inheritance***

Оптимизировано свойство описание свойств, которые селектор наследует от своего родительского Sass & Stylus

.block {

  padding: 2px;

}

p {

  @extend .block; /\* Inherit styles from '.block' \*/

  border: 1px solid #EEE;

}

ul, ol {

  @extend .block; /\* Inherit styles from '.block' \*/

  color: #333;

}

LESS не поддерживает подобный стиль наследования в чистом виде, вместо этого используются миксины без аргументов.

.block {

  margin: 10px 5px;

  padding: 2px;

}

p {

  .block; /\* Inherit styles from '.block' \*/

  border: 1px solid #EEE;

}

ul, ol {

  .block; /\* Inherit styles from '.block' \*/

  color: #333;

  text-transform: uppercase;

}

***Importing***

В отличии от обычного CSS в препроцессорах позволяется использовать import, потому что в таком случае он не требует отдельного HTTP запроса, а всего лишь подгружается во время добавится на фазе компиляции.

Импортировать можно также миксины и переменные.

Sass, LESS, & Stylus

|  |
| --- |
| /\* file.{type} \*/  body {    background: #EEE;  } |
| @import "reset.css";  @import "file.{type}";    p {    background: #0982C1;  } |

***Color Functions*** – встроенные функции, которые позволяют менять цвет на этапе компиляции. Используется для создания градиентов, помогают сделать цвет светлее/темнее

Sass

lighten($color, 10%); /\* returns a color 10% lighter than $color \*/

darken($color, 10%);  /\* returns a color 10% darker than $color \*/

saturate($color, 10%);   /\* returns a color 10% more saturated than $color \*/

desaturate($color, 10%); /\* returns a color 10% less saturated than $color \*/

grayscale($color);  /\* returns grayscale of $color \*/

complement($color); /\* returns complement color of $color \*/

invert($color);     /\* returns inverted color of $color \*/

mix($color1, $color2, 50%); /\* mix $color1 with $color2 with a weight of 50% \*/

LESS

lighten(@color, 10%); /\* returns a color 10% lighter than @color \*/

darken(@color, 10%);  /\* returns a color 10% darker than @color \*/

saturate(@color, 10%);   /\* returns a color 10% more saturated than @color \*/

desaturate(@color, 10%); /\* returns a color 10% less saturated than @color \*/

spin(@color, 10);  /\* returns a color with a 10 degree larger in hue than @color \*/

spin(@color, -10); /\* returns a color with a 10 degree smaller hue than @color \*/

mix(@color1, @color2); /\* return a mix of @color1 and @color2 \*/

Список существующих функций:  [LESS Documentation](http://lesscss.org/#-color-functions).

Stylus

lighten(color, 10%); /\* returns a color 10% lighter than 'color' \*/

darken(color, 10%);  /\* returns a color 10% darker than 'color' \*/

saturate(color, 10%);   /\* returns a color 10% more saturated than 'color' \*/

desaturate(color, 10%); /\* returns a color 10% less saturated than 'color' \*/

Список существующих функций  [Stylus Documentation](http://learnboost.github.com/stylus/docs/bifs.html).

***Operations***

Математические операции в стилях

Sass, LESS, & Stylus

body {

  margin: (14px/2);

  top: 50px + 100px;

  right: 100px - 50px;

  left: 10 \* 10;

}

***Практическое применение***

Vendor Prefixes – с помощью миксинов позволяют писать кроссбраузерные стили

Sass

@mixin border-radius($values) {

  -webkit-border-radius: $values;

     -moz-border-radius: $values;

          border-radius: $values;

}

div {

  @include border-radius(10px);

}

LESS

.border-radius(@values) {

  -webkit-border-radius: @values;

     -moz-border-radius: @values;

          border-radius: @values;

}

div {

  .border-radius(10px);

}

Stylus

border-radius(values) {

  -webkit-border-radius: values;

     -moz-border-radius: values;

          border-radius: values;

}

div {

  border-radius(10px);

}

Error Reporting – CSS препроцессоры репортят об ошибках

Links

<http://www.vanseodesign.com/css/css-preprocessors/>

<http://csspre.com/compare/>

http://code.tutsplus.com/tutorials/sass-vs-less-vs-stylus-a-preprocessor-shootout--net-24320