PEC 1- MARTA PERELLÓ LÓPEZ

Antes de empezar a escribir código y a decidir de qué iba a realizar mi proyecto, he clonado el repositorio en Visual Studio Code usando este enlace https://github.com/uoc-advanced-html-css/uoc-boilerplate.git. Posteriormente opté por hacer mi trabajo de la Puerta de Alcalá ya que soy de Madrid y es uno de los sitios más bonitos de la capital. Tenía el entorno de trabajo preparado ya que cursé otras asignaturas de HTML y CSS, por lo que ya me sonaba el funcionamiento de Npm, Parcel y Netlify.

He querido realizar un diseño sencillo utilizando un menú, imágenes y vídeos responsive acompañados de información acerca del monumento en cuestión. Es una página que sigue una aproximación Mobile First ya que se puede acceder al contenido de manera sencilla, contiene un menú corto, los enlaces son visibles y se puede hacer click en ellos.

El código HTML es muy sencillo: consta de un <header>, un <nav> para el menú de navegación principal y distintos donde he acumulado la información insertando <img>, <iframe>, <ul>. Por último, un <footer>.

He utilizado la dependencia de Font Awesome en mi proyecto para poder hacer uso de los iconos que ofrece. Para obtenerla he puesto el comando correspondiente (npm install --save @fortawesome/fontawesome-free) en la terminal y el script de mi kit en el <head> tras haberme creado una cuenta. Después he elegido los iconos que quería utilizar y los he añadido en el <body>, en mi caso, en el menú principal y en el <footer>.

Las lecturas me han hecho comprender la importancia del uso de las metodologías, ya que aportan extremada limpieza y orden. Es importante no tener un proyecto caótico porque los cambios que se van realizando sobre la marcha se van olvidando, y es crucial aclarar de dónde sale cada trozo de código ya no solo para uno mismo, si no también para los usuarios que quieran ver el proyecto. En mi caso, he combinado la metodología BEM y OOCSS. Ambas van de la mano, ya que la metodología OOCSS utiliza la misma arquitectura que BEM; misma sintaxis de clases. Ayudan a reutilizar código de manera sencilla y además se solventan el 'problema' de la especificidad y eficiencia.

BEM:

Este método consiste en tres conceptos básicos: bloques, elementos y modificadores y se basa en el uso de clases. He usado el anidado de acuerdo a la metodología BEM y las pseudo clases utilizando &__ y el nombre la clase del elemento del bloque.

He dividido mi trabajo en distintos bloques: el primero hace referencia al <header>. La clase se llama así y el elemento que depende del bloque se denomina header__title. Para el menú de navegación he utilizado la clase “menú” y sus elementos de denominan menú__list y menú__option. Después encontramos tres bloques más formados por llamados “contenedor”, “contenedor2”, “contenedor3”, “contenedor4” con sus respectivos elementos en su interior. Por último, la clase “footer”.

OOCSS:

La metodología OOCSS se basa en dos ideas básicas: separar la estructura del aspecto y el contenedor del contenido. En mi caso me he centrado en la primera regla. El <header> y el <nav> compartían las propiedades de width y height, por lo que he creado una clase denominada “vw-100vh-10” que define esas propiedades comunes para más adelante darles estilo por individual con la metodología BEM. Esto hace que se reduzca la cantidad de código que hay que generar. También he creado una clase llamada “center” que contiene la propiedad text-align: center, ya que los títulos de mi proyecto todos están alineados en el centro de la página.

Smacss divide los estilos en distintas partes para que el código quede más organizado. En este caso, dentro de la carpeta “styles” he importado las parciales en el fichero _main.scss, las variables dentro de _variables.scss y los mixin en _mixin.scss. Por otro lado, dentro de “layouts” tenemos _home.scss y un fichero para cada bloque de la metodología BEM.

En home.scss he añadido las clases comunes básicas y luego he creado distintos ficheros basándome en la aproximación Flat donde se encuentra el código de cada bloque con sus respectivos elementos: menú.scss, contenedor.scss, contenedor2.scss, contenedor3.scss, contenedor4.scss y footer.scss.

He creado variables definiendo colores que me pueden ser útiles más adelante y las he añadido al fichero _variables.scss. Además, he añadido una nueva carpeta denominada mixins.scss y la he vinculado a mi proyecto con @import “mixin” en el fichero _main.scss.

He creado una función donde está el estilo, el cual es variable:

```
@mixin bg-color ($color) {
 background-color:$color;
}
```

Otra función que incluye las propiedades de una imagen responsive, para no tener que estar escribiéndolas de nuevo cada vez que quiera incluir una imagen:

```
@mixin propiedades-imagen {
 width: 100%;
 height: auto;
```

```
}
```

Y por último, una función que define la estructura del <body>, útil para proyectos futuros:

```
@mixin center-content {
  display: flex;
  flex-direction: column;
  height: 100vh;
  margin: 0;
  font-family: $font-family-sans-serif;
  font-size: 16px;
  line-height: 24px;
 }
```

En main.scss he importado los diferentes ficheros que contienen la información de mi trabajo escribiendo @import y el nombre de la ruta entre comillas:

```
@import "variables";
@import "mixin";
@import "layouts/home";
@import "layouts/menu";
@import "layouts/contenedor";
@import "layouts/contenedor2";
@import "layouts/contenedor3";
@import "layouts/contenedor4";
@import "layouts/footer";
```

STYLELINT

En cuanto a la instalación de Stylelint, he seguido los pasos ofrecidos en el módulo 2. Una vez escrito el comando correspondiente en mi terminal (npm install --save-dev stylelint-scss stylelint-config-recommended-scss), he creado una nueva carpeta denominada .stylelintrc.json. De esta manera he podido establecer una regla personalizada a la metodología BEM para ver si las clases estaban bien escritas. No me ha aparecido ningún error a la hora de ejecutar el comando npm run stylelint.

```
{
  "extends": ["stylelint-config-recommended-scss"],
  "rules": {
    "selector-class-pattern": "^.[a-z]([a-z0-9-]+)?(__([a-z0-9]+-?)+)?(--
([a-z0-9]+-?)+){0,2}$"
    }
  }
```

GITHUB Y NETLIFY

https://github.com/Martaperello/PEC-1--MARTA-PERELL---PUERTA-DE-ALCALA.git

https://jalea-suave-a7a7aa.netlify.app