

UD4. Activity 3

GRAPHIC USER INTERFACE DESIGN PRINCIPLES

Toni Barceló García

IES JOAN RAMIS I RAMIS | DAW

Tabla de contenido

Tabla de contenido	1
UD4. Activity 3	2
Source Github project	3
Source project:	3
Root folder for this activity: UD4. Activity 2	3
github-pages (latest version)	3
The website structure	3
Anotations:	4
First Part	5
Sass & Less	5
Variables	5
Nested structure	7
Reusing properties	9
Operations	10
Bootstrap Sass	11
Colors	11
Fonts	12
Buttons & inputs	13
Modal	13
Carousel	14

UD4. Activity 3

This activity has two parts. First, we will continue with the project we have started creating a new page to display an article. In this case, we are going to create the css file using Less and Sass, so we will practice with both of them using the basic concepts of these two preprocessors:

- Creating variables.
- Applying styles using a nested structure.
- Reusing properties to other selectors.
- Doing operations to calculate some measures (width, height, font-size...).

You have to create two different files with .less and .scss extensions and add the <link> tag on the head of the HTML document. You can comment one of the two files included, so I can check later manually.

You need to add a link to your home to access this page, as you did with the register page.

The second part of this activity is about using Sass with Bootstrap. You need to configure your project in order to use Node.js.

The idea is to change the theme colours of Bootstrap (primary, secondary, success, info, warning, danger, light, dark, primary-light), the font family used, and maybe some other options. After that change, instead of applying the colour using a custom class or editing our stylesheet, you will apply Bootstrap classes to do so.

Source Github project

If you clone the project, you will have all the activities I have done so far sorted by folders in "activities" root directory.

Source project:

<https://github.com/abarcelogarcia/abarcelogarcia.github.io>

Root folder for this activity: UD4. Activity 2

[activities/UD4A3_SASS_LESS](#)

github-pages (latest version)

<https://abarcelogarcia.github.io/>

The website structure

Directory	Files	Concept
root	index.html	main website file.
css	avatar_effect.css	css file to apply avatar's effect without js
	bootstrap_custom.css	css file generated by sass
	bootstrap_custom.css. map	Css map file generated by sass
	bootstrap_custom.scss	sass file code
	estils_less.css	css file generated by less
	estils_less.less	less file code
	estils_sass.css	css file generated by sass
	estils_sass.css.maps	css map file generated by sass
	estils_sass.scss	sass file code
fonts	BAHNSCHRIFT.TTF	typography chosen in the guide style.
img	*.png, *.jpg	images directory for the web.
node_modules	*	Bootstrap sass modules
views	post.html	Post file

Anotations:

In post.html there are linked both css styles files; sass and less. You have to comment/uncomment to test diferents files.

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.2/dist/css/bootstrap.min.css" rel="stylesheet"
    integrity="sha384-T3c6CoIi6uLrA9TneNEoa7RxnatzjcDSCmG1MXxSR1GAsXEV/Dwvykc2MPK8M2HN" crossorigin="anonymous" />

  <!-- Sass -->
  <link rel="stylesheet" href="../css/estils_sass.css">

  <!-- Less -->
  <!-- <link rel="stylesheet" href="../css/estils_less.css"> -->

  <title>Post</title>
</head>
```

First Part

Sass & Less

Sass code file is estisl_sass.scss

Less code file is estils_less.less

I have added two fonts and the avatar effects css from the previous exercise to have clean css code for the post site.

```
@font-face {
  font-family: 'Bahnschrift';
  src: url(../fonts/BAHNSCHRIFT.TTF);
}

@import url(https://fonts.googleapis.com/css?family=Roboto);
@import url("avatar_effect.css");
```

Variables

I have created the color variables from my style guide and the two types of fonts.

Sass

Less

```
// Variables

/* COLORS */

$blanco-corp: #f3f5f7;
$azul-corp: #001B2E;
$rojo-corp: #CC0000;
$amarillo-corp: #FF9900;
$verde-corp: #294C60;
$gris-corp: #ADB6C4;

/* FONTS */
$titulo: 'Bahnschrift';
$texto: 'Roboto';
```

```
// Variables

/* COLORS */

@blanco-corp: #f3f5f7;
@azul-corp: #001B2E;
@rojo-corp: #CC0000;
@amarillo-corp: #FF9900;
@verde-corp: #294C60;
@gris-corp: #ADB6C4;

/* FONTS */
@titulo: 'Bahnschrift';
@texto: 'Roboto';
```

I have used the variables to set the attributes of classes

estils_sass.scss

```
.navbar-bg {
  background-color: $gris-corp;
  color: $azul-corp;
}

.bg-page {
  background-color: $blanco-corp;
}

.heading {
  background-color: $azul-corp;
  font-family: $titulo;
  font-size: $titulo_size;
  color: $blanco-corp;
}

.topic,
.topic a {
  color: $azul-corp;
  text-decoration-line: none;
}
```

estils_less.less

```
.navbar-bg {
  background-color: @gris-corp;
  color: @azul-corp;
}

.bg-page {
  background-color: @blanco-corp;
}

.heading {
  background-color: @azul-corp;
  font-family: @titulo;
  font-size: @titulo_size;
  color: @blanco-corp;
}

.topic,
.topic a {
  color: @azul-corp;
  text-decoration-line: none;
}
```

estils_sass.css & estils_less.css

```
.navbar-bg {
  background-color: #ADB6C4;
  color: #001B2E;
}

.bg-page {
  background-color: #f3f5f7;
}

.heading {
  background-color: #001B2E;
  font-family: "Bahnschrift";
  font-size: 28px;
  color: #f3f5f7;
}

.topic,
.topic a {
  color: #001B2E;
  text-decoration-line: none;
}
```

Nested structure

The post is divided into 3 divs. Two of them with *principal* class and the third with *moreinfo* class.


```
<article>
  <div class="container p-4">
    <h1 class="display-1">Sport</h1>
    
  </div>
  <div class="container principal">...
</div>
  <div class="container moreinfo">...
</div>
  <div class="container principal">...
</div>
</article>
```

Each of them consists of a title header and a paragraph.

```
<div class="container principal">
  <div class="row">
    <div class="col">
      <h2>title</h2>
    </div>
    <div class="col-1">
      
    </div>
  </div>
  <div class="row">
    <p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Quia bla
      enim nam eos perferendis delectus cupiditate incidunt, aperiam asp
      perspiciatis, ducimus possimus in ea eos. Lorem ipsum dolor si
```



With this scenario, I created different nested styles according to their tags

estils_sass.scss



```
.principal {
  h2{
    font-family: $titulo;
    background-color: $azul-corp;
    color: $blanco-corp;
    padding-left: 2px;
  }
  p{
    font-family: $texto;
    text-align: justify;
    font-size: calc(#{ $titulo_size}/2)
  }
}
```

estils_less.less



```
.principal {
  h2{
    font-family: @titulo;
    background-color: @azul-corp;
    color: @blanco-corp;
    padding-left: 2px;
  }
  p{
    font-family: @texto;
    text-align: justify;
    font-size: calc(@titulo_size/2)
  }
}
```

estils_sass.css & estils_less.css

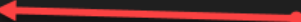
```
.principal h2 {
  font-family: 'Bahnschrift';
  background-color: #001B2E;
  color: #f3f5f7;
  padding-left: 2px;
}
.principal p {
  font-family: 'Roboto';
  text-align: justify;
  font-size: calc(28px/2);
}
```

Reusing properties

To reuse properties from other selectors I have added the style of *principal* class and added the background color attribute as a distinction of the header.

estils_sass.scss

```
.moreinfo{
  @extend .principal;
  h2{
    background-color: $gris-corp;
  }
}
```



estils_less.less

```
.moreinfo{
  .principal;
  h2{
    background-color: @gris-corp;
  }
}
```



estils_sass.css & estils_less.css

```
.moreinfo p {
  font-family: 'Roboto';
  text-align: justify;
  font-size: calc(28px/2);
}
.moreinfo h2 {
  background-color: #ADB6C4;
}
```

Operations

I assigned 28px to the font size of titles, and then assigned half the font size of the paragraphs based on the size of the title.

estils_sass.scss

```
/* FONTS */
$titulo: 'Bahnschrift';
$texto: 'Roboto';
$titulo_size: 28px;
```

```
.principal {
  h2{
    font-family: $titulo;
    background-color: $azul-corp;
    color: $blanco-corp;
    padding-left: 2px;
  }
  p{
    font-family: $texto;
    text-align: justify;
    font-size: calc(#{ $titulo_size }/2)
  }
}
```

estils_less.less

```
/* FONTS */
@titulo: 'Bahnschrift';
@texto: 'Roboto';
@titulo_size: 28px;
```

```
.principal {
  h2{
    font-family: @titulo;
    background-color: @azul-corp;
    color: @blanco-corp;
    padding-left: 2px;
  }
  p{
    font-family: @texto;
    text-align: justify;
    font-size: calc(@titulo_size/2)
  }
}
```

estils_sass.css & estils_less.css

```
.principal p {
  font-family: 'Roboto';
  text-align: justify;
  font-size: calc(28px/2);
}
```

Bootstrap Sass

Colors

I have added the colors from web style guide to variables and then assigned them to bootstrap themes and use their classes but with the web's own colors.

```
// Variables

/* COLORS */

$blanco-corp: #f3f5f7;
$azul-corp: #001B2E;
$rojo-corp: #CC0000;
$amarillo-corp: #FF9900;
$verde-corp: #294C60;
$gris-corp: #ADB6C4;

$theme-colors: (
  "primary": $azul-corp,
  "danger": $rojo-corp,
  "warning": $amarillo-corp
);
```

```
<div class="col-md-8">
  <div class="card-body">
    <h5 class="card-title fs-2">Title</h5>
    <p class="card-text fs-3">
      Lorem ipsum dolor sit amet consectetur
      adipisicing elit. Perferendis provident
      consectetur accusamus enim magni eveniet et
      deserunt unde quam, iste atque porro neque
      possimus necessitatibus eum blanditiis nulla
      doloribus. Nam.
    </p>
    
    <div class="text-end">
      <button class="btn btn-warning">Read</button>
    </div>
  </div>
</div>
```

Sport



Title

Lorem ipsum dolor sit amet consectetur adipisicing elit. Perferendis provident consectetur accusamus enim magni eveniet et deserunt unde quam, iste atque porro neque possimus necessitatibus eum blanditiis nulla doloribus. Nam.



Read

```
<!-- Section Article of the week -->
<section id="article_week">
  <div class="heading mb-2 fs-1">
    <div class="row align-items-center">
      <div class="col">
        <div class="ms-2 bg-primary">Article of the week</div>
      </div>
      <div class="col-auto">
        
      </div>
    </div>
  </div>
</section>
```

Article of the week



Title

Lorem ipsum dolor sit amet consectetur adipisicing elit.

Fonts

In addition to assigning my fonts to variables, I have modified font and sizes of Bootstrap's *display* text class.

I also changed the size of Bootstrap's *fs* class in the 3 sizes that the website uses.

```
/* FONTS */
$titulo: 'Bahnschrift';
$texto: 'Roboto';

$display-font-sizes: ( //Default
  1: 4rem,      //5rem
  2: 3.5rem,    //4.5rem
  3: 3rem,      //4rem
  4: 2.5rem,    //3.5rem
  5: 2rem,      //3rem
  6: 1.5rem     //2.5rem
);

// DISPLAY FONT (h1 class="display-x")
$display-font-family: 'Bahnschrift';
$display-font-style: null;
$display-font-weight: 400;

// FONT SIZE (class="fs-x")
$font-sizes: ( //default
  1: 28px,      //$h1-font-size
  2: 22px,      //$h2-font-size
  3: 14px       //$h3-font-size
);
```

```
<div class="container">
  <!-- mobile 1 column / big desktop 2 columns -->
  <div class="row row-cols-sm-1 row-cols-xl-2">
    <div id="carouselSport1" class="carousel slide">
      <h1 class="topic display-3">Sport</h1>
      <div class="carousel-inner">
        <div class="carousel-item active">
          <article>
            <div class="col">
              <div class="card bg-page mx-5 border-0">
                <div class="row g-2">
                  <div class="col-md-4 text-center my-2">
                    
                  </div>
                  <div class="col-md-8">
                    <div class="card-body">
                      <h5 class="card-title fs-2">title</h5>
                      <p class="card-text fs-3">
                        Lorem ipsum dolor sit amet consectetur
                        adipisicing elit. Perferendis provident
                        consectetur accusamus enim magni eveniet et
                        deserunt unde quam, iste atque porro neque
                        possimus necessitatibus eum blanditiis nulla
                        doloribus. Nam.
                      </p>
                      
                      <div class="text-end">
                        <button class="btn btn-warning">Read</button>
                      </div>
                    </div>
                  </div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
```



Buttons & inputs

To follow the web design, I edited the Bootstrap variables with Sass to remove the rounding of the buttons and inputs.



Modal

I have increased the sizes in all the breackpoints modals and I have slowed down the rancidity when it appears.

```
// MODAL
// SIZE
$modal-sm: 400px; //Default
$modal-md: 600px; //500px
$modal-lg: 900px; //800px
$modal-xl: 1240px; //1140px
// TRANSITION
$modal-transition: transform 1s ease-out;
```

Carousel

In carousel's posts I have highlighted the arrows of the corporate blue color because by default they are white and they were not seen with the background of the web. I have also reduced the transition time and now change post is faster.

```
// CAROUSEL                                     //Default
$carousel-control-color:$azul-corp;              //$white
$carousel-transition-duration:.3s;                //.6s
```

