HTML5&CSS3 Advanced

Основы SASS

После урока обязательно

Повторите этот урок в видео формате на ITVDN.com

Проверьте как Вы усвоили данный материал на TestProvider.com

Основы SASS

Препроцессор SASS

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.

SASS

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.

Преимущества
1. Совместимость с различными версиями CSS.
2. Большое количество функций для работы с переменными.
3. Синтаксис – SASS или SCSS.

Форматы файлов

Есть 2 варианта написания синтаксиса SASS: SASS и SCSS.

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.

Браузер, не различает ни одного из них, поэтому для взаимопонимания 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.

Форматы файлов

SASS

```
$blue: lightblue
$margin: 16px
$fontSize: 14px

.content
        border: 1px solid $blue
        color: $blue

.border
        padding: $margin / 2
        margin: $margin / 2
        border-color: $blue
```

SCSS

```
$blue: lightblue;
$margin: 16px;
$fontSize: 14px;

.content {
        border: 1px solid $blue;
        color: $blue;
}
.border {
        padding: $margin / 2;
        margin: $margin / 2;
        border-color: $blue;
}
```

CSS

```
.content {
        border: 1px solid lightblue;
        color: #217882;
}

.border {
        padding: 8px;
        margin: 8px;
        border-color: lightblue;
}
```

Компиляторы

Для автоматической компиляции из *.sass или *.scss в готовые стили в формат *.css используется несколько вариантов:

1) Через расширения текстового редактора, например, для VS Code:

- Sass/Less/Stylus/Typescript/Javascript/Jade/Pug Compile Hero Pro wscats.eno
- Sass/Less/Typescript/Jade/Pug Compile-Superhero bananaacid.compile-superhero
- Или другое расширение

2) Через установку специальных программ для компиляции, например, Node.js и Gulp или Webpack.

Компиляторы

Node.js - представляет среду выполнения кода, которая позволяет писать производительный код с использованием языка JavaScript, а также писать транслировать вызовы на данном языке в машинный код.

Node.js позволяет применить систему управления пакетами (npm) – для получения пакетов из сети, а также публикации своих собственных.

Gulp — это система сборки и задач (таск-менеджер для автоматического выполнения часто используемых задач, например, минификации, тестирования, объединения файлов, отслеживания изменений), написанный на языке программирования JavaScript.

Webpack — это сборщик модулей. Он анализирует модули приложения, создает граф зависимостей, затем собирает модули в правильном порядке в один или более бандл (bundle), на который может ссылаться файл «index.html».

SassScript

SASS позволяет назначать переменные — и это одно из ключевых преимуществ. Переменная, начинается со знака доллара ($). Переменные в SASS можно разделить на 4 типа:

- число;
- строка;
- логический тип;
- цвета.

```
// SCSS

$color: #442213;

.class1 {
  color: $color;
          h2 {
          color: $color;
          }
}
```

```
/*Скомпилированный CSS */



.class1 {
  color: #442213;
}
.class1 h2 {
  color: #442213;
}
```

SassScript

SASS дает возможность использовать стандартные арифметические операции над числами, такие как сложение, вычитание, деление, умножение и деление по модулю (%).
Кроме того, в SASS есть возможность конкатенировать строки.

```
// SCSS

$size: 100;
$color: #221113;

.class1 {
  color: $color*2;
  width: $size+px
            h2 {
            width: $size/2+px;
            }
}
```

```
/*Скомпилированный CSS */

.class1 {
  color: #442226;
  width: 100px;
}
.class1 h2 {
   width: 50px;
}
```

SASSScript

Интерполяция — способ нахождения промежуточных значений по имеющемуся набору известных значений.
Чаще всего интерполяция в SASS используется для получения нового значения переменной, благодаря "интегрированию" в значение другой переменной, посредством конструкции #{}

```
// SCSS

$box: box;

.class1 {
  box-sizing: border-#{$box};
          h2 {
             width: content-#{$box};
             }
}
```

```
/*Скомпилированный CSS */



.class1 {
 box-sizing: border-box;}
.class1 h2 {
    box-sizing: content-box;}
```

SASS import

@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.

file.scss – фрагменты. Используется «» – тогда файлы не требуют расширения и компиляции.

@import "file" – в основном SCSS или SASS файле.

```
// SCSS основной

@import 'file';

.class2 {
        background:$color;
        width: $size;
}
```

```
// _file.scss - фрагмент

$size: 25px;
$color: green;

.class1 {
        font-size:$size*2;
}
```

```
/*Скомпилированный CSS */
.class1 {
        font-size: 50px;
}
.class2 {
        background: green;
        width: 25px;
}
```

SASS mixin

Правило DRY (Don't Repeat Yourself) реализовано в SASS при помощи техники mixin.

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.

SASS директивы

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.

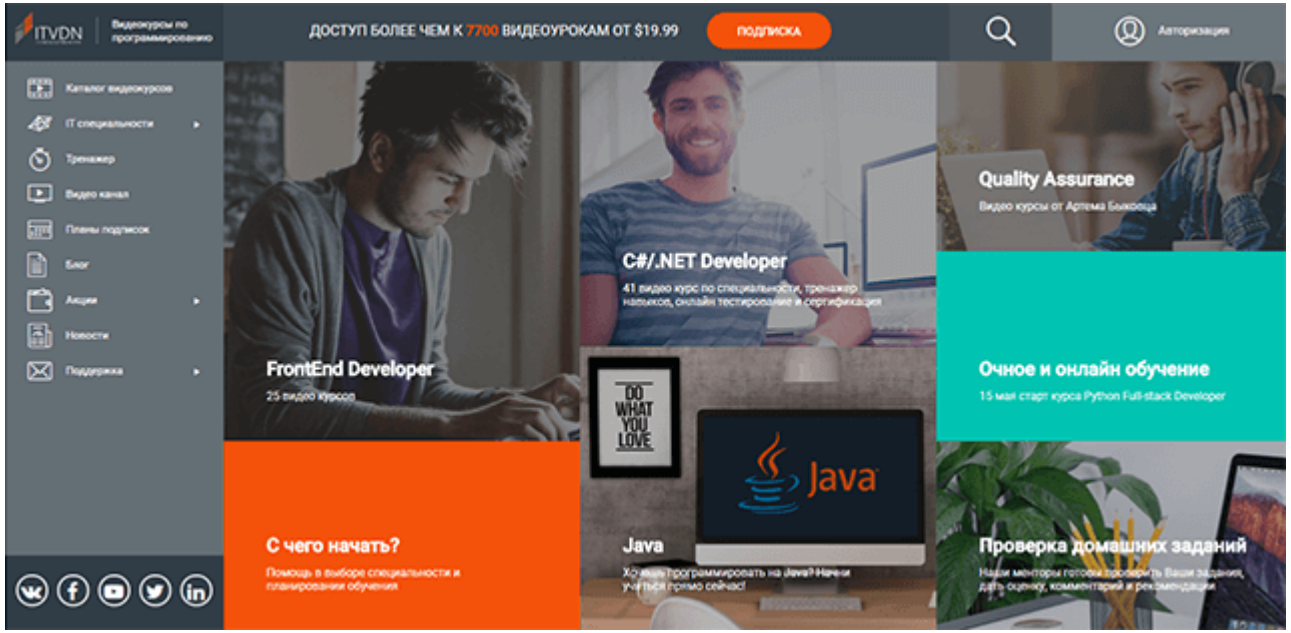
Директива @for выводит набор стилей заданное число раз. Для каждого повторения используется переменная-счётчик для изменения вывода.

Директива @each позволяет пройтись 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.

Смотрите наши уроки в видео формате

ITVDN.com



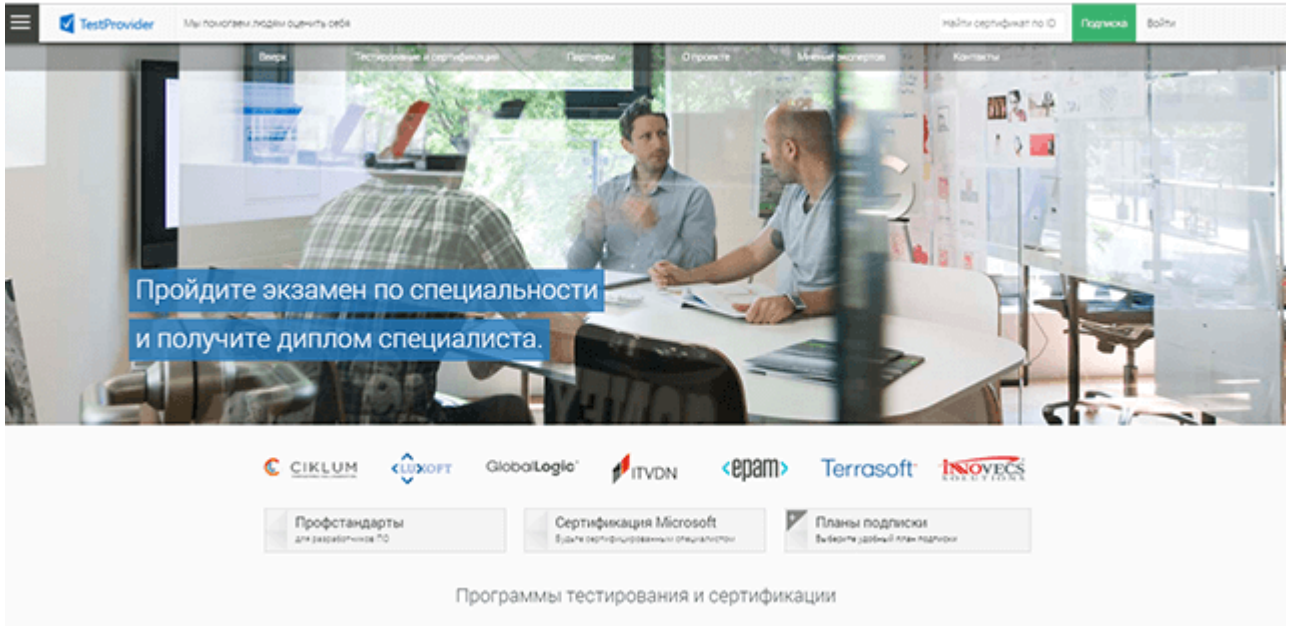
Посмотрите этот урок в видео формате на образовательном портале ITVDN.com для закрепления пройденного материала.

Курсы записаны сертифицированными тренерами, которые работают в учебном центре CyberBionic Systematics и другими высококвалифицированными разработчиками.

ITVDN
IT VIDEO DEVELOPERS NETWORK

Проверка знаний

TestProvider.com



TestProvider – это online сервис проверки знаний по информационным технологиям. С его помощью Вы можете оценить Ваш уровень и выявить слабые места. Он будет полезен как в процессе изучения технологии, так и для общей оценки знаний IT специалиста.

После каждого урока проходите тестирование для проверки знаний на TestProvider.com

Успешное прохождение финального тестирования позволит Вам получить соответствующий Сертификат.

TestProvider

Q&A

Информационный видеосервис для разработчиков программного обеспечения

ITVDN

IT VIDEO DEVELOPERS NETWORK

TestProvider PROMETRIC TEST CENTER CyberBionic systematics Microsoft Partner Silver Learning Windows Azure Cloud Partner EBA European Business Association