SASS/SCSS

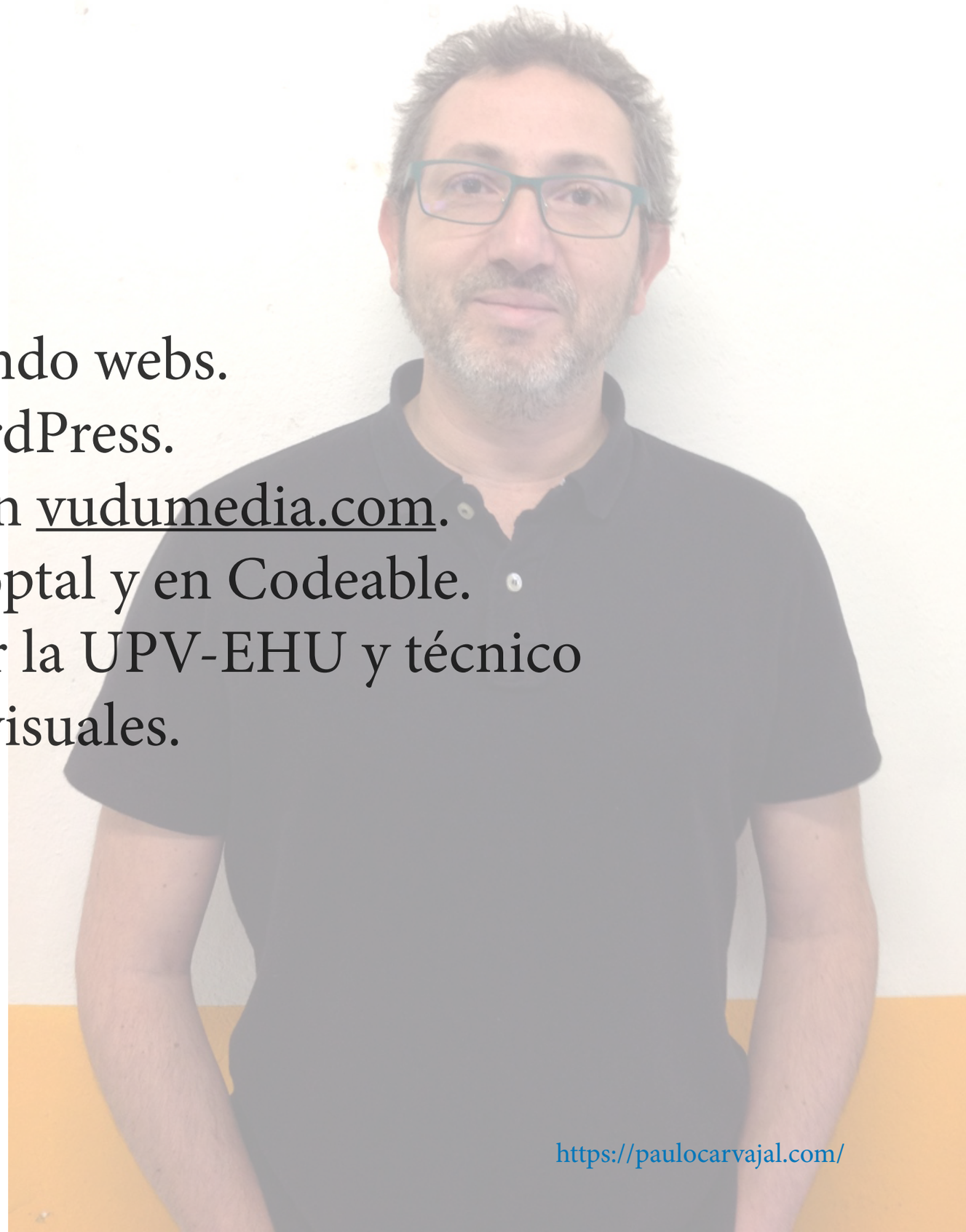
Qué es y cómo usar algunas funciones avanzadas

https://paulocarvajal.com/

SASS/SCSS

Hola, soy Paulo Carvajal

- 20+ años de experiencia creando webs.
- 12+ años trabajando con WordPress.
- Soy encargado del área web en vudumedia.com.
- Desarrollador freelance en Toptal y en Codeable.
- Licenciado en Bellas artes por la UPV-EHU y técnico especialista en Medios audiovisuales.



https://paulocarvajal.com/

¿Qué es SASS/SCSS?

https://paulocarvajal.com/

¿Qué es SASS/SCSS?

Un preprocesador de CSS

- La principal ventaja de SASS es la posibilidad de convertir CSS en algo **dinámico**.
- Permite trabajar más rápido con la posibilidad de crear **funciones** y reutilizar el código gracias a los **mixins**.
- Hay dos formatos diferentes para la sintaxis, lo que hace se traduce en dos extensiones de fichero diferentes: **.sass** y **.scss**
- SASS no puede ser interpretado por los navegadores, por lo se debe compilar para convertir nuestro archivo SASS en un clásico fichero CSS.

¿Como funciona SASS?

https://paulocarvajal.com/

¿Como funciona SASS?

Las reglas de estilo son la base de SASS, igual que para CSS. Y funcionan de la misma manera: se eligen los elementos a los que dar estilos con un selector y se declaran las propiedades que afectan la apariencia de esos elementos.

```
$color01: #b4d553;

.alert {
  color: $color01
  &:hover {
    font-weight: bold;
  }

}
```

https://paulocarvajal.com/

Funcionalidades principales

https://paulocarvajal.com/

Funcionalidades principales

- Variables.
- Anidamiento.
- Parciales e importación.
- Mixins.
- Extensión o herencia.

Comentarios

https://paulocarvajal.com/

Comentarios

SASS permite tres tipos de comentarios, los normales de CSS, se mantienen en la compilación si no es en “compress mode”:

```
/* Algo que recordar */
```

Los más comunes en otros lenguajes:

```
// Algo más
```

Y unos especiales que se mantienen siempre:

```
/*! Otra cosa más */
```

Variables

https://paulocarvajal.com/

Variables

Se asigna un valor a un nombre que comienza con $, y luego puede referirse a ese nombre en lugar de al valor.

```scss
$color01: #b4d553;

$mi-padding: 10px 20px 10px 30px;

$colores: #00FF00, #008000, #00FFFF, #008080;

$foto: 'https://mi-dominio.com/imagen.jpg';
```

Se pueden usar números, cadenas y listas. No hace falta que vayan entrecomilladas, pero es recomendable.

Variables

Tratan - y _ indistintamente.
Tienen “scope”.

```scss
$color: tomato;

body {
  $color: lightgreen;
  background: $color; // lightgreen;
}
```

Las variables de Sass son compiladas por Sass. Las variables CSS se incluyen en la salida CSS y pueden cambiar de valor.

Variables

En algunos casos se necesita usar el nombre de una variable como valor, en estos casos se usa esta notación para interpolar su valor:

`#{$name}`

```
$name: lacosa;
span.emoji-#{$name} {
  color: red;
}

span.emoji-lacosa{color:red}
```

La interpolación se puede usar casi en cualquier parte para incrustar el resultado de una expresión.

Anidamiento

https://paulocarvajal.com/

Anidamiento

En lugar de repetir los mismos selectores una y otra vez, pueden escribirse reglas de un estilo dentro de otras. SASS combina automáticamente el selector de la regla externa con la regla interna.

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

  li { display: inline-block; }

  a {
    display: block;
    padding: 6px 12px;
    text-decoration: none;
  }
}
```

Anidamiento

Lo que producirá el siguiente CSS:

```css
nav ul {
  margin: 0;
  padding: 0;
  list-style: none;
}
nav li {
  display: inline-block;
}
nav a {
  display: block;
  padding: 6px 12px;
  text-decoration: none;
}
```

Anidamiento

El selector **&**, es un selector especial que se usa en los selectores anidados para referirse al selector externo (padre). Permite reutilizar el selector externo en formas más complejas, como agregar una pseudo-clase o agregar un selector antes que el padre:

```scss
.alert {
  font-weight: normal;
  &:hover {
    font-weight: bold;
  }
  :not(&) {
    opacity: 0.8;
  }
}
```

```css
.alert {
  font-weight: normal;
}

.alert:hover {
  font-weight: bold;
}

:not(.alert) {
  opacity: 0.8;
}
```

Anidamiento

También permite añadir sufijos, lo cual es muy util si se usa metodología BEM:

```scss
.accordion {
    padding: 30px;
    &__open {
        padding: 20px;
    }
}
```

```css
.accordion {
  padding: 30px;
}

.accordion__open {
  padding: 20px;
}
```

Anidamiento

Y **&** también puede usarse como sufijo, lo cual es muy util para clases en el body, o para usar clases en JavaScript:

```
.alert {
  font-weight: normal;
  .bold & {
    font-weight: bold;
  }
}
```

```
.alert {
  font-weight: normal;
}

.bold .alert {
  font-weight: bold;
}
```

Parciales e importación

https://paulocarvajal.com/

Parciales e importación

SASS nos permite “partir” nuestras hojas de estilo en varios ficheros, lo que facilita la organización, para lo que se usa la función **@import**:

```
@import "variables";
@import "normalize";
@import "mixins";
@import "base";
@import "interface";

@import "modules/backgrounds";
@import "modules/media";
@import "modules/tables";
@import "modules/navigation";
```

Los ficheros parciales deben comenzar por **_nombre.scss**, si no los compila por separado.

Mixins

https://paulocarvajal.com/

Mixins

Un **mixin** permite aprovechar un trozo de nuestro código al que podemos llamar repetidamente y que encapsula una o más líneas de código.

Lo que en otros lenguajes se conoce como **función**.

Para crearlos se usa la directiva **@mixin** seguida de un nombre.

Para invocarlos se usa el término **@include** seguido del nombre.

Mixins

Los **mixins** pueden tener parámetros o no, y si los tienen, se les puede asignar un valor por defecto opcional. Y también pueden anidarse, llamar a uno desde otro.

```scss
@mixin circle($width: 50px) {
  width: $width;
  height: $width;
  border-radius: 100%;
  background:black;
}

// Uso:
div{
  @include circle(100px);
}
```

Mixins

La directiva **@content** nos permite pasar “contenido” a un mixin.

```scss
@mixin media($width) {
  @media only screen and (max-width: $width) {
    @content;
  }
}

@include media(320px) {
  background: red;
}

// Genera:
@media only screen and (max-width: 320px) {
  background: red;
}
```

Extensión o herencia

https://paulocarvajal.com/

Extensión o herencia

SASS nos permite aprovechar una clase ya existente y variarla o añadirle propiedades. Para ello se usa la directiva **@extend**.

```
.animal {
    background: gray;
}
.gato {
    @extend .animal;
    color: white;
}
```

```
.animal, .gato {
  background: gray;
}

.gato {
  color: white;
}
```

En resumen…

- Variables.
- Anidamiento.
- Parciales e importación.
- Mixins.
- Extensión o herencia.

https://paulocarvajal.com/

Funcionalidades avanzadas

https://paulocarvajal.com/

Funcionalidades avanzadas

- Control de flujo: if/else, each, for, while.
- Color: alpha, opacity, darken, lighten, desaturate, mix …
- Lists y maps.
- Debug: @error, @warn, @debug.

Y más…

https://paulocarvajal.com/

Enlaces

- https://sass-lang.com/
- https://sass-lang.com/documentation
- http://thesassway.com/
- https://github.com/Famolus/awesome-sass