

UD4. Activity 4

CUSTOMIZING THE USERINTERFACE BY USINGCSS STYLES

Toni Barceló García

IES JOAN RAMIS I RAMIS | DAW

Tabla de contenido

UD4. Activity 4	2
Source Github project	3
Source project:.....	3
Root folder for this activity: UD4. Activity 3.....	3
github-pages (latest version).....	3
The website structure	3
Anotations:.....	4
GENERAL.....	5
Fonts, colors and variables.....	5
Position and flex	5
Header	6
Input & button search.....	6
<i>Position and flex</i>	6
Header menu bar	7
<i>Position</i>	7
<i>Flex</i>	7
MAIN.....	8
Aside.....	8
Position.....	8
Flex.....	10
Section.....	11
<i>Position</i>	11
<i>Flex</i>	11
Load More	12
Position Flex	12
Back to top button	12
<i>Position & Flex</i>	12
Footer	13
Footer bar.....	13

UD4. Activity 4

You need to create 2 versions of the same web structure without using Bootstrap.

- The first one using only the display and position properties in order to review some basic CSS.
- The second one using flex properties (and grid layout if you want to).

You have a video, html and images attached to this activity. The idea is to have the same structure:

- A **header** with an icon in the central part, and underneath a menu and a search engine positioned like the example.
- Then, the **main** part of the page will be formed by a sidebar and a section where there will be: a title, a menu with options to filter (which will always be seen, despite scrolling) and the main section where there will be the products or posts.
- Finally, at the bottom there will be a **footer**, where you can add the typical options (privacy policy, copyright, social networks, etc.).

You can:

- Use the given code to do the activity.
- Create a new and better version of it with a different colour palette and images.
- Use your own post design colours, but following this structure.

You have to do a desktop version and a mobile version using media queries on both versions.

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 3

[activities/UD4A4_FLEX_GRID](#)

github-pages (latest version)

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

The website structure

Directory	Files	Concept
root	ud4a4.html	main website file.
	ud4a4_flex.css	structure css file with flex
	ud4a4_position.css	structure css file with position
	DIW_UD4_A4.webm	Example video file
fonts	BAHNSCHRIFT.TTF	typography chosen in the guide style.
img	*.png, *.jpg	images directory for the web.

Anotations:

- I have decided to use the same structure but in the style of my own blog.
- In the html file there are two links to css files. One for ***position*** structure and one for ***flex*** structure. Just ***comment and uncomment*** to use of each type.
- According to the characteristics and elements of the website, I have considered that the mobile versions are those that are below 1000 pixels.

GENERAL

Fonts, colors and variables

First, I add the colors and fonts of my project to variables.

```
/* ----- */
/* FONTS */
/* ----- */

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

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

/* ----- */
/* VARIABLES */
/* ----- */

:root {

  /* COLORS */

  --blanco-corp: #f3f5f7;
  --azul-corp: #001b2e;
  --rojo-corp: #cc0000;
  --amarillo-corp: #ff9900;
  --verde-corp: #294c60;
  --gris-corp: #adb6c4;

  /* FONTS */
  --titulo: 'Bahnschrift';
  --texto: 'Roboto';
}
```

Position and flex

There are general styles that apply in both structures in the desktop and mobile versions, with the exception of the main

Position

```
/* Skip general margins */
* {
  margin: 0;
  padding: 0;
  font-family: var(--texto);
}

main {
  position: relative;
  background-color: var(--blanco-corp);
}

h1 {
  font-family: var(--titulo);
}

a {
  text-decoration: none;
  color: black;
}

a:hover {
  color: var(--amarillo-corp);
  text-decoration: underline;
}

li {
  list-style: none;
}
```

Flex

```
/* Skip general margins */
* {
  margin: 0;
  padding: 0;
  font-family: var(--texto);
}

main {
  display: flex;
  flex: 1;
  background-color: var(--blanco-corp);
}

h1 {
  font-family: var(--titulo);
}

a {
  text-decoration: none;
  color: black;
}

a:hover {
  color: var(--amarillo-corp);
  text-decoration: underline;
}

li {
  list-style: none;
}
```

Header

Input & button search

Position and flex

All the elements are centred. In the mobile version¹ the search button and input are located below the main bar while in the desktop version it is located next to it in absolute position.



```
/* ----- */
/* HEADER */
/* ----- */

header {
  text-align: center;
}

/* button & input search*/
.menu>div {
  position: relative;
  padding: 10px;
}

.menu>div>:not(#search) {
  background-color: var(--amarillo-corp);
  padding: 5px 10px;
  border: none;
}

#search {
  border: none;
  padding: 5px 10px;
}

/* Header menu */
.menu {
  background-color: var(--gris-corp);
  padding: 15px;
}

.menu-element {
  display: inline-block;
  padding: 10px;
}
```

Desktop

```
@media (min-width: 1000px) {

  /* ----- */
  /* HEADER */
  /* ----- */

  .menu>div {
    position: absolute;
    top: 93px;
    right: 30px;
    padding: 0px;
  }

  .menu>div>:not(#search) {
    background-color: var(--amarillo-corp);
    padding: 5px 10px;
    border: none;
  }

  #search {
    border: none;
    padding: 5px 10px;
  }
}
```

¹ below 1000 pixels

Header menu bar

Position

In the header menu bar I have used the inline-block attribute on *.menu-element* class so that it does not jump from line to line at each *li* tag. Both the mobile and desktop versions behave in the same way.

```
/* Header menu */
.menu {
  background-color: var(--gris-corp);
  padding: 15px;
}

.menu-element {
  display: inline-block;
  padding: 10px;
}
```

Flex

In the flex version, I have made the menu *ul* list to be a flex container, centre justified and the wrap value to scape lines if the resolution is low and not cut off.

```
/* Header menu */
.menu {
  background-color: var(--gris-corp);
  padding: 15px;
}

/* list menu */
.menu>ul {
  display: flex;
  justify-content: center;
  flex-wrap: wrap;
}

.menu>ul>li {
  padding: 10px;
}
```


MAIN

Aside

Position

Desktop

In the desktop version the position is absolute (on a relative main) with zero values on top, left and bottom to fit between header and footer.

```
@media (min-width: 1000px) {
  /* ----- */
  /* HEAD */
  /* ----- */

  /* ASIDE */
  /* ----- */

  aside {
    width: 200px;
    padding: 10px;
    position: absolute;
    top: 0;
    left: 0;
    bottom: 0;
  }

  .list-group {
    margin-bottom: 25px;
    display: block;
  }

  .list-group>ul>li {
    display: block;
  }

  .menu2-element {
    margin-left: 10px;
  }
}
```



Mobile

In the mobile version, position is static so that it is below the header but above the section.

```
/* ----- */
/* ASIDE */
/* ----- */

aside {
  position: static;
  box-sizing: border-box;
  width: 100%;
  padding: 20px 20px 20px 0;
  background-color: var(--blanco-corp);
}

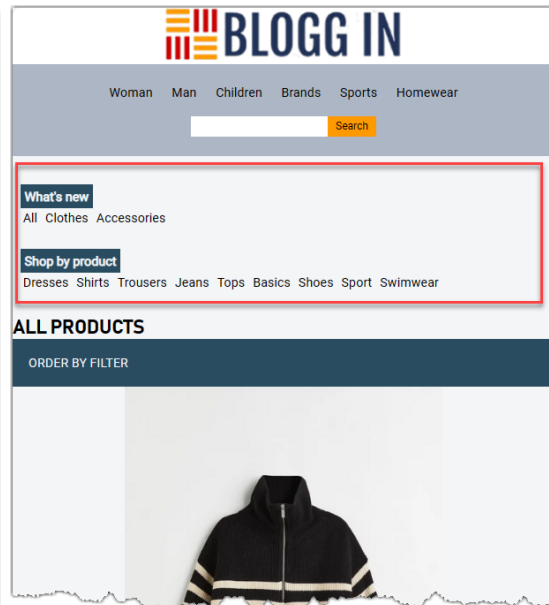
.list-group {
  display: block;
  padding: 10px;
  margin-top: 10px;
}

.list-group>ul {
  margin-top: 5px;
}

.list-group>ul>li {
  display: inline-block;
}

.list-group strong {
  background-color: var(--verde-corp);
  color: var(--blanco-corp);
  width: 100%;
  padding: 5px;
}

.menu2-element {
  padding: 3px;
}
```



Flex

Desktop

The flex container requires a column format and a fixed width of 200px.

```
@media (min-width: 1000px){
```

```
    aside {
        width: 200px;
        padding: 10px;
    }

    .menu2-element {
        padding: 3px;
        margin-left: 10px;
    }

    /* Change menu aside direction */
    .list-group>ul{
        flex-direction: column;
    }
}
```

Mobile

In the mobile version, the direction list group element must be row.

```
/* ----- */
/* ASIDE */
/* ----- */

aside {
    padding: 10px;
}

.list-group{
    padding: 10px;
    margin-top: 10px;
}

.list-group>ul{
    display: flex;
    flex-direction: row;
    margin-top: 5px;
    flex-wrap: wrap;
}

.list-group>strong{
    background-color: var(--verde-corp);
    color: var(--blanco-corp);
    width: 100%;
    padding: 5px;
}

.menu2-element {
    padding: 8px;
}
```

Section

Position

The product section has the sticky position and article has the inline-block property so that it does not make the line break between articles.

Mobile

```
/* ----- */
/* PRODUCTS SECTION */
/* ----- */

/* in mobile version, center product */
section>ul {
  text-align: center;
}

/* Filter Bar */
#products-filter {
  background-color: var(--verde-corp);
  color: var(--blanco-corp);
  padding: 20px;
  position: sticky;
  top: 0;
}

article {
  display: inline-block;
}

.product-name {
  font-weight: bold;
  display: block;
}

.item-details {
  margin-bottom: 5px;
}

/* limit size to show almost 4 products on full screen */
img {
  max-width: 400px;
}
```

Desktop

```
@media (min-width: 1000px) {

}

/* ----- */
/* PRODUCTS SECTION */
/* ----- */

section {
  margin-left: 220px;
}

section>ul {
  text-align: start;
}
```

Flex

```
/* ----- */
/* PRODUCTS SECTION */
/* ----- */

section {
  display: flex;
  flex-direction: column;
  flex: 1;
  margin-top: 10px;
}

/* Filter Bar */
#products-filter {
  background-color: var(--verde-corp);
  color: var(--blanco-corp);
  padding: 20px;
  position: sticky;
  top: 0;
}

article {
  display: inline-flex;
  flex-direction: column;
}

article>div {
  display: flex;
  flex-direction: column;
  justify-content: flex-start;
  flex-wrap: wrap;
}
```

Load More

Position

```
/* Load more */
#load-more-products {
  text-align: center;
  padding: 30px;
}

#load-more-products>button {
  background-color: var(--rojo-corp);
  color: var(--blanco-corp);
  padding: 10px 32px;
  border: none;
  border-bottom-left-radius: 50px;
  border-bottom-right-radius: 50px;
}
```

Flex

```
/* Load more */

#load-more-products {
  display: flex;
  justify-content: center;
  align-items: center;
  padding: 24px;
}

#load-more-products>button {
  background-color: var(--rojo-corp);
  color: var(--blanco-corp);
  padding: 10px 32px;
  border: none;
  border-bottom-left-radius: 50px;
  border-bottom-right-radius: 50px;
}
```

Back to top button

Position & Flex

In both versions the position is fixed so that it is displayed by scrolling.

```
#back-to-top {
  position: fixed;
  bottom: 80px;
  right: 20px;
  background-color: var(--amarillo-corp);
  padding: 10px 32px;
  border: none;
  border-top-left-radius: 50px;
  border-top-right-radius: 50px;
}
```

Footer

Footer bar

The footer bar simply requires centring the elements. A *ul* list and a *div* element. Depending on the structure it requires *text-align* for position or *justify-content* for flex. Both desktop and mobile have the same design.

Position

```
/* ----- */
/* FOOTER */
/* ----- */

footer {
  background-color: var(--gris-corp);
  text-align: center;
}
```

Flex

```
/* ----- */
/* FOOTER */
/* ----- */

footer{
  background-color: var(--gris-corp);
}

footer>ul>li{
  display: flex;
  justify-content: center;
}

footer>div{
  display: flex;
  justify-content: center;
}
```