SASS, UN PREPROCESADOR CSS

ADOLFO SANZ DE DIEGO

@ASANZDIEGO

EL AUTOR

ADOLFO SANZ DE DIEGO

- Empec **desarrollando aplicaciones web**, hasta que di el salto a la docencia.
- Actualmente soy **Asesor Técnico Docente** en el servicio TIC de la D.G de Infraestructuras y Servicios de la Consejer a de Educaci n, Juventud y Deporte de la Comunidad de Madrid.
- Adem s colaboro como **formador especializado en tecnologías de desarrollo**.

ALGUNOS PROYECTOS

- **Hackathon Lovers** http //hackathonlovers.com un grupo creado para emprendedores y desarrolladores amantes de los hackathones.
- **Password Manager Generator** http //pasmangen.github.io un gestor de contrase as online.
- **MarkdownSlides** https //github.com/asanzdiego/markdownslides un script para crear slides a par r de ficheros MD.

¿DONDE ENCONTRARME?

- Mi nick **asanzdiego**
  - AboutMe http //about.me/asanzdiego
  - GitHub http //github.com/asanzdiego
  - Twitter http //twitter.com/asanzdiego
  - Blog http //asanzdiego.blogspot.com.es
  - LinkedIn http //www.linkedin.com/in/asanzdiego
  - Google+ http //plus.google.com/+AdolfoSanzDeDiego

INTRODUCCIÓN

¿QUÉ ES?

- Sass es un **pre-procesador de CSS**.
- A ade caracter s cas como **variables, mixins, funciones, etc.**

VENTAJAS

- El CSS es as **más fácil de mantener, personalizable y extensible**.
- Sass ene una **sintaxis parecida a CSS**.
- Es el preprocesador que usa Bootstrap 4.0.

INSTALACIÓN

INSTALAR RUBY

- Ubuntu

```
sudo apt-get install ruby-full
```

- Windows http //rubyinstaller.org/
- Mac incluido

INSTALAR SASS

- Despu s de instalar Ruby, ejecutar

```
sudo gem install sass
```

PREPROCESAR

- Un fichero

```
sass --watch input.scss:output.css
```

- Un directorio

```
sass --watch input/dir:output/dir
```

PROBAR SIN INSTALAR

- http //www.sassmeister.com/

CARACTERÍSTICAS

VARIABLES (I)

- El siguiente c digo

```
$font-stack:    Helvetica, sans-serif;
$primary-color: #333;

body {
  font: 100% $font-stack;
  color: $primary-color;
}
```

VARIABLES (II)

- Se compila a

```
body {
  font: 100% Helvetica, sans-serif;
  color: #333;
}
```

VALORES POR DEFECTO (I)

- El siguiente c digo

```
$content: "First content";
$content: "Second content?" !default;
$new_content: "First time reference" !default;

#main {
  content: $content;
  new-content: $new_content;
}
```

VALORES POR DEFECTO (I)

- Se compila a

```
#main {
  content: "First content";
  new-content: "First time reference";
}
```

SCOPE

- Los **ámbitos de las variables** en Sass es muy similar a otros lenguajes

```
$var: red;

#page {
  $var: white;
  #header {
    color: $var; // white
  }
}
```

REGLAS ANIDADAS (I)

- El siguiente c digo

```
nav {
  ul {
    margin: 0;
    padding: 0;
    list-style: none;
  }

  li { display: inline-block; }
}
```

REGLAS ANIDADAS (II)

- Se compila a

```
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}

nav li {
  display: inline-block;
}
```

PARCIALES

- Es un archivo con un gui n bajo

```
_partial.scss
```

- Con esto Sass en ende que es un archivo parcial y que no debe generar CSS.

IMPORTS (I)

- Imaginemos el archivo **_reset.scss**

```
html, body, ul {
  margin: 0;
  padding: 0;
}
```

IMPORTS (II)

- Y el archivo **base.scss**

```
@import 'reset';

body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

IMPORTS (III)

- Se compila a

```
html, body, ul, ol {
  margin: 0;
  padding: 0;
}

body {
  font: 100% Helvetica, sans-serif;
  background-color: #efefef;
}
```

PARENT (I)

- El c digo ‘&’ hace referencia al padre

```
a {
  font-weight: bold;
  text-decoration: none;
  &:hover { text-decoration: underline; }
  body.firefox & { font-weight: normal; }
}
```

PARENT (II)

- Se compila a

```
a {
  font-weight: bold;
  text-decoration: none;
}
a:hover { text-decoration: underline; }
body.firefox a { font-weight: normal; }
```

MIXINS (I)

- El siguiente c digo

```
@mixin border-radius($radius) {
  -webkit-border-radius: $radius;
     -moz-border-radius: $radius;
      -ms-border-radius: $radius;
          border-radius: $radius;
}

.box {
  @include border-radius(10px);
}
```

MIXINS (II)

- Se compila a

```
.box {
  -webkit-border-radius: 10px;
  -moz-border-radius: 10px;
  -ms-border-radius: 10px;
  border-radius: 10px;
}
```

ARGUMENTOS VARIABLES (I)

- El siguiente c digo

```
@mixin box-shadow($shadows...) {
  -moz-box-shadow: $shadows;
  -webkit-box-shadow: $shadows;
  box-shadow: $shadows;
}

.shadows {
  @include box-shadow(0px 4px 5px #666, 2px 6px 10px #999);
}
```

ARGUMENTOS VARIABLES (II)

- Se compila a

```
.shadows {
  -moz-box-shadow: 0px 4px 5px #666, 2px 6px 10px #999;
  -webkit-box-shadow: 0px 4px 5px #666, 2px 6px 10px #999;
  box-shadow: 0px 4px 5px #666, 2px 6px 10px #999;
}
```

PASAR CONTENIDO A MIXIN (I)

- El siguiente c digo

```
@mixin apply-to-ie6-only {
  * html {
    @content;
  }
}
@include apply-to-ie6-only {
  #logo {
    background-image: url(/logo.gif);
  }
}
```

PASAR CONTENIDO A MIXIN (II)

- Se compila a

```
* html #logo {
  background-image: url(/logo.gif);
}
```

INTERPOLACIÓN (I)

- Podemos sus tuir el palor de una variable en nombre de propiedades o selectores con #{$variable}

```
$position: left;
@mixin car($brand, $color) {
  .coche.#{$brand} {
    border-#{$position}: 2px;
    background-color: $color;
    background-image: url('img/#{$brand}-#{$color}.png');
  }
}
@include car('audi', 'green');
```

INTERPOLACIÓN (II)

- Se compila a

```
.coche.audi {
  border-left: 2px;
  background-color: green;
  background-image: url('img/audi-green.png');
}
```

EXTEND (I)

- El siguiente c digo

```
.message {
  color: #333;
}
.success {
  @extend .message;
  border-color: green;
}
.error {
  @extend .message;
  border-color: red;
}
```

EXTEND (II)

- Se compila a

```
.message, .success, .error, .warning {
  color: #333;
}
.success {
  border-color: green;
}
.error {
  border-color: red;
}
```

OPERADORES (I)

- El siguiente c digo

```
article[role="main"] {
  float: left;
  width: 600px / 960px * 100%;
}

aside[role="complementary"] {
  float: right;
  width: 300px / 960px * 100%;
}
```

OPERADORES (II)

- Se compila a

```
article[role="main"] {
  float: left;
  width: 62.5%;
}

aside[role="complementary"] {
  float: right;
  width: 31.25%;
}
```

FUNCIONES (I)

- Sass dispone de una variedad de **funciones matemáticas, que manipulan cadenas, y que transforman los colores**

```
$base: #f04615;
$list: 200, 500, 1200;

.class {
  width: nth($list, 3);
  color: darken($base, 5%);
  background-color:
    lighten($base, 25%);
}
```

FUNCIONES (II)

- Se compila a

```
.class {
  width: 1200;
  color: #f6430f;
  background-color: #f8a58d;
}
```

DEFINIR FUNCIONES (I)

- Podemos definir nuestras propias funciones

```
$grid-width: 40px;
$gutter-width: 10px;

@function grid-width($n) {
  @return $n * $grid-width + ($n - 1) * $gutter-width;
}

#sidebar { width: grid-width(5); }
```

DEFINIR FUNCIONES (II)

- Se compila a

```
#sidebar {
  width: 240px;
}
```

IF (I)

- El siguiente c digo

```
$animal: cat;
p {
  @if 1+1 == 2 { border: 2px solid black; }
  @if $animal == cat { color: white; }
}
```

IF (I)

- Se compila a

```
p {
  border: 2px solid black;
  color: white;
}
```

FOR (I)

- El siguiente c digo

```
@for $i from 1 to 4 {
  .item-#{$i} { width: 2em * $i; }
}
```

FOR (II)

- Se compila a

```
.item-1 { width: 2em; }
.item-2 { width: 4em; }
.item-3 { width: 6em; }
```

EACH (I)

- El siguiente c digo

```
@each $animal in puma, tiger, salamander {
  .#{$animal}-icon {
    background-image: url('/images/#{$animal}.png');
  }
}
```

EACH (II)

- Se compila a

```
.puma-icon {
  background-image: url('/images/puma.png'); }
.tiger-icon {
  background-image: url('/images/tiger.png'); }
.salamander-icon {
  background-image: url('/images/salamander.png'); }
```

WHILE (I)

- El siguiente c digo

```
$i: 6;
@while $i > 0 {
  .item-#{$i} { width: 2em * $i; }
  $i: $i - 2;
}
```

WHILE (II)

- Se compila a

```
.item-6 { width: 12em; }
.item-4 { width: 8em; }
.item-2 { width: 4em; }
```

LOGS

- El siguiente c digo

```
@debug 10em + 12em;
```

- Saca por pantalla a la hora de compilar

```
Line 1 DEBUG: 22em
```

ACERCA DE

LICENCIA

- Estas **transparencias** est n hechas con
  - MarkdownSlides https //github.com/asanzdiego/markdownslides
- Estas **transparencias** est n bajo una licencia Crea ve Commons Reconocimiento-Compar rIgual 3.0
  - http //crea vecommons.org/licenses/by-sa/3.0/es

FUENTES

- Transparencias
  - https //github.com/asanzdiego/curso-interfaces-web-2020/tree/master/04-sass/slides
- Ejercicios
  - https //github.com/asanzdiego/curso-interfaces-web-2020/tree/master/04-sass/src

BIBLIOGRAFÍA

- Documentaci n oficial de Sass
  - http //sass-lang.com/
- Para probar Sass
  - http //www.sassmeister.com/