2. Метаязыки: SASS

SASS расшифровывается как Syntactically Awesome Style Sheets – если переводить дословно, то это звучит как: «Синтаксически потрясающие таблицы стилей».

Эта технология была придумана и воплощена Хэмптоном Катлином (Hampton Catlin). SASS манипулирует CSS-правилами, используя переменные, так называемые миксины (mixins), наследование и вложенность.

Исходные и скомпилированные файлы имеют расширения .sass и .scss, соответственно. Исходники переводятся в хорошо отформатированный CSS-код с помощью командной строки или веб-плагина.

SASS упрощает написание CSS-кода и позволяет динамически им манипулировать. Это отличный способ создания более функциональных CSS-кодов, который позволяет ускорить выполнение ежедневной работы веб-разработчиков и дизайнеров.

2.1 Особенности SASS

2.1.1 Вложенность

Одна из главных особенностей SASS – вложенность. Она облегчает написание написание стилей. К примеру данный код:

#container p {

font-family: Arial;

font-size: 13px;

}

#container h1 {

font-family: Tahoma;

font-size: 15px;

}

#container h2 {

font-family: Helvetica;

font-size: 14px;

}

Для SASS будет выглядеть иначе:

$myFontsize1: 13px;

$myFontsize2: 18px;

$myFontsize3: 25px;

$myWidth: 500px;

$myMargin: 0px auto;

#container {

width: $myWidth;

margin: $myMargin;

p {

font-family: Arial;

font-size: $myFontsize1;

}

h1 {

font-family: Tahoma;

font-size: $myFontsize3;

}

h2 {

font-family: Helvetica;

font-size: $myFontsize2;

}

}

В данном коде все стили элементов расположены под идентификатором container, вместо того, чтобы предварять этим id каждый из них.

2.1.2 Переменные

Переменные SASS объявляются с предваряющим их название символом $ и записываются аналогично CSS-свойствам. С помощью SASS, вы можете определять переменные для таких стилей, как font size, margin, padding и так далее.

Использование переменных дает вам возможность повторного использования заданных ранее значений.

В SASS существует шесть разных типов переменных: Строковые, Цветовые, Логические, Списковые, Тип null.

2.1.3. Миксины.

Миксины позволяют определять общие свойства, а затем использовать их повторно, то есть это реализация наследования. Миксины определяются с помощью директивы @mixin и включают в себя блок кода, который затем можно использовать с помощью директивы @include. Внизу приведен пример использования миксина:

@mixin border {

border: 1px solid red;

}

#container {

width: 960px;

margin: 0 auto;

@include border;

}

Директива @mixin была использована для создания стиля свойства border, затем эта же директива была включена в стиль идентификатора container с помощью директивы @include.