Parte 4 de 4: Boostrap

PERSONALIZAR BOOTSTRAP CON SASS

Apuntes de: **Rosa María Medina Gómez**

Adaptados a Power Point por José Jesús Torregrosa García

Curso de Formación del Profesorado a distancia
Cefire Específico de FP de Cheste

Generalitat Valenciana Curso 2018 - 2019

Bootstrap

INTRODUCCIÓN

PERSONALIZAR BOOTSTRAP CON SASS

Sass

- Estructura
- Variables por defecto
- Mapas y bucles
- Funciones
- Contrastes de colores
- Opciones con Sass
- Colores
- Colores en los temas
- Grises
- Modificadores
- Responsive

Bibliografía

PERSONALIZAR BOOTSTRAP CON SASS

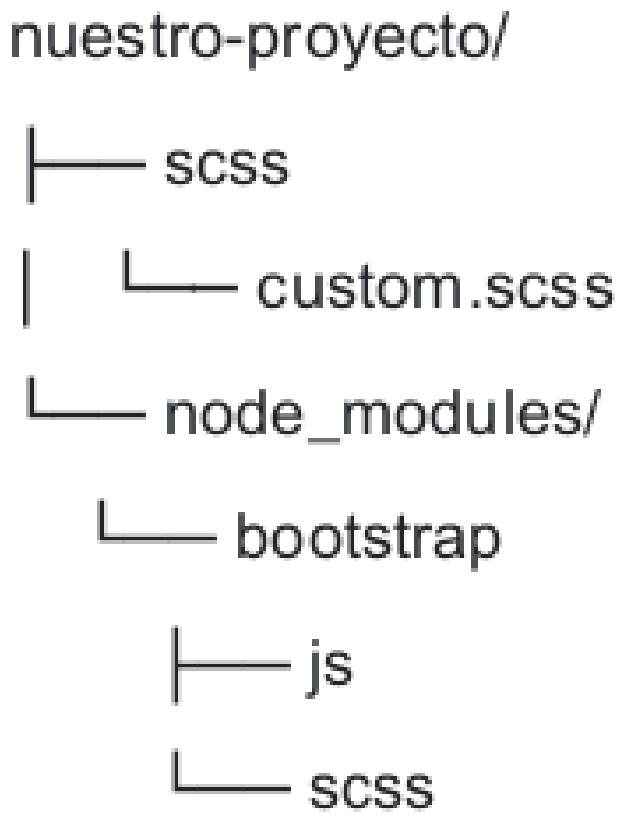
- Bootstrap 4 se ha proporcionado un enfoque familiar y ligeramente diferente a versiones anteriores
- Personalizar Sass con Bootstrap se logra mediante variables en Sass, mapas y CSS personalizados. Ahora, no encontramos hojas de estilo personalizadas de forma que podemos habilitar mucho más fácil los temas incorporados, añadir sombras, degradados, y mucho más.

USO DE SASS Y ESTRUCTURA

- Boostrap 4, nos permite utilizar los archivos de Sass para aventajarnos y modificar sus variables, mapas, mixin, etc. (utilizad la última versión, es decir: bootstrap-4.1.2). Estos archivos se encuentran en el directorio **bootstrap-4.1.2/scss**
- **Siempre que sea posible, debemos evitar modificar los archivos troncales de Bootstrap,** eso significa que debemos crear nuestras propias hojas de estilos, de modo que importemos Bootstrap, modifiquemos y extendamos los estilos ahí. Asumiendo que usamos la librería de Bootstrap que hemos descargado, la estructura que este tiene es:

USO DE SASS Y ESTRUCTURA

- Esta es la estructura en el proyecto de **Bootstrap**:



USO DE SASS Y ESTRUCTURA

- En nuestro **custom.scss**, debemos importar nuestros archivos de Bootstrap.
- Para ello tenemos dos opciones:
  - **Opción 1:** Incluir todo Bootstrap
  - **Opción 2:** Coger sólo las partes que necesitamos teniendo en cuenta las dependencias y requerimientos que pueden tener. Además, deberemos incluir algún JS para nuestros plugins.

USO DE SASS Y ESTRUCTURA

Opción 1: Incluir todo Bootstrap

```
/ Custom.scss
/ Opción A: Incluimos todo bootstrap

@import "node_modules/bootstrap/scss/bootstrap";
```

USO DE SASS Y ESTRUCTURA

Opción 2: Coger sólo las partes que necesitamos

```
/ Custom.scss
/ Opción B: Incluimos sólo aquellas partes que necesitamos

/ Requerido
@import "node_modules/bootstrap/scss/functions";
@import "node_modules/bootstrap/scss/variables";
@import "node_modules/bootstrap/scss/mixins";

// Opcional
@import "node_modules/bootstrap/scss/reboot";
@import "node_modules/bootstrap/scss/type";
@import "node_modules/bootstrap/scss/images";
@import "node_modules/bootstrap/scss/code";
@import "node_modules/bootstrap/scss/grid";
```

USO DE SASS Y ESTRUCTURA

- Con esta configuración, podemos modificar cualquier variable con Sass y mapear nuestro custom.scss.
- **¡IMPORTANTE!**: Si lo compiláis desde consola con una versión de Sass v3.5.2+ No tendréis problema (sudo gem update && sudo gem update sass), si lo hacéis con la app y con una versión anterior de Sass os dará un error en la línea 4 de bootstrap, comentarlo y el error desaparecerá.
- **¡IMPORTANTE!**: Los import deben ir debajo de nuestro código (el que modifica las variables).

VARIABLES POR DEFECTO

- Todas y cada una de las variables de Bootstrap incluyen la posibilidad de utilizar !default para poder sobrescribir cualquier valor en tu código de Sass sin necesidad de tener que modificar el código original de Bootstrap.
- Podemos por tanto copiar y pegar todas las variables que necesitemos y modificar así posteriormente sus valores quitando el valor de !default.
- Si una variable ha sido ya asignada, a esta variable no se va a reasignar por los valores que tenga por defecto en Bootstrap.

VARIABLES POR DEFECTO

- Las anulaciones de variables dentro del mismo archivo de Sass pueden estar antes o después de las variables predeterminadas, sin embargo, si se anula en todos los archivos de Sass estas anulaciones deben realizarse antes de importar los archivos Sass de Bootstrap.
- Veamos un ejemplo de cómo modificar el color de fondo del body al importar Bootstrap

```
//Your variable overrides $body-bg: #000;
$body-color: #111;
/ |
// Bootstrap and its default variables
@import "node_modules/bootstrap/scss/bootstrap";
```

MAPAS Y BUCLES

- Bootstrap v4 incluye mapas en Sass y pares de clave-valor para facilitar la generación de nuestros CSS.
- Usaremos por tanto mapas de Sass para crear nuestros colores, puntos de cortes en una cuadrícula y más cosas. Al igual que las variables de Sass, todos los mapas incluyen el valor **!default** que se puede anular y/o ampliar.

MAPAS Y BUCLES

- Por ejemplo, para modificar un color existente en nuestro mapa $theme-colors, podemos añadir lo siguiente a nuestro archivo customizado de Sass:

```
$theme-colors: (
  "primary": #0074d9,
  "danger": #ff4136
);
```

- Si por el contrario lo que queremos es añadir un nuevo color, le daremos un nuevo **atributo-valor**:

```
$theme-colors: (
  "mi-color": #900
);
```

FUNCIONES

- Bootstrap utiliza numerosas funciones en Sass, pero sólo un subconjunto de ellas son aplicables en los temas. Por tanto, se han incluido tres funciones para obtener los valores de los mapas de colores en bootstrap:

```
@function color($key: "blue") {
        @return map-get($colors, $key);
}

@function theme-color($key: "primary") {
        @return map-get($theme-colors,$key);
}
@function gray($key: "100") {
        @return map-get($grays, $key);
}
```

FUNCIONES

- Y de esta forma podríamos crearnos un color personalizado:

```
.custom-element {
    color: gray("100");
    background-color: theme-color("dark");
}
```

FUNCIONES

- O si queremos especificar el nivel que queremos de un determinado color podríamos utilizar la función proporcionada

```
@function theme-color-level($color-name: "primary", $level: 0) {
    $color: theme-color($color-name);
    $color-base: if($level > 0, #000, #fff);
    $level: abs($level);
    @return mix($color-base, $color, $level * $theme-color-interval);
}
```

- En la práctica lo que haríamos sería llamar a la función y pasar los dos parámetros: el nombre del color ($theme-colors) y el valor numérico del nivel:

```
.mi-elemento {
    color: theme-color-level(primary, -10);
}
```

FUNCIONES: CONTRASTES DE COLORES

La función de abajo utiliza YIQ color spacee para automáticamente devolver un contasate de color claro #fff o un color oscuro #111 basado en un color especifico. Esta función es útil en mixin donde tenemos multiples clases.

```
@each $color, $value in $theme-colors {
 .swatch-#{$color} {
  color: color-yiq($value);

 }
}
```

FUNCIONES: CONTRASTES DE COLORES

```
.mi-elemento {
    color: color-yiq(#000); // returns `color: #fff`
}
```

O incluso, podemos especificar una base de color con nuestro mapa de colores.

```
.mi-elemento {
color: color-yiq(theme-color("dark")); // returns `color: #fff`
}
```

OPCIONES CON SASS

- Personalizar Bootstrap v4 con Sass es bastante fácil con el archivo de variables personalizadas incorporado, de forma que sólo tendríamos que cambiar las preferencias globales de CSS con las nuevas variables $enable-* en el archivo _variables.sccs con esto anulamos el valor de una variable, compilaríamos (npm run test).
- En las diapositivas siguientes vemos las posibilidades de personalización de variables

OPCIONES CON SASS

Variable	Valor	Descripción
$spacer	1rem (por defecto), o cualquier valor >0	Especifica el espacio por defecto que generará nuestro spacer utilities.
$enable-rounded	true (por defecto) o false	Por defecto en varios componentes está habilitado el border-radius
$enable-shadows	true o false (por defecto)	Por defecto en varios componentes está habilitado el box-shadow
$enable-gradients	true o false (por defecto)	Habilita los gradientes vía background-image en varios compoentes
$enable-transitions	true (por defecto) o false	Habilita las transiciones en varios componentes

OPCIONES CON SASS

$enable-hover-media-query	true o false (por defecto)	...
$enable-grid-classes	true (por defecto) o false	Habilita la generación de las clases CSS para el grid (Por ejemplo: .container, .row, .col-md-1,...)
$enable-caret	true (por defecto) o false	Habilita el pseudo elemento caret en la clase .dropdown-toggle
$enable-print-styles	true (por defecto) o false	Habilita estilos para optimizar la impresión

COLORES

Muchos de los componentes construidos en Bootstrapt tienen una serie de colores definidas en el mapa de Sass. Este mapa puede ser modificado para generar nuestro propio conjunto de reglas y por tanto de colores. Los colores se encuentran definidos en el archivo scss/_variables.scss.

Este archivo se expandirá en las siguientes versiones con el objetivo de añadir tonos adicionales similares a la paleta de escala de grises que sacaron.

COLORES

La forma de usar estos colores en nuestro Sass es:

```
// With variable

.alpha { color: $purple; }

// From the Sass map with our `color()` function

.beta { color: color("purple"); }
```

Estas utilidades podemos utilizarlas tanto con color como con background-color

COLORES EN LOS TEMAS

Al igual que en el punto anterior, los colores se encuentran situados en el archivo scss/_variables.scss, el cual podremos utilizar para modificar el color del tema.

GRISES

Tenemos una amplia gama de grises incluidas en el mapa de colores que se encuentra en el archivo scss/_variables.scss para jugar con las sombras de grises. Veamos un ejemplo de los colores en el mapa de Sass:

```
$colors: (
  "blue": $blue,
  "indigo": $indigo,
  "purple": $purple,
  "pink": $pink,
  "red": $red,
  "orange": $orange,
  "yellow": $yellow,
  "green": $green,
  "teal": $teal,
  "cyan": $cyan,
  "white": $white,
  "gray": $gray-600,
  "gray-dark": $gray-800
) !default;
```

GRISES

- Al añadir, eliminar o modificar los valores en el mapa se nos actualizara si se utilizan en otros componentes.
- **Actualmente** desafortunadamente en este momento no todos los componentes utilizan este mapa de colores en Sass.
- **Para las futuras versiones sí** que estará actualizado (o eso pretenden). Hasta entonces tendremos que usar la variable ${color} y este mapa de Sass.

MODIFICADORES

- Muchos **componentes**(por ejemplo clases bases .btn → ver archivo **_buttons.btn**) de Bootstrap han sido construidos con el objetivo de poder ser modificados posteriormente mediante unas clases modificadoras(por ejemplo la clases modificadora .btn-danger).
- Estas clases se crean a partir del mapa $theme-color para personalizar el nombre y número de clases modificadoras

```
//Generate alert modifier classes
//@each $color, $value in $theme-colors {
  .alert-#{$color} {
    @include alert-variant(theme-color-level($color, -10), theme-color-level($color,
-9), theme-color-level($color, 6));
  }
}


//Generate `.bg-*` color utilities
@each $color, $value in $theme-colors {
  @include bg-variant('.bg-#{$color}', $value);
}
```

MODIFICADORES

- Recorre el mapa $theme-color para generar modificadores para el componente .alert y todas nuestras utilidades de fondo: .bg-*

```
//Generate alert modifier classes
//@each $color, $value in $theme-colors {
  .alert-#{$color} {
    @include alert-variant(theme-color-level($color, -10), theme-color-level($color,
-9), theme-color-level($color, 6));
  }
}


//Generate `.bg-*` color utilities
@each $color, $value in $theme-colors {
  @include bg-variant('.bg-#{$color}', $value);
}
```

RESPONSIVE

- Antes en la diapositiva anterior hemos visto dos ejemplos donde se recorre el mapa $theme-color para generar modificadores para el componente .alert y todas nuestras utilidades de fondo: .bg-*
- Podemos modificar los responsive por ejemplo de nuestros componentes o utilities. Por ejemplo, si queremos modificar la alineación del texto donde tenemos un mix **@each** para cada **$grid-breakpoints** en Sass podríamos hacer:

```
@each $breakpoint in map-keys($grid-breakpoints) {

    @include media-breakpoint-up($breakpoint) {

        $infix: breakpoint-infix($breakpoint, $grid-breakpoints);
        .text#{$infix}-left   { text-align: left !important; }
        .text#{$infix}-right  { text-align: right !important; }
        .text#{$infix}-center { text-align: center !important; }
    }
}
```

BIBLIOGRAFÍA

- Página oficial: https://getbootstrap.com/docs/4.1/getting-started/theming/