

ОСНОВЫ SASS

№ урока: 7 **Курс:** HTML5&CSS3 Advanced

Средства обучения: Visual Studio, Visual Studio Code, NotePad++

Обзор, цель и назначение урока

Цель урока – ознакомление с препроцессором SASS. Установка его в рабочее приложение, подключение и компиляция.

Изучив материал данного занятия, учащийся сможет:

- Устанавливать и подключать SASS.
- Пользоваться различными синтаксисами написания препроцессора.
- Использовать различные методы и свойства SASS.
- Понимать основные директивы препроцессора.

Содержание урока

1. Определение SASS.
2. Расширение файлов – SASS и SCSS.
3. Преобразование SASS формата в CSS.
4. Вложенность.
5. Использование переменных.
6. Импортирование и расширение файлов.
7. Использование функций (mixin).
8. Интерполяция переменных.
9. Работа с цветами.
10. Директивы в SASS.

Резюме

- SASS («Syntactically Awesome StyleSheets») — это препроцессор, представляющий собой промежуточное звено между таблицами стилей, и самими .css файлами, которые отправляются в браузер. Он заполняет пробелы в языке CSS, позволяя писать более быстрый, более продуктивный и более простой в поддержке код, полностью соблюдая принцип DRY (Don't repeat yourself). В 2007-м появилась первая версия SASS, разработанная Гемптоном Кетлином.
- SASS известен как синтаксис отступов и является расширением CSS3, а именно, его синтаксиса SCSS («Sassy CSS») – стильный CSS.
- Самый старый вариант написания SASS - это синтаксис отступов. Расширение файлов - *.sass.
- В SASS-синтаксисе нет фигурных скобок, точек с запятой, вложенность элементов в нем реализована при помощи отступов, а стилевые правила обязательно отделены новыми строками. Синтаксис отступов SASS требует очень четкого соблюдения отступов у вложенных свойств. Если в качестве отступов используются табуляция (Tab), компилятор выдаст ошибку при попытке конвертировать SASS, в котором помимо табуляции, также, используются пробелы в качестве отступов.
- Второй вариант – это синтаксис, расширяющий синтаксис CSS – Sassy CSS. SCSS пишется как обычный CSS, но расширен дополнительными возможностями SASS. Расширение файлов с SCSS синтаксисом - *.scss.
- Для взаимопонимания формата нужно использовать компилятор. Его задача — привести SASS в понятный классический CSS, который будет распознан любым браузером. Роль компилятора может выполнять серверный js или программа. Программы-компиляторы проверяют *.scss и *.sass-файлы на изменения и автоматически компилируют из них готовые стили в формат *.css.

- SASS позволяет вкладывать правила CSS друг в друга. Вложенные правила применяются только для элементов, соответствующих внешним селекторам. Это помогает избежать повторения родительских селекторов, и делает сложные CSS макеты с большим количеством вложенных селекторов гораздо проще.
- `lighten` – функция позволяет осветлить выбранный цвет (цвет значение, на которое нужно осветлить - в %). `darken` – затемнение.
- Препроцессор позволяет выполнять операции: сложение, вычитание, умножение, деление, деление по модулю.
- SASS позволяет назначать переменные — и это одно из ключевых преимуществ. Переменная, начинается со знака доллара (\$).
- `@import` – функция, позволяющая импортировать в основной sass файл – дополнительные *.sass, *.scss файлы, при этом все переменные будут работать в основном файле. В главный компилируемый файл – может быть загружено несколько фрагментов.
- `Mixin` - блоки SASS кода (или примеси-шаблоны), которые могут принимать аргументы и позволяют значительно расширить возможности написания стилей и сократить затраты времени на применении однотипных правил и даже целых CSS блоков. `Mixin` – подобие функции, которая может принять аргумент, выполнить огромный объем работы и выдать результат в зависимости от входного параметра. Миксин объявляется директивой `@mixin`, после объявления должно быть указано имя миксин. Вызывается - директивой `@include`, которая принимает имя миксина и передаваемые аргументы, если такие имеют место быть.
- Директива `@extend` позволяет сообщать Sass, что один селектор должен наследовать стили другого.
- Интерполяция — способ нахождения промежуточных значений по имеющемуся набору известных значений. Интерполяция в SASS используется для получения нового значения переменной, благодаря "интегрированию" в значение другой переменной, посредством конструкции `#{}.`
- `transparentize` – функция, задающая альфа канал выбранному элементу.
- Встроенная функция `if()` позволяет осуществлять выполнение, опираясь на условие, и возвращает один из возможных результатов. Она может быть использована в любом месте скрипта. Функция `if()` вычисляет только тот аргумент, который будет возвращён – это позволяет ссылаться на переменные, которые могут быть не определены или иметь вычисления, которые в других случаях привели бы к ошибке (например, деление на ноль).
- Директива `@for` выводит набор стилей заданное число раз. Для каждого повторения используется переменная-счётчик для изменения вывода.
- Директива `@each` позволяет пройти по списку значений, а не просто чисел, и вывести определенный стиль для выбранного элемента списка.
- Директива `@while` принимает выражение SASS и циклично выводит вложенные в неё стили, пока выражение вычисляется как `true`.

Закрепление материала

- Какие вы знаете препроцессоры для CSS?
- Какая основная задача SASS?
- В чем отличие SASS и SCSS?
- Как расшифровывается SCSS?
- Можно ли подключить на страницу HTML файлы *.sass и *.scss?
- Как можно расширить стили текущего элемента, стилями другого?
- Что позволяет сделать `mixin`?
- Какие математические операции можно выполнять в препроцессоре?
- Для чего используется директива `@for`?
- Что позволяет установить свойство `transparentize`?
- Как сделать затемнение цвета в SASS?

Дополнительное задание

Создайте страницу новостей, используя, header, nav, article, footer. Выполните оформление стилей основного блока контейнера – в файле main.scss. Стили для остальных блоков – подключите, используя @import. В стилях каждого блока страницы – должно быть минимум 2 переменные и цикл, а остальные свойства scss на Ваше усмотрение.

Самостоятельная деятельность учащегося

Задание 1

Выучить основные понятия, рассмотренные на уроке.

Задание 2

Создайте страницу, используя новые теги HTML5 – header, footer, article, aside, nav. Выполните оформление стилей основного блока контейнера – в файле main.scss. Стили для остальных блоков – подключите, используя @import. В стилях каждого блока страницы – должно быть:

- минимум 3 переменных;
- mixin для кроссбраузерной верстки браузеров;
- смешивание цветов.

Рекомендуемые ресурсы

<http://htmlbook.ru/>

<https://html.com/>

<https://www.html5rocks.com/>

<https://sass-scss.ru>