

## Práctica sesión 3

---

Para realizar esta partiremos de la práctica anterior, y usaremos Compass.App. Lo primero que debemos de hacer será configurar nuestro proyecto, asociando los directorios de trabajo de css, sass e imágenes. Además, es importante que tengamos la línea de `relative_assets = true` **descomentada**.

### Archivo config.rb

```
css_dir = "css"
sass_dir = "sass"
images_dir = "img"
relative_assets = true
```

### Modificaciones establecidas en style.scss

**Eliminad la línea : @import "compass/reset"**

La semana anterior nos habíamos creado cuatro colores (#eee, #fff, #666, #888), pero en esta sesión nos vamos a quedar con dos ya que los otros los crearemos mediante funciones. De los colores anteriores nos quedaremos con:

- El color para el texto: #eee, y
- El color de fondo: #15a (antes #666).

**Nos vamos a crear tres colores nuevos como resultado de las funciones lighten y darken:**

- **Color del texto al pasar sobre él (hover):** será el resultado de aclarar (lighten) un 10% el color del texto (color #eee que tenemos definida de la práctica 1).
  - **Color del fondo al pasar sobre él:** será el resultado de aclarar el color de fondo un 10% (color #15a)
  - **Color del título,** será el resultado de oscurecer el color del fondo un 10% (color #15a)
- De modo que

Práctica sesión 2	Práctica sesión 3 (actual)
<code>\$colorText: #eee;</code> <code>\$colorTextHover : #fff;</code> <code>\$colorBack: #666;</code> <code>\$colorBackHover: #888;</code>	<code>\$colorText: #eee;</code> <code>\$colorTextHover : // DEBEMOS CALCULARLO</code> <code>\$colorBack: #15a;</code> <code>\$colorBackHover: // DEBEMOS CALCULARLO</code> <code>\$colorTitulo: // DEBEMOS CALCULARLO</code>

Además en esta sesión los mixin los guardaremos en el archivo: `_mixin.scss`, de modo que sólo vamos a tener en `style.scss`: los colores, el `@keyframes`, `body`, `header` y `footer`. (OJO: En esta práctica la position del footer lo estableceremos como `fixed` en lugar de como `absolute`) por tanto **debemos modificarlo!** para que el footer siempre esté fijo al final de la página.

Al tener los mixin en un fichero separado deberemos incluirlos en nuestro fichero style.scss al igual que el menu (incluido en la práctica anterior) y el archivo contenido que explicaremos más adelante.

### Archivo \_mixin.css

En este archivo vamos a copiar todos los mixin que creamos en la práctica anterior, además crearemos uno nuevo destinado al diseño responsive. Este nuevo mixin recibirá un parámetro de entrada que será: \$size, y dependiendo de:

- **size == medium:** comprobará si el ancho máximo son 1024px
- **size == small:** comprobará si el ancho máximo son 800px

### Archivo contenido.scss

1. Vamos a crearnos dos placeholder (%):

a. %button-block con:

- ✓ box-sizing con border-box;
- ✓ width con 100%;
- ✓ padding con 5px 10px;
- ✓ border-radius con 5px;
- ✓ border con 1px solid transparent;
- ✓ cursor con pointer;

b. %input-block con:

- ✓ box-sizing con border-box;
- ✓ width con 100%;
- ✓ border-radius con 5px;
- ✓ padding con 5px 10px;
- ✓ border con 1px solid #bbb;

2. Importamos compass: (@import "compass";)

3. Nos creamos una variable \$icons indicándole el directorio de los iconos y que coja **todos** los png, tal y como se hizo en la documentación aportada.

4. Y los recorremos estableciéndolos en el fondo y en la posición dada, tal y como viene en la documentación aportada.

5. El siguiente paso será la modificación del div cuyo identificador es “contenido”. Este div tendrá:

- a. Un **padding** de 30px 20px;
- b. La clase **col-left** del div deberá tener:
  - ✓ box-sizing con border-box;
  - ✓ width con 30%;
  - ✓ float con left;
  - ✓ padding con 10px;
  - ✓ los input del login extenderán del input-block
  - ✓ Los botones del login extenderán del button-block
  - ✓ Y esta clase además tendrá un diseño responsive para tamaños small, donde se le establecerá las propiedades de: float none y width: 100%
- c. La clase **col-right** del div:
  - ✓ box-sizing con border-box;
  - ✓ margin-left con un 30%;
  - ✓ width con un 70%;
  - ✓ padding de 10px;
  - ✓ Para los h1 de este div o color con \$colorTitulo; o font-size con 1.6em;
  - ✓ Además, para los tamaños responsive size == small le daremos un margin-left: 0; y width: 100%;
- d. La clase **button-action** del div tendrá:
  - ✓ background-color: será el color de fondo establecido en el screen.scss
  - ✓ color: blanco
  - ✓ Al pasar sobre ese botón, le daremos de background-color: el valor del fondo establecido en el screen.scss pero aclarado un 20%
- e. Estableceremos un diseño responsive al div para size == medium de forma que el padding sea 10px 20px

### **Modificaciones en menu.scss**

Llamamos al mixin para poner display:none cuando el tamaño sea medium al menú (<nav>)

**NOTA:** Al tener una resolución inferior a 1024px el menú desaparece de nuestra pantalla, esto será algo que solucionaremos en la siguiente práctica.

**¡IMPORTANTE!** Probar con `@import "compass"` o con `@import "compass/utilities/sprites";`

El bucle que te genera las imágenes es

```
@each $i in sprite-names($icons) {  
  .icon-#{$i} {  
    background-position: sprite-position($icons, $i);  
    @include sprite-dimensions($icons, $i);  
  }  
}
```