

# HTML y CSS

Repositorio: <https://github.com/Garcia091/sesion-dos-aprendices-2021.git>



1. Crear la estructura básica de html
2. Vamos a crear la barra de navegación en el header de la aplicación, para tal fin necesitamos las siguientes etiquetas:
  - <header>
  - <nav>
  - <ul>
  - <li>

```
<header>
  <nav>
    <ul class="menu">
      <li class="lista"><a href="index.html">Inicio</a></li>
      <li class="lista"><a href="#table-card">Tabla</a></li>
      <li class="lista lista-r"><a href="formulario.htm">Contactenos</a></li>
    </ul>
  </nav>
</header>
```

Como podemos ver en la imagen tenemos dos formas de vincular los elementos en html:

1. Enlace a un archivo externo
2. Enlace interno, para navegar internamente por la página debemos usar el selector de tipo id, para poder enlazarlas agregamos el “#” + “NombreSelector”

CSS del Navbar:

El contenedor principal Nav será de color gris y un ancho del 100% de la página

```
nav{
  background-color: lightslategray;
  width: 100%;
}
```

El selector de clase “.menu” representa la lista de la barra de navegación <ul>, los estilos deseados para este elemento son: queremos una lista sin estilos que nos permita consumir el 100% del contenedor <Nav>, adicional quitamos los espacio de la margen y del relleno de dicho elemento.

```
.menu{
  list-style: none;
  margin: 0;
  padding: 0;
  width: 100%;
}

.menu .lista{
  display: inline-block;
}
```

Para los elemento <li>, los estamos representando con su selector de tipo clase “.lista”, para el contenido de la lista asignamos la propiedad de display para lograr que la lista se vea de forma horizontal, ya que los elementos <li> tiene por defecto la distribución de etiqueta de bloque.

```
.lista-r{
  float: right;
}
```



Para lograr tener el efecto de elementos separados lo hacemos con la propiedad float y la dirección que desea.

```
.lista a{
  text-decoration: none;
  color: white;
  display: block;
  padding: 15px 50px;
}

.lista a:hover{
  background-color: tomato;
}
```

Para los elementos de enlace <a>, le quitamos los estilos por defecto de hipervínculo, le agregamos padding para lograr el espacio deseado, adicional agregamos la propiedad “display: block” para que ocupe el espacio asignado.

Para los elementos de enlace se le agrega una pseudoclase para obtener un menor diseño de la barra de navegación

### 3. Crear las tarjetas de los Super Heroes



#### Estructura HTML

Para lograr 1 tarjeta se implementan las siguientes etiquetas:

```
<main>
  <div class="container">
    <div class="container-grid">
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-spider.jpg');">
        <h3>Spider Man</h3>
      </div>
    </div>
  </div>
</main>
```

La división del panel se debe repetir la cantidad de tarjetas deseadas, así como se ve en la imagen:

```

<main>
  <div class="container">
    <div class="container-grid">
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-spider.jpg');">
        <h3>Spider Man</h3>
      </div>
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-captain.jpg');">
        <h3>Captain America</h3>
      </div>
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-iron.jpg');">
        <h3>Iron Man</h3>
      </div>
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-thor.jpg');">
        <h3>Thor</h3>
      </div>
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-hulk.jpg');">
        <h3>Hulk</h3>
      </div>
      <div class="panel "
        style="background-image: url('https://raw.githubusercontent.com/jennymontoya1001/heroes/main/marvel-wolverine.jpg');">
        <h3>Wolverine</h3>
      </div>
    </div>
  </div>

```

CSS: para lograr el diseño deseado vamos a implementar los siguientes elementos:

El contenedor principal estará distribuido con el display flex, el cual le permite organizar sus elementos en columnas o filas. Los elementos hijo estarán centrados y su tamaño de alto será el 100% del tamaño de la pantalla.

```

/* Cards*/
.container{
  display: flex;
  flex-wrap: wrap;
  align-items: center;
  justify-content: center;
  height: 100vh;
  overflow: hidden;
  margin: 0;
}

```

Contamos con otro contenedor principal, de un tamaño del 90% del tamaño de su padre y tener una distribución de display:flex

```

.container-flex{
  display: flex;
  width: 90vh;
}

```

Las tarjetas o ítems del flex, permitirá que su fondo tenga la mejor presentación, cada una de las tarjetas tendrán el 80% del tamaño del padre, tendrá los bordes redondeados.

La propiedad flex: 0.5 le dará el tamaño a la tarjeta que no esté seleccionada.

En este ejercicio aplicación animaciones y transformaciones de los elementos.

```

.panel{
  background-size: auto 100%;
  background-position: center;
  background-repeat: no-repeat;
  height: 80vh;
  border-radius: 50px;
  color: black;
  cursor: pointer;
  flex:0.5;
  margin: 10px;
  position: relative;
  transition: flex 0.7s ease-in;
  border: 2px dotted white;
  animation: moverArriba 1s ease-in;
}

```

```

@keyframes moverArriba{
  0%{
    opacity: 0;
    transform: translateY(100%);
  }
  100%{
    opacity:1;
    transform: translate(0);
  }
}

```

Las animaciones se asignan con la palabra **keyframes** seguido del comportamiento deseado.

El texto al interior de la tarjeta tiene los siguientes estilos:

El título tendrá una posición absoluta, teniendo como padre al “.panel” con posición relativa y así desplazarse como se desee al interior de la tarjeta. La propiedad opacity permite desaparecer el título de la tarjeta y así solo mostrar el contenido del <h3> cuando la tarjeta este activa.

```

.panel h3{
  font-size: 24px;
  position: absolute;
  bottom: 20px;
  left:20px;
  margin: 0;
  opacity: 0;
  background-color: white;
  border-radius: 20%;
  border: 2px solid blueviolet;
  padding: 5px;
}

```

Agregamos los estilos necesarios para obtener la tarjeta activa con un tamaño más grande que las otras tarjetas.

```
.panel.active{
  flex: 5;
  border: 2px solid white;
}

.panel.active h3{
  opacity: 1;
  transition: opacity 0.3s ease-in 0.4s;
}
```



Para que las tarjetas se adapten al tamaño de la pantalla utilizamos la propiedad @media (media Query).

La pseudo-clase **:nth-of-type()** de CSS selecciona uno o más elementos de un tipo dado, en función de su posición entre un grupo de hermanos.

```
@media screen and(max-width:480px) {
  .container{
    width: 100vw;
  }

  .panel:nth-of-type(4),
  .panel:nth-of-type(5){
    display: none;
  }

  .panel h3{
    font-size: 18px;
  }
}
```

Para terminar el ejercicio debemos agregar elementos de JavaScript para lograr la tarjeta activa de forma dinámica.

- Utilizar un `querySelectorAll` para llamar a todos los elementos con la clase `.panel`
- Utilizamos un `forEach` para recorrer todos los elementos del panel
- Agregamos un evento de tipo `click` para estar escuchando cuando se debe activar la propiedad de CSS de la tarjeta.
- Creamos una función para remover el efecto de la tarjeta activa.

```

script > JS script.js > ...
1  const panels = document.querySelectorAll('.panel')
2
3  panels.forEach(panel =>{
4      panel.addEventListener('click', ()=>{
5          removeActiveClass()
6          panel.classList.add('active')
7      })
8  })
9
10 function removeActiveClass(){
11     panels.forEach(panel =>{
12         panel.classList.remove('active')
13     })
14 }
15

```

- La propiedad `classList`:  
Usar `classList` es una forma práctica de acceder a la lista de clases de un elemento como una cadena de texto delimitada por espacios a través de `element.className`.

### Métodos de `classList`

**Add:** Añade las clases indicadas

**Remove:** Elimina las clases indicadas.

**Nota:** Eliminar una clase que no existe NO produce un error.

**Ítem:** Devuelve el valor de la clase por índice en la colección.

**Toggle:** Cuando sólo hay un argumento presente:

**Contains:** Comprueba si la clase indicada existe en el atributo de clase del elemento.

**Replace:** Reemplaza una clase existente por una nueva.

**Tabla:**

Agregar la tabla al interior de la etiqueta <main>. La tabla tiene una cabecera y un cuerpo

```
<div class="container-table">
  <table id="table-card">
    <thead>
      <tr>
        <th>id</th>
        <th>superhero</th>
        <th>publisher</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Spider Man</td>
        <td>Marvel Comics</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Captain America</td>
        <td>Marvel Comics</td>
      </tr>
      <tr>
        <td>3</td>
        <td>Iron Man</td>
        <td>Marvel Comics</td>
      </tr>
      <tr>
        <td>4</td>
        <td>Thor</td>
        <td>Marvel Comic</td>
      </tr>
    </tbody>
  </table>
</div>
</main>
```

CSS de la tabla:

Para el contenedor principal de la tabla le asignamos una margen y el ancho de la tabla

```
✓ .container-table{
  margin:150px auto;
  width: 80vw;
}
```

Para la tabla, le asignaremos un fondo blanco, el texto debe ser centrado. El contenido de la tabla es del 100% del tamaño del contenedor padre.

Agregamos la propiedad border-collapse para separar los border.



```
table{
  background-color: white;
  text-align: left;
  width: 100%;
  border-collapse: collapse;
  animation: moverDerecha 5s ease-in;
}
```

En este ejercicio vemos un concepto nuevo llamado animaciones, para agregar animaciones debemos crear la siguiente estructura:

```
@keyframes moverDerecha{
  0%{
    opacity: 0;
    transform: translateX(100%);
  }

  100%{
    opacity:1;
    transform: translate(0);
  }
}
```

Para las filas le asignamos un padding para separar el contenido del borde.

Para la cabecera de la tabla, asignamos un fondo con un borde en la parte inferior y un tamaño de fuente.

```
td, th{
  padding: 20px;
}

thead{
  background-color: teal;
  border-bottom: 5px solid #0f362d;
  color: white;
  font-size:24px;
}

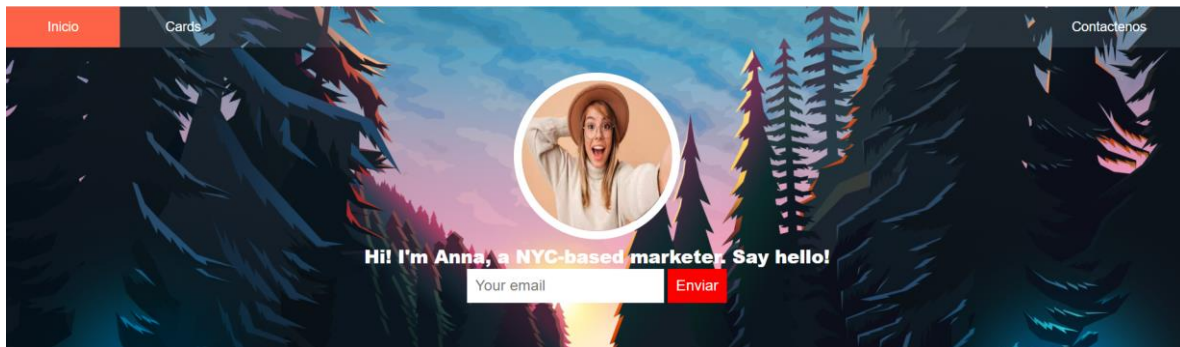
tr:nth-child(even){
  background-color: #ddd;
}

tr:hover td{
  background-color: rgba(0, 128, 128, 0.479);
  color: white;
}
```

Con la pseudoclase :nth-child le agregamos un efecto de fondo a la tabla

id	superhero	publisher
1	Spide Man	Marvel Comics
2	Captain America	Marvel Comics
3	Iron Man	Marvel Comics
4	Thor	Marvel Comics

## Formulario:



## HTML:

```
<> formulario.htm > html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <link rel="stylesheet" href="./style/formulario.css">
8      <title>Document</title>
9  </head>
10
11  <body>
12      <header>
13          <nav>
14              <ul class="menu">
15                  <li class="lista"><a href="index.html">Inicio</a></li>
16                  <li class="lista"><a href="index.html">Cards</a></li>
17                  <li class="lista lista-r"><a href="formulario.htm">Contactenos</a></li>
18              </ul>
19          </nav>
20      </header>
21      <main>
22          
25              <p>Hi! I'm Anna, a NYC-based marketer. Say hello!</p>
26              <input type="email" placeholder="Your email">
27              <input type="submit">
28          </form>
29      </main>
30  </body>
```

Agregamos la estructura de la barra de navegación utilizada en el ejercicio anterior. Para este ejercicio utilizamos otra hoja de HTML diferente.

CSS:

1. Agregar fondo a la página

```
style > # formulario.css > 🚩 nav
1  *{
2    box-sizing: border-box;
3    margin: 0;
4  }
5
6  body {
7    background: url('https://images5.alphacoders.com/964/964482.jpg');
8    background-size: cover;
9    background-position: center;
10   background-repeat: no-repeat;
11   color: white;
12   font-family: helvetica;
13 }
```

2. Agregar los estilos de la barra de navegación

```
16  nav{
17    background-color: rgba(119, 136, 153, 0.212);
18    width: 100%;
19  }
20
21  .menu{
22    list-style: none;
23    margin: 0;
24    padding: 0;
25    width: 100%;
26  }
27
28  .menu .lista{
29    display: inline-block;
30  }
31
32  .lista-r{
33    float: right;
34  }
35
36  .lista a{
37    text-decoration: none;
38    color: white;
39    display: block;
40    padding: 15px 50px;
41  }
42
43  .lista a:hover{
44    background-color: tomato;
45  }
```

3. Dale estilos a la imagen y al formulario

```
main{
  text-align: center;
}

p{
  font-size:22px;
  font-weight: 800;
}

img{
  margin-top:30px;
  border-radius: 50% ;
  height: 200px;
  width: 200px;
  border: 10px solid ■white ;
}

input{
  border:0;
  padding:10px;
  font-size:18px;
}

input[type="submit"]{
  background: ■red;
  color: ■white;
}
```