

Aprende HTML

Iniciación para principiantes

David Moreno Lumbrales
david.morenolu@urjc.es

<https://dlumbrer.gitlab.io/curso-html>

CTIF Madrid-Sur

Terminología

Plan

Introducción

Terminología

HTML (HyperText Markup Language): Lenguaje de marcado utilizado para crear páginas web.

Etiqueta (Tag): Elemento de HTML que define la estructura y el contenido de una página web. Se escribe entre corchetes angulares, por ejemplo, `<html>`, `<head>`, `<body>`.

Elemento (Element): Componente completo de HTML, incluyendo la etiqueta de apertura, el contenido y la etiqueta de cierre. Ejemplo: `<p>Este es un párrafo.</p>`.

Atributo (Attribute): Información adicional sobre un elemento HTML, escrita en la etiqueta de apertura. Ejemplo: ``.

Documento HTML: Archivo que contiene el código HTML y que es interpretado por los navegadores web para mostrar una página.

CSS (Cascading Style Sheets): Lenguaje de hojas de estilo utilizado para describir la presentación de un documento HTML. Permite aplicar estilos como colores, fuentes y diseño de página.

Códigos JS (JavaScript): Lenguaje de programación utilizado para crear contenido web dinámico e interactivo. Se puede insertar directamente en un documento HTML o enlazarse como un archivo externo.

DOM (Document Object Model): Representación estructural del documento HTML en forma de árbol. Permite a los lenguajes de programación como JavaScript interactuar con y modificar la estructura y contenido de la página web.

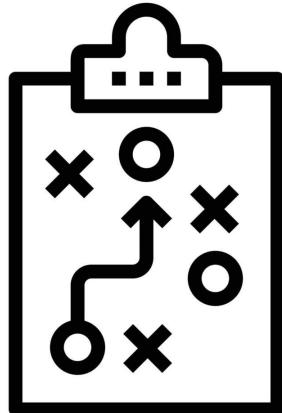
Servidor HTTP: Software que sirve contenido web a los navegadores. Cuando se solicita una página web, el servidor HTTP responde enviando el documento HTML correspondiente.

GitHub: Plataforma de alojamiento de código fuente que utiliza Git para el control de versiones. Permite a los desarrolladores colaborar en proyectos, compartir código y desplegar sitios web mediante GitHub Pages.

GitLab: Similar a GitHub, GitLab es una plataforma de gestión de repositorios que también utiliza Git para el control de versiones. Ofrece características de integración continua y despliegue continuo (CI/CD), y puede ser utilizado para alojar y colaborar en proyectos de desarrollo web.



Plan



Sesión 1: Introducción a HTML

Sesión 2: Estructuración de Contenido

Ejercicio intermedio

Sesión 3: CSS

Sesión 4: Introducción a Bootstrap

Sesión 5: Despliegue en GitHub Pages y Otras Bibliotecas

Sesión 6: Presentación de Proyectos y Revisión Final

¡Lo vamos adaptando según avancemos!

Primero, vamos a conocernos!



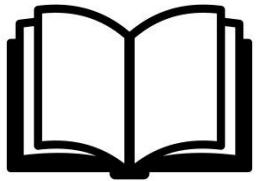
¿Cómo te llamas?

¿Que clase das?

¿Que esperas de este curso, o cómo aplicarías HTML en tus
asignaturas?

Recursos

Aula virtual



<https://dlumbrer.gitlab.io/curso-html>

Introducción a HTML

Sesión 1

¿Que es HTML?

HTML es el **lenguaje de marcado** (OJO, no es un lenguaje de programación) estándar utilizado para **crear y diseñar páginas web**. Proporciona la estructura básica de una página web, que es mejorada y modificada por otras tecnologías como **CSS** y **JavaScript** (que introduciremos más adelante).



Marcado = usar etiquetas (tags)

Estructura definida

Despliegue mediante servidores **HTTP**

HTML 1.0 (1991) - HTML 2.0 (1995) - ... - HTML5 (2015)

Estructura básica

Declaración DOCTYPE y html:

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  </html>
```

Estructura básica

Cabecera



```
1 <head>
2     <meta charset="UTF-8">
3     <meta name="viewport" content="width=device-width, initial-scale=1.0">
4     <title>Document</title>
5 </head>
```

Estructura básica

Cuerpo



```
1  <body>
2      <p>Hola!</p>
3  </body>
```

Ejercicio Práctico o

Uso del Entorno de Desarrollo (IDE)

Todo junto



```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9      <p>Hola!</p>
10 </body>
11 </html>
```

Entorno de desarrollo

Visual Studio Code

+

Live Server extension

(¿Por qué no abrir directamente el archivo en el navegador?)

+

Dev Tools Firefox/Chrome

¿Qué son las etiquetas?

```
<etiqueta atributo = "valor"> contenido </etiqueta>
```

Etiquetas: Las etiquetas HTML son las unidades básicas que definen la estructura y el contenido de una página web.

Las etiquetas generalmente vienen en pares: una etiqueta de apertura y una de cierre. Ejemplo: `<p>Este es un párrafo.</p>`

Etiqueta de apertura: Comienza con un corchete angular izquierdo (`<`), seguido del nombre de la etiqueta, y termina con un corchete angular derecho (`>`).

Etiqueta de cierre: Similar a la etiqueta de apertura, pero con una barra inclinada (`/`) antes del nombre de la etiqueta.

Etiquetas auto-cerradas: Algunas etiquetas no necesitan una etiqueta de cierre. Se cierran automáticamente. Ejemplo: ``.

Etiquetas básicas

Etiquetas de Encabezado (<h1> a <h6>)

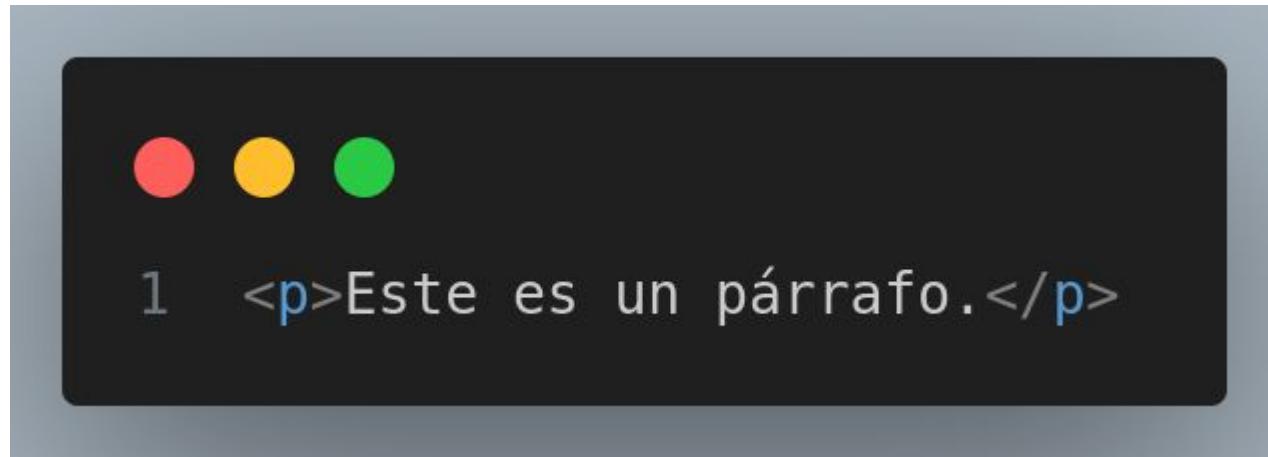


```
1  <body>
2      <h1>Encabezado 1</h1>
3      <h2>Encabezado 2</h2>
4      <h3>Encabezado 3</h3>
5  </body>
```

Etiquetas básicas

Etiqueta de Párrafo (`<p>`)

Los párrafos son bloques de texto que se muestran con un espacio antes y después en el navegador.



¿Qué son los atributos?

```
<etiqueta atributo = "valor"> contenido </etiqueta>
```

Atributos: Los atributos proporcionan información adicional sobre un elemento HTML.

Se colocan dentro de la etiqueta de apertura y consisten en un nombre de atributo y un valor, separados por un signo igual (=). Ejemplo: ``.

Atributos booleanos: Algunos atributos no necesitan un valor; su mera presencia indica que están activos. Ejemplo: `<input type="checkbox" checked>`

Atributos Básicos Generales en HTML

class

- **Descripción:** Define una o más clases para un elemento HTML. Las clases se utilizan principalmente para aplicar estilos CSS o para seleccionar elementos con JavaScript.
- **Uso:** Puedes asignar múltiples clases a un elemento separándolas con espacios.

id

- **Descripción:** Define un identificador único para un elemento HTML. Cada id debe ser único dentro de un documento HTML. Se utiliza principalmente para seleccionar y manipular elementos con CSS y JavaScript.
- **Uso:** El valor del id debe ser único y no puede repetirse en otros elementos.

style

- **Descripción:** Permite aplicar estilos **CSS en línea** (lo veremos más adelante) directamente a un elemento HTML.
- **Uso:** Define estilos CSS específicos dentro del atributo `style`.

title

- **Descripción:** Proporciona información adicional sobre un elemento HTML. Se muestra como una información emergente (tooltip) cuando el usuario pasa el cursor sobre el elemento.
- **Uso:** El valor del atributo `title` es el texto que se mostrará en la información emergente.

Ejercicio Práctico 1

Crear un documento HTML con al menos 2 encabezados y 2 párrafos.

Mi solución (subida a la página del curso)



```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mi primera pagina</title>
8  </head>
9
10 <body>
11     <h1>Hola Mundo!</h1>
12     <p>Esta es mi primera página web</p>
13
14     <h3>Adiós!</h3>
15     <p>Me despido en este otro párrafo.</p>
16
17 </body>
18
19 </html>
```

¿Cómo veo mi resultado?

Servidor HTTP



Abrir archivo

Etiquetas básicas

Etiqueta de Enlace ()

Utilizada para crear hipervínculos que permiten navegar a otras páginas o recursos.

Atributo href: Especifica la URL del recurso vinculado.



```
1 <a href="https://www.dlumbrer.gitlab.io/curso-html">Visita la web del curso pinchando aquí!</a>
2
```

Etiquetas básicas

Etiqueta de Imagen ()

Utilizada para insertar imágenes en una página web.

Atributos **src** y **alt**:

- **src**: Especifica la ruta de la imagen.
- **alt**: Proporciona un texto alternativo para la imagen.



```
1 <!-- Imagen local -->
2 
3
4 <!-- Imagen remota -->
5 
6
```

Etiquetas básicas

Atributos `width` y `height`: Estos atributos permiten especificar el ancho y alto de una imagen directamente en HTML.

Ojo, en **pixeles**.



```
1 <!-- Imagen con tamaño específico -->
2 
3
```

¿Donde encontrar imágenes?

OJO! Poner una imagen en una web sin saber los derechos puede ser **peligroso!**

Wikimedia Commons = Imágenes libres y te enseñan cómo citar el autor

También hay videos, sonidos, etc...

<https://commons.wikimedia.org/wiki/Category:Images>

Ejercicio Práctico 2

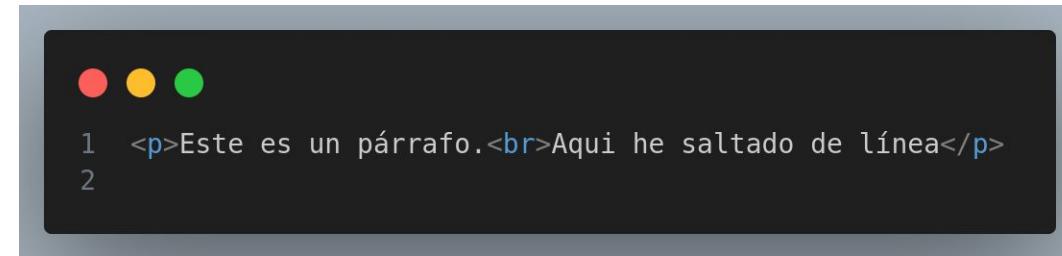
Añadir imágenes y un enlace

Mi solución (subida a la página del curso)

```
● ● ●
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Mi primera pagina</title>
8  </head>
9
10 <body>
11     <h1>Hola Mundo!</h1>
12     <p>Esta es mi primera página web</p>
13
14     <a href="https://www.dlumbrer.gitlab.io/curso-html">Visita la web del curso pinchando aquí!</a>
15
16     <!-- Imagen local -->
17     
18
19     <!-- Imagen remota -->
20     
21
22     <!-- Imagen con tamaño específico -->
23     
24
25     <h3>Adiós!</h3>
26     <p>Me despido en este otro párrafo.</p>
27
28 </body>
29
30 </html>
```

Etiquetas básicas

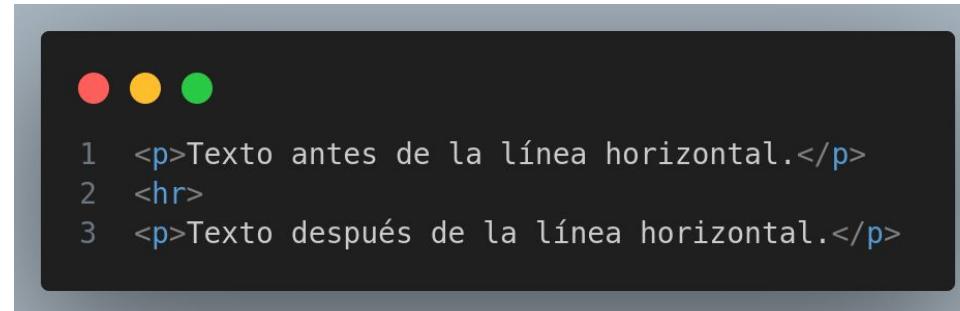
**Eiqueta de Salto de Línea (`
`):** Utilizada para insertar un salto de línea en el texto, similar a presionar "Enter" en un editor de texto.



The screenshot shows a dark-themed code editor window. At the top left are three colored circular icons: red, yellow, and green. Below them, the code is displayed in white text on a black background:

```
1 <p>Este es un párrafo.<br>Aqui he saltado de línea</p>
2
```

Eiqueta de Línea Horizontal (`<hr>`): Utilizada para insertar una línea horizontal en la página, que generalmente se usa para separar contenido.



The screenshot shows a dark-themed code editor window. At the top left are three colored circular icons: red, yellow, and green. Below them, the code is displayed in white text on a black background:

```
1 <p>Texto antes de la línea horizontal.</p>
2 <hr>
3 <p>Texto después de la línea horizontal.</p>
```

Listas en HTML

Lista Desordenada (): Define una lista de elementos sin orden específico.

Elemento de Lista (): Define un elemento dentro de una lista desordenada u ordenada. Siempre dentro de un parente

 o



```
1  <ul>
2      <li>Elemento</li>
3      <li>Otro Elemento</li>
4      <li>Y otro Elemento</li>
5  </ul>
```



```
1  <ol>
2      <li>Primer elemento</li>
3      <li>Segundo elemento</li>
4      <li>Tercer elemento</li>
5  </ol>
```

Ejercicio Práctico 3

Añade una lista ordenada y otra desordenada
junto a saltos de línea y líneas horizontales

Mi solución (subida a la página del curso)

```
● ● ●  
1  <!DOCTYPE html>  
2  <html lang="en">  
3  
4  <head>  
5      <meta charset="UTF-8">  
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">  
7      <title>Mi primera pagina</title>  
8  </head>  
9  
10 <body>  
11  
12     <p>Este es un párrafo.<br>Aqui he saltado de línea</p>  
13  
14     <ul>  
15         <li>Elemento</li>  
16         <li>Otro Elemento</li>  
17         <li>Y otro Elemento</li>  
18     </ul>  
19  
20     <p>Texto antes de la línea horizontal.</p>  
21     <hr>  
22     <p>Texto después de la línea horizontal.</p>  
23  
24     <ol>  
25         <li>Primer elemento</li>  
26         <li>Segundo elemento</li>  
27         <li>Tercer elemento</li>  
28     </ol>  
29  
30  
31 </body>  
32  
33 </html>
```

Estructuración de Contenido

Introducción a CSS

Sesión 2

Favicon

El **favicon** es un pequeño ícono que aparece en la pestaña del navegador junto al título de la página web. Ayuda a identificar visualmente la página web en las pestañas abiertas.

Ojo, tiene un **href** como los enlaces!



```
1 <head>
2   <meta charset="UTF-8">
3   <meta name="viewport" content="width=device-width, initial-scale=1.0">
4   <title>Mi segunda pagina</title>
5   <link rel="icon" href="favicon.ico" type="image/x-icon">
6 </head>
```

rel="icon": Especifica que el enlace es un ícono.

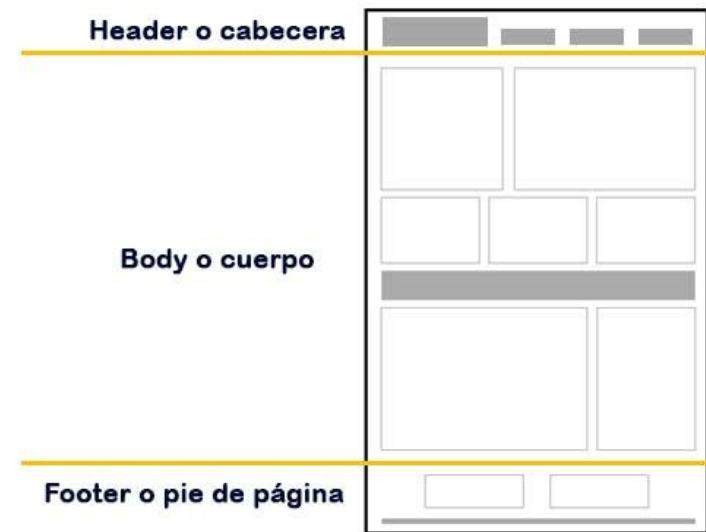
href="favicon.ico": Especifica la ruta al archivo del favicon.

type="image/x-icon": Especifica el tipo MIME del ícono.

También puedes poner **varios favicons** para que funcionen según el tamaño de la pantalla, con el atributo “size”

Estructura de una página web

La estructura de una página web se refiere a cómo se organizan y presentan los elementos en el documento HTML. Una buena estructura facilita la navegación y mejora la accesibilidad y el SEO.



Etiquetas de estructuración

<div> (División)

Etiqueta genérica utilizada para dividir contenido en secciones. No tiene un significado semántico específico.

Se utiliza principalmente para aplicar estilos CSS o scripts JavaScript a secciones del contenido.



```
1 <div class="contenido">
2     <p>Este es un párrafo dentro de un div.</p>
3 </div>
```

Etiquetas de estructuración

<section> (Sección)

Define una sección temática del contenido, generalmente con un título.

Agrupa contenido relacionado.



```
1  <section>
2      <h2>Sección 1</h2>
3      <p>Contenido de la sección 1.</p>
4  </section>
```

Etiquetas de estructuración

Otras etiquetas no tan utilizadas para estructurar contenido

`<article>` (Artículo)

`<aside>` (Barra Lateral)

`<nav>` (Navegación)

Actualmente TODO se puede (y se suele construir) con DIVs!!!!

Junto a CSS y JS, por supuesto

Cabecera y Pie de Página

<header> (Encabezado)

Sección de introducción o conjunto de enlaces de navegación.

Contiene logotipos, títulos y elementos de navegación.

<footer> (Pie de Página)

Sección final del documento, generalmente contiene información de contacto, derechos de autor, etc.

Información de cierre o complementaria.



```
1 <header>
2   <h1>Mi Página Web</h1>
3   <nav>
4     <ul>
5       <li><a href="#inicio">Inicio</a></li>
6       <li><a href="#about">Acerca de</a></li>
7     </ul>
8   </nav>
9 </header>
```



```
1 <footer>
2   <p>&copy; 2024 Mi Página Web. Todos los derechos reservados.</p>
3 </footer>
```

Ejercicio Práctico 1

Crear una página estructurada con <header>, <footer>, <div> y <section>. Que contenga un Favicon.

/sesion1

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Mi segunda pagina</title>
8   <!-- Favicon -->
9   <!-- El favicon se especifica en la sección <head> utilizando la etiqueta <link> con los atributos rel="icon" y href="favicon.ico". -->
10  <link rel="icon" href="favicon.ico" type="image/x-icon">
11 </head>
12
13 <body>
14   <!-- Encabezado -->
15   <!-- El encabezado contiene el título de la página y una barra de navegación con enlaces a diferentes secciones de la página. -->
16   <header>
17     <h1>Mi Página Estructurada</h1>
18     <nav>
19       <ul>
20         <li><a href="#inicio">Inicio</a></li>
21         <li><a href="#sobre">Sobre</a></li>
22         <li><a href="#contacto">Contacto</a></li>
23       </ul>
24     </nav>
25   </header>
26
27   <!-- Contenedor Principal -->
28   <!-- Envuelve las secciones principales del contenido. Utiliza la clase "container" para aplicar estilos de forma más sencilla. -->
29   <div classe="container">
30     <!-- Sección de Inicio -->
31     <!-- Cada sección tiene un identificador único (id) para facilitar la navegación mediante los enlaces del encabezado. -->
32     <section id="inicio">
33       <h2>Inicio</h2>
34       <p>Bienvenido a mi página web. Aquí encontrarás información sobre diversos temas.</p>
35     </section>
36     <!-- Sección Sobre -->
37     <section id="sobre">
38       <h2>Sobre</h2>
39       <p>Esta es una página web de ejemplo creada para aprender sobre la estructura de HTML.</p>
40     </section>
41     <!-- Sección de Contacto -->
42     <section id="contacto">
43       <h2>Contacto</h2>
44       <p>Puedes contactarme a través del correo: ejemplo@correo.com</p>
45     </section>
46   </div>
47
48   <!-- Pie de Página -->
49   <!-- Contiene un mensaje de derechos reservados. -->
50   <footer>
51     <p>&copy; 2024 Mi Página Web. Todos los derechos reservados.</p>
52   </footer>
53 </body>
54
55 </html>
```

Formularios en HTML

Los formularios en HTML permiten a los usuarios **ingresar y enviar datos al servidor**.

Etiqueta Principal: `<form>` define un formulario HTML. Todos los elementos de entrada están contenidos dentro de esta etiqueta.

Atributos Importantes:

`action`: URL a la que se enviarán los datos del formulario.

`method`: Método HTTP para enviar los datos (`GET` o `POST`).

```
1 <form action="/submit" method="post">
2     <!-- Aquí irán los elementos del formulario -->
3 </form>
4
```

Formularios en HTML

Etiqueta <input>

Utilizada para diferentes tipos de entradas de datos.

Atributos Importantes:

type: Especifica el tipo de entrada (`text`, `password`, `email`, `checkbox`, `radio`, etc.).

name: Nombre del campo para identificar los datos en el servidor.

placeholder: Texto de sugerencia dentro del campo.

value: Valor predeterminado del campo.



```
1 <form action="/submit" method="post">
2   <label for="nombre">Nombre:</label>
3   <input type="text" id="nombre" name="nombre" placeholder="Ingresa tu nombre">
4 </form>
```

Formularios en HTML

Etiqueta <label>

Asociada a un elemento de entrada (<input>) para describirlo.

Atributos Importantes:

for: Identificador del elemento de entrada al que se asocia.



```
1 <form action="/submit" method="post">
2   <label for="nombre">Nombre:</label>
3   <input type="text" id="nombre" name="nombre" placeholder="Ingresa tu nombre">
4 </form>
```

Formularios en HTML

Etiqueta <textarea>

Utilizada para entradas de texto largas.

Atributos Importantes:

`rows` y `cols`: Especifican el tamaño del área de texto.



```
1 <form action="/submit" method="post">
2   <label for="mensaje">Mensaje:</label>
3   <textarea id="mensaje" name="mensaje" rows="4" cols="50" placeholder="Escribe tu mensaje aquí"></textarea>
4 </form>
```

Formularios en HTML

Etiqueta <button>

Utilizada para crear botones de envío.

Atributos Importantes:

type: Tipo de botón (`submit`, `reset`, `button`).



```
1 <form action="/submit" method="post">
2   <label for="nombre">Nombre:</label>
3   <input type="text" id="nombre" name="nombre" placeholder="Ingresa tu nombre">
4   <button type="submit">Enviar</button>
5 </form>
```

Formularios en HTML

Etiqueta <select> y <option>

Utilizada para crear botones de envío.

Atributos Importantes:

- <select>: Atributo `name` para identificar el campo.
- <option>: Atributo `value` para el valor enviado al servidor.

```
1  <form action="/submit" method="post">
2      <label for="pais">País:</label>
3      <select id="pais" name="pais">
4          <option value="es">España</option>
5          <option value="mx">México</option>
6          <option value="ar">Argentina</option>
7      </select>
8  </form>
```

Ejercicio Práctico 2

Crear un formulario simple de contacto.
Usa los distintos tipos de input que hay.

/sesion1

```
1 <form action="/submit" method="post">
2     <!-- Campo de texto -->
3     <label for="nombre">Nombre:</label>
4     <input type="text" id="nombre" name="nombre" placeholder="Ingresa tu nombre"><br><br>
5
6     <!-- Campo de correo electrónico -->
7     <label for="email">Correo Electrónico:</label>
8     <input type="email" id="email" name="email" placeholder="Ingresa tu correo"><br><br>
9
10    <!-- Área de texto -->
11    <label for="mensaje">Mensaje:</label>
12    <textarea id="mensaje" name="mensaje" rows="4" cols="50"
13        placeholder="Escribe tu mensaje aquí"></textarea><br><br>
14
15    <!-- Lista desplegable -->
16    <label for="pais">País:</label>
17    <select id="pais" name="pais">
18        <option value="es">España</option>
19        <option value="mx">México</option>
20        <option value="ar">Argentina</option>
21    </select><br><br>
22
23    <!-- Casillas de verificación -->
24    <label>Intereses:</label><br>
25    <label><input type="checkbox" name="intereses" value="musica"> Música</label><br>
26    <label><input type="checkbox" name="intereses" value="deportes"> Deportes</label><br>
27    <label><input type="checkbox" name="intereses" value="lectura"> Lectura</label><br><br>
28
29    <!-- Botones de radio -->
30    <label>Género:</label><br>
31    <label><input type="radio" name="genero" value="masculino"> Masculino</label><br>
32    <label><input type="radio" name="genero" value="femenino"> Femenino</label><br>
33    <label><input type="radio" name="genero" value="otro"> Otro</label><br><br>
34
35    <!-- Botón de envío -->
36    <button type="submit">Enviar</button>
37 </form>
```

Validación de formularios

La validación de formularios asegura que los datos ingresados por los usuarios cumplan con ciertos criterios antes de ser enviados al servidor.

Tipos de Validación:

- **Validación del lado del cliente:** Realizada en el navegador utilizando HTML y JavaScript.
- **Validación del lado del servidor:** Realizada en el servidor después de que los datos se hayan enviado.

Atributos de Validación en HTML5

- **required:** Indica que el campo es obligatorio.
- **minlength y maxlength:** Especifican la longitud mínima y máxima de los datos de entrada.
- **min y max:** Definen los valores mínimo y máximo para los datos de entrada numéricos.
- **pattern:** Define una expresión regular que los datos de entrada deben coincidir.
- **type:** Especifica el tipo de entrada, lo que determina la validación automática para ciertos tipos de datos (por ejemplo, `email`, `url`, `number`).

Ejercicio Práctico 3

Añadir validación al formulario creado.

/sesion2

```
● ○ ● ●  
1 <form action="/submit" method="post">  
2     <!-- Campo de texto obligatorio -->  
3     <label for="nombre">Nombre:</label>  
4     <input type="text" id="nombre" name="nombre" required><br><br>  
5  
6     <!-- Campo de correo electrónico obligatorio -->  
7     <label for="email">Correo Electrónico:</label>  
8     <input type="email" id="email" name="email" required><br><br>  
9  
10    <!-- Área de texto con longitud mínima -->  
11    <label for="mensaje">Mensaje:</label>  
12    <textarea id="mensaje" name="mensaje" rows="4" cols="50" minlength="10" required></textarea><br><br>  
13  
14    <!-- Campo numérico con valores mínimo y máximo -->  
15    <label for="edad">Edad:</label>  
16    <input type="number" id="edad" name="edad" min="18" max="99" required><br><br>  
17  
18    <!-- Campo con patrón de expresión regular -->  
19    <label for="codigo">Código Postal:</label>  
20    <input type="text" id="codigo" name="codigo" pattern="[0-9]{5}" required><br><br>  
21  
22    <!-- Lista desplegable obligatoria -->  
23    <label for="pais">País:</label>  
24    <select id="pais" name="pais" required>  
25        <option value="es">España</option>  
26        <option value="mx">México</option>  
27        <option value="ar">Argentina</option>  
28    </select><br><br>  
29  
30    <!-- Casillas de verificación -->  
31    <label>Intereses:</label><br>  
32    <label><input type="checkbox" name="intereses" value="musica" required> Música</label><br>  
33    <label><input type="checkbox" name="intereses" value="deportes"> Deportes</label><br>  
34    <label><input type="checkbox" name="intereses" value="lectura"> Lectura</label><br><br>  
35  
36    <!-- Botones de radio -->  
37    <label>Género:</label><br>  
38    <label><input type="radio" name="genero" value="masculino" required> Masculino</label><br>  
39    <label><input type="radio" name="genero" value="femenino"> Femenino</label><br>  
40    <label><input type="radio" name="genero" value="otro"> Otro</label><br><br>  
41  
42    <!-- Botón de envío -->  
43    <button type="submit">Enviar</button>  
44 </form>
```

¿Que es CSS?

CSS



Ya sabemos HTML, pero por qué no se parecen las páginas que hacemos a las de la web?

Razón 1: le falta estilo = le falta CSS

Razón 2: le falta comportamiento = le falta JS

CSS (Cascading Style Sheets) Es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML

CSS permite separar el contenido de la presentación, mejorando la flexibilidad y control sobre la apariencia de los elementos en una página web.

Ejemplo muy básico



```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Ejemplo de CSS</title>
7      <style>
8          body {
9              font-family: Arial, sans-serif;
10             background-color: #f0f0f0;
11         }
12         h1 {
13             color: #333;
14         }
15         p {
16             color: #666;
17         }
18     </style>
19 </head>
20 <body>
21     <h1>Bienvenido a CSS</h1>
22     <p>Este es un párrafo estilizado con CSS.</p>
23 </body>
24 </html>
```

Bienvenido a CSS

Este es un párrafo estilizado con CSS.

Sintaxis

Selector: Indica a qué elementos HTML se aplicarán los estilos.

Propiedad: Especifica qué aspecto del elemento se va a modificar.

Valor: Define cómo se va a modificar la propiedad.

Regla CSS: Conjunto de uno o más selectores y declaraciones de estilo.



```
1 selector {  
2     propiedad: valor;  
3 }
```



```
1 h1 {  
2     color: blue;  
3     /* Cambia el color del texto a azul */  
4     font-size: 24px;  
5     /* Establece el tamaño de la fuente a 24 píxeles */  
6 }
```

Selectores básicos

Selector de Elemento

Aplica estilos a todos los elementos de un tipo específico.

```
● ● ●  
1 p {  
2   color: red;  
3   /* Aplica color rojo a todos los párrafos  
4     a todos los elementos p */  
5 }
```

Selector de Clase

Aplica estilos a todos los elementos con una clase específica. Las clases se definen con un punto (.) seguido del nombre de la clase.

```
● ● ●  
1 .miClase {  
2   background-color: yellow;  
3   /* Aplica un fondo amarillo a todos los elementos con la clase "miClase" */  
4 }
```

Selectores básicos

Selector de ID: Aplica estilos a un único elemento con un id específico (Por que los ID son **únicos!!**). Los IDs se definen con un símbolo de almohadilla (#) seguido del nombre del id.



```
1 #miID {  
2     font-weight: bold;  
3     /* Aplica negrita al elemento con el id "miID" */  
4 }
```

Selectores básicos

Se pueden poner **las mismas reglas** a distintos selectores y se pueden **repetir** los selectores.

OJO! Si pisamos atributos CSS prevalecerá el último que se lea/cargue

```
1 h1, h2, h3, .miClase, #miID {  
2   color: #8A8E27;  
3   font-weight: normal;  
4   font-family: Arial, Helvetica, sans-serif;  
5 }  
6 h1 { font-size: 2em; }  
7 h2 { font-size: 1.5em; }  
8 h3 { font-size: 1.2em; }
```

Propiedades básicas

color

- **Descripción:** Cambia el color del texto de un elemento.
- **Valores:** Puede ser un nombre de color, un valor hexadecimal, RGB, RGBA, HSL, HSLA.

font-family

- **Descripción:** Especifica la familia de fuentes para el texto de un elemento.
- **Valores:** Una lista de fuentes, donde la primera disponible será utilizada.

font-size

- **Descripción:** Define el tamaño del texto.
- **Valores:** Puede ser una longitud (px, em, rem) o una palabra clave (`small`, `medium`, `large`).

margin

- **Descripción:** Establece el espacio exterior alrededor de un elemento.
- **Valores:** Puede ser una longitud, un porcentaje o `auto`.

Propiedades básicas

padding

- **Descripción:** Establece el espacio interior entre el contenido de un elemento y su borde.
- **Valores:** Puede ser una longitud o un porcentaje.

border

- **Descripción:** Define el borde alrededor de un elemento.
- **Valores:** Consiste en el ancho, el estilo y el color del borde.

Lista de propiedades disponibles:

<https://www.htmlhelp.com/es/reference/css/properties.html>

Propiedades básicas

Ejemplo de CSS

Este es un párrafo con texto azul y tamaño de fuente de 16 píxeles.

Esta es una caja con fondo amarillo, relleno de 15 píxeles y borde negro de 2 píxeles.

Posicionamiento



```
1 /* Estilo general para los divs */
2 div {
3     width: 200px;
4     height: 100px;
5     padding: 10px;
6     text-align: center;
7     color: white;
8     border: 2px solid black;
9 }
```

Posicionamiento

```
1 /* Div centrado horizontalmente */
2 .centered {
3     width: 50%;
4     /* Centrado horizontalmente */
5     margin: 0 auto;
6     background-color: green;
7 }
8
9 /* Div alineado a la derecha */
10 .right {
11     width: 200px;
12     /* Alinear a la derecha */
13     margin-left: auto;
14     margin-right: 0;
15     background-color: coral;
16 }
```

```
1 /* Div en una posición específica usando píxeles */
2 .specific-px {
3     width: 300px;
4     /* Separado 100px desde la izquierda */
5     margin-left: 100px;
6     /* Separado 150px desde la parte superior */
7     margin-top: 150px;
8     background-color: pink;
9 }
10
11 /* Div en una posición específica usando porcentaje */
12 .specific-percent {
13     width: 30%;
14     /* 20% desde la izquierda */
15     margin-left: 20%;
16     /* 30% desde la parte superior */
17     margin-top: 30%;
18     background-color: blue;
19 }
```

Posicionamiento

Centrado Horizontalmente

Alineado a la Derecha

Posición en pixeles

Ejercicio Práctico 4

Cambiar colores y fuentes de texto, utilizando selectores de elementos, de clase y de ID

/sesion3

```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Ejercicio Práctico 1: CSS</title>
7      <style>
8          /* Selector de Elemento */
9          h1 {
10              color: blue; /* Cambia el color del texto de todos los <h1> a azul */
11             font-family: Arial, sans-serif; /* Cambia la fuente del texto de todos los <h1> */
12             padding: 10px; /* Añade un relleno de 10 pixeles */
13             margin-bottom: 20px; /* Añade un margen inferior de 20 pixeles */
14             border: 2px solid black; /* Añade un borde sólido de 2 pixeles de color negro */
15             background-color: lightgray; /* Cambia el color de fondo a gris claro */
16         }
17
18         /* Selector de Clase */
19         .importante {
20             color: red; /* Cambia el color del texto de los elementos con la clase "importante" a rojo */
21             font-family: 'Courier New', Courier, monospace; /* Cambia la fuente del texto de los elementos con la clase "importante" */
22             padding: 10px; /* Añade un relleno de 10 pixeles */
23             margin: 10px 0; /* Añade un margen superior e inferior de 10 pixeles */
24             border: 1px dashed black; /* Añade un borde discontinuo de 1 pixel de color negro */
25             background-color: lightyellow; /* Cambia el color de fondo a amarillo claro */
26         }
27
28         /* Selector de Clase adicional */
29         .destacado {
30             font-size: 20px; /* Cambia el tamaño del texto de los elementos con la clase "destacado" */
31             font-weight: bold; /* Cambia el texto a negrita para los elementos con la clase "destacado" */
32             padding: 15px; /* Añade un relleno de 15 pixeles */
33             margin: 10px 0; /* Añade un margen superior e inferior de 10 pixeles */
34             border: 1px solid red; /* Añade un borde sólido de 1 pixel de color rojo */
35             background-color: lightpink; /* Cambia el color de fondo a rosa claro */
36         }
37
38         /* Selector de ID */
39         #unico {
40             color: green; /* Cambia el color del texto del elemento con el ID "unico" a verde */
41             font-family: 'Times New Roman', Times, serif; /* Cambia la fuente del texto del elemento con el ID "unico" */
42             padding: 20px; /* Añade un relleno de 20 pixeles */
43             margin: 20px 0; /* Añade un margen superior e inferior de 20 pixeles */
44             border: 2px solid green; /* Añade un borde sólido de 2 pixeles de color verde */
45             background-color: lightgreen; /* Cambia el color de fondo a verde claro */
46         }
47     </style>
48 </head>
49 <body>
50     <h1>Bienvenido a CSS</h1>
51     <p>Este es un párrafo normal.</p>
52     <p class="importante">Este es un párrafo importante.</p>
53     <p class="importante destacado">Este es un párrafo importante y destacado.</p>
54     <p class="destacado">Este es un párrafo destacado.</p>
55     <p id="unico">Este es un párrafo único.</p>
56 </body>
57 </html>
```

Bienvenido a CSS

Este es un párrafo normal.

Este es un párrafo importante.

Este es un párrafo importante y destacado.

Este es un párrafo destacado.

Este es un párrafo único.

Ejercicio Intermedio

Crea una página HTML.

Que tenga enlaces a otro fichero HTML.

Al menos 2 imágenes

Con 1 lista

Uso de encabezados Encabezados

CSS

Tema Libre

Avanzando con CSS

Sesión 3

CSS interno, externo y en linea

No lo sabíamos, pero estábamos usando CSS interno.

Estilos Internos (Internal Styles)

- **Descripción:** Los estilos internos se definen dentro de la sección `<head>` de un documento HTML utilizando la etiqueta `<style>`.
- **Uso:** Útil para aplicar estilos a una única página sin afectar otras páginas.
- **Ventajas:**
 - Mantiene los estilos centralizados dentro del documento HTML.
 - Más limpio que los estilos en línea.
- **Desventajas:**
 - No es reutilizable en otros documentos HTML.
 - Puede llevar a un archivo HTML desordenado si hay muchos estilos.

CSS interno, externo y en linea

Estilos Externos (External Styles)

- **Descripción:** Los estilos externos se definen en un archivo CSS separado y se enlazan al documento HTML utilizando la etiqueta `<link>`.
- **Uso:** Útil para aplicar estilos consistentes a múltiples páginas web.
- **Ventajas:**
 - Facilita el mantenimiento y la actualización de estilos.
 - Reutilizable en múltiples documentos HTML.
 - Mantiene el documento HTML limpio y ordenado.
- **Desventajas:**
 - Requiere un archivo CSS separado.
 - Puede aumentar el tiempo de carga inicial de la página debido a la solicitud adicional del archivo CSS.



```
1 <head>
2   <link rel="stylesheet" type="text/css" href="estilos.css">
3 </head>
```

CSS interno, externo y en linea

Estilos en Línea (Inline Styles)

- **Descripción:** Los estilos en línea se aplican directamente a los elementos HTML utilizando el atributo `style`.
- **Uso:** Útil para aplicar estilos rápidos a elementos individuales sin afectar otros elementos.
- **Ventajas:**
 - Fácil de usar para cambios rápidos.
 - No requiere un archivo CSS separado.
- **Desventajas:**
 - Difícil de mantener y actualizar.
 - No es reutilizable.
 - Puede llevar a un código desordenado.



```
1 <p style="color: red; font-size: 20px;">Texto con estilo en línea</p>
```

CSS interno, externo y en linea

Característica	En Línea	Internos	Externos
Definición	<code>style</code> en el elemento	<code><style></code> en <code><head></code>	Archivo CSS separado
Reutilizable	No	No	Sí
Mantenimiento	Difícil	Moderado	Fácil
Organización	Desordenado	Centralizado en HTML	Separado y limpio
Eficiencia	Rápido para cambios únicos	Útil para una sola página	Ideal para múltiples páginas

Ejercicio Práctico 1

Aplica tu CSS de manera externa y en linea

Solucion (externa)

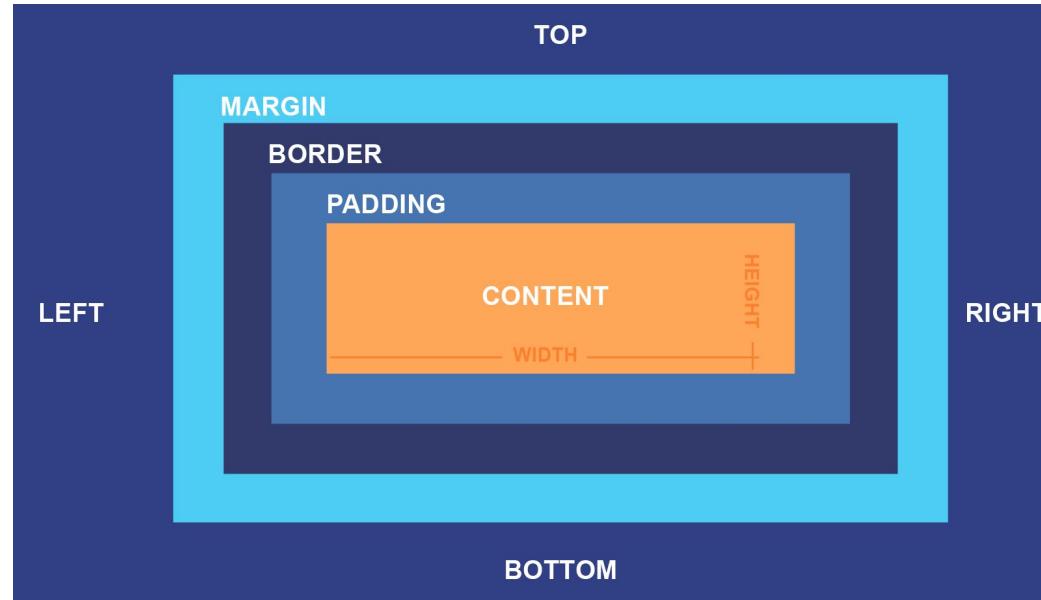


```
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <title>Ejercicio Práctico 1: CSS</title>
7  </head>
8    <link rel="stylesheet" type="text/css" href="ejercicio2.css">
9  </head>
10 </body>
11 <body>
12   <h1>Bienvenido a CSS</h1>
13   <p>Este es un párrafo normal.</p>
14   <p class="importante">Este es un párrafo importante.</p>
15   <p class="importante destacado">Este es un párrafo importante y destacado.</p>
16   <p class="destacado">Este es un párrafo destacado.</p>
17   <p id="unico">Este es un párrafo único.</p>
18 </body>
19 </html>
20
```

```
1  /* Selector de Elemento */
2  h1 {
3    color: blue; /* Cambia el color del texto de todos los <h1> a azul */
4    font-family: Arial, sans-serif; /* Cambia la fuente del texto de todos los <h1> */
5    padding: 10px; /* Añade un relleno de 10 pixeles */
6    margin-bottom: 20px; /* Añade un margen inferior de 20 pixeles */
7    border: 2px solid black; /* Añade un borde sólido de 2 pixeles de color negro */
8    background-color: lightgray; /* Cambia el color de fondo a gris claro */
9  }
10
11 /* Selector de Clase */
12 .importante {
13   color: red; /* Cambia el color del texto de los elementos con la clase "importante" a rojo */
14   font-family: "Courier New", Courier, monospace; /* Cambia la fuente del texto de los elementos con la clase "importante" */
15   padding: 10px; /* Añade un relleno de 10 pixeles */
16   margin: 10px 0; /* Añade un margen superior e inferior de 10 pixeles */
17   border: 1px dashed black; /* Añade un borde discontinuo de 1 pixel de color negro */
18   background-color: lightyellow; /* Cambia el color de fondo a amarillo claro */
19 }
20
21 /* Selector de Clase adicional */
22 .destacado {
23   font-size: 20px; /* Cambia el tamaño del texto de los elementos con la clase "destacado" */
24   font-weight: bold; /* Cambia el texto a negrita para los elementos con la clase "destacado" */
25   padding: 15px; /* Añade un relleno de 15 pixeles */
26   margin: 10px 0; /* Añade un margen superior e inferior de 10 pixeles */
27   border: 1px solid red; /* Añade un borde sólido de 1 pixel de color rojo */
28   background-color: lightpink; /* Cambia el color de fondo a rosa claro */
29 }
30
31 /* Selector de ID */
32 #unico {
33   color: green; /* Cambia el color del texto del elemento con el ID "unico" a verde */
34   font-family: "Times New Roman", Times, serif; /* Cambia la fuente del texto del elemento con el ID "unico" */
35   padding: 20px; /* Añade un relleno de 20 pixeles */
36   margin: 20px 0; /* Añade un margen superior e inferior de 20 pixeles */
37   border: 2px solid green; /* Añade un borde sólido de 2 pixeles de color verde */
38   background-color: lightgreen; /* Cambia el color de fondo a verde claro */
39 }
```

Box Model en CSS

El modelo de caja en CSS (Box Model) es un concepto fundamental que define cómo se dispone y se espacian los elementos en una página web. Cada elemento se representa como una caja rectangular que comprende margen, borde, relleno (padding) y contenido.



Box Model en CSS

Diagrama del Box Model

- **Contenido:** El área donde se muestra el texto "Este es un ejemplo del modelo de caja."
- **Relleno (Padding):** El espacio entre el contenido y el borde (10px).
- **Borde (Border):** El contorno alrededor del relleno (2px sólido negro).
- **Margen (Margin):** El espacio exterior alrededor del borde (20px).

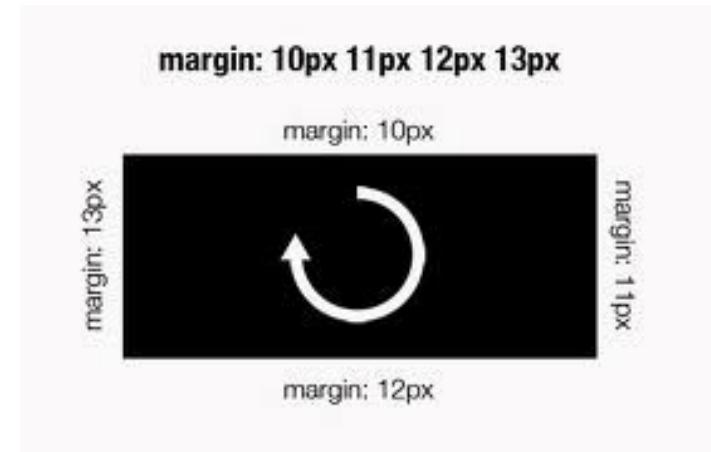
Propiedades CSS del Box Model

padding

- **Descripción:** Define el espacio entre el contenido y el borde.
- **Valores:** Longitudes en px, em, %, etc.

border

- **Descripción:** Define el borde alrededor del contenido y el relleno.
- **Valores:** Ancho (px), estilo (solid, dashed, etc.), y color.



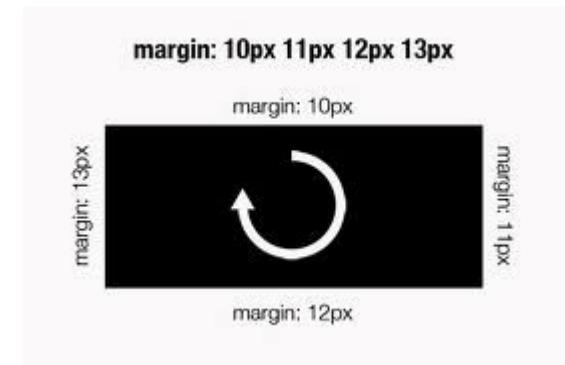
Box Model en CSS

margin

- **Descripción:** Define el espacio exterior alrededor del borde.
- **Valores:** Longitudes en px, em, %, etc.

width y height

- **Descripción:** Define el ancho y la altura del contenido.
- **Valores:** Longitudes en px, em, %, etc.



```
1 .box {  
2   width: 200px; /* Ancho del contenido */  
3   height: 100px; /* Altura del contenido */  
4   padding: 10px; /* Relleno */  
5   border: 2px solid black; /* Borde */  
6   margin: 20px; /* Margen */  
7   background-color: lightblue; /* Color de fondo del contenido */  
8 }
```

Tamaños

Píxeles (px): Unidad fija, ideal para precisión.

Ems (em): Unidad relativa al tamaño de la fuente del parente, útil para escalabilidad.

Rems (rem): Unidad relativa al tamaño de la fuente raíz, proporciona consistencia.

Porcentaje (%): Unidad relativa al tamaño del contenedor, ideal para diseños fluidos.

Vista de Ancho (vw) y Vista de Altura (vh): Unidad relativa a la ventana gráfica, útil para adaptabilidad.

Ejercicio Práctico 2

Aplica Box Model

```
1  /* Estilo para el contenedor principal */
2  .contenedor {
3      width: 80%; /* Ancho del contenedor */
4      margin: 20px auto; /* Margen superior e inferior de 20px, centrado horizontalmente */
5      padding: 10px; /* Relleno de 10px */
6      border: 2px solid #333; /* Borde sólido de 2px de color gris oscuro */
7      background-color: #f9f9f9; /* Color de fondo */
8  }
9
10 /* Estilo para los párrafos */
11 p {
12     margin: 15px 0; /* Margen superior e inferior de 15px */
13     padding: 10px; /* Relleno de 10px */
14     border: 1px solid #666; /* Borde sólido de 1px de color gris */
15     background-color: #e9e9e9; /* Color de fondo */
16 }
17
18 /* Estilo para los encabezados */
19 h1, h2 {
20     margin: 20px 0; /* Margen superior e inferior de 20px */
21     padding: 15px; /* Relleno de 15px */
22     border: 1px solid #000; /* Borde sólido de 1px de color negro */
23     background-color: #d9d9d9; /* Color de fondo */
24 }
```

The screenshot shows a web browser window with the following details:

- Address Bar:** localhost:8080/ejercicios/sesion3/ejercicio3.html
- Content Area:** A large gray box containing the following text:
 - Section Header:** **Aplicando el Box Model en CSS**
 - Text:** Este es un párrafo con margen, relleno, borde y color de fondo aplicados utilizando el modelo de caja.
 - Section Header:** **Subtítulo con Box Model**
 - Text:** Otro párrafo con estilos aplicados. Los márgenes y el relleno ayudan a separar el contenido y mejorar la presentación visual.

Extra CSS

Selectores avanzados

Selector Descendiente

- **Descripción:** Selecciona todos los elementos que son descendientes de un elemento específico.

```
● ● ●  
1 div p {  
2   color: blue; /* Todos los <p> dentro de <div> serán azules */  
3 }
```

Selector Hijo

- **Descripción:** Selecciona todos los elementos que son hijos directos de un elemento específico.

```
● ● ●  
1 ul > li {  
2   color: green; /* Todos los <li> hijos directos de <ul> serán verdes */  
3 }
```

Selectores avanzados (menos usados)

Selector de Hermano Adyacente

- **Descripción:** Selecciona el elemento que es inmediatamente precedido por un elemento específico.

```
● ● ●  
1 h1 + p {  
2   color: red; /* El primer <p> inmediatamente después de <h1> será rojo */  
3 }
```

Selector de Hermanos Generales

- **Descripción:** Selecciona todos los elementos que son precedidos por un elemento específico.

```
● ● ●  
1 h1 ~ p {  
2   color: orange; /* Todos los <p> precedidos por <h1> serán naranjas */  
3 }
```

Selectores avanzados (menos usados)

Selector de Atributos

- **Descripción:** Selecciona elementos que tienen un atributo específico.

```
● ● ●  
1 input[type="text"] {  
2   border: 1px solid black; /* Todos los <input> de tipo texto tendrán un borde negro */  
3 }
```

Selector de Pseudo-clase: `nth-child`

- **Descripción:** Selecciona el elemento hijo que coincide con un patrón específico.

```
● ● ●  
1 li:nth-child(2) {  
2   color: purple; /* El segundo <li> en cada <ul> será púrpura */  
3 }
```

Ejercicio Práctico 3

Aplicar selectores avanzados a una página HTML.

Este es un párrafo dentro de un div.

Otro párrafo dentro de un div.

- Elemento de lista 1
- Elemento de lista 2
- Elemento de lista 3

Encabezado 1

Este párrafo sigue al encabezado 1 y es rojo. (?)

Este párrafo sigue al encabezado 1 y es naranja.

Texto	Contraseña
-------	------------

- Elemento de lista 1
- Elemento de lista 2 (púrpura)
- Elemento de lista 3

Posicionamiento en CSS

Tipo	Descripción
Static	Posicionamiento por defecto. Los elementos se colocan según el flujo normal del documento.
Relative	Posiciona el elemento relativo a su posición original en el flujo del documento.
Absolute	Posiciona el elemento relativo a su contenedor posicionado más cercano (no estático).
Fixed	Posiciona el elemento relativo a la ventana del navegador. El elemento no se desplaza con el scroll.
Sticky	Posiciona el elemento de forma relativa hasta que se cruza un umbral, momento en el cual se fija en esa posición.

/sesion4

```
● ● ●
1 /* Estático: posición por defecto, sigue el flujo normal del documento */
2 .static {
3   position: static;
4   background-color: lightgray;
5 }
6
7 /* Relativo: se posiciona relativo a su posición original */
8 .relative {
9   position: relative;
10  top: 20px; /* Desplaza el elemento 20px hacia abajo desde su posición original */
11  left: 10px; /* Desplaza el elemento 10px hacia la derecha desde su posición original */
12  background-color: lightcoral;
13 }
14
15 /* Absoluto: se posiciona relativo al contenedor posicionado más cercano */
16 .absolute {
17   position: absolute;
18   top: 1000px; /* Desplaza el elemento 50px desde la parte superior del contenedor posicionado */
19   right: 20px; /* Desplaza el elemento 20px desde la parte derecha del contenedor posicionado */
20   background-color: lightgreen;
21 }
22
23 /* Fijo: se posiciona relativo a la ventana del navegador y no se desplaza con el scroll */
24 .fixed {
25   position: fixed;
26   top: 0;
27   left: 0;
28   width: 100%; /* Ancho del 100% de la ventana */
29   background-color: yellow;
30 }
31
32 /* Pegajoso: se comporta como relativo hasta que se cruza un umbral, luego se fija */
33 .sticky {
34   position: -webkit-sticky; /* Soporte para navegadores basados en WebKit */
35   position: sticky;
36   top: 0; /* Se fija en la parte superior cuando se cruza ese punto */
37   background-color: lightblue;
38 }
39
40 /* Contenedor: utilizado para contener los elementos absolutos y relativos */
41 .container {
42   position: relative; /* Establece un contexto de posicionamiento para elementos hijos absolutos */
43   height: 200px;
44   bac
```

Ejercicio Práctico 4

Practica con los distintos posicionamientos

Elemento Fixed (Fijo)

Elemento Static (Estático)

Elemento Relative (Relativo)

Elemento Sticky (Pegajoso)

↓
Scroll down...

CSS Grid

CSS Grid Layout es un sistema de diseño basado en una cuadrícula bidimensional.

Proporciona una manera poderosa y flexible de crear diseños complejos de manera sencilla y eficiente, permitiendo al desarrollador definir tanto filas como columnas.

A diferencia de otros sistemas de diseño, CSS Grid puede controlar tanto el diseño horizontal como vertical de una página al mismo tiempo.

https://www.w3schools.com/css/css_grid.asp

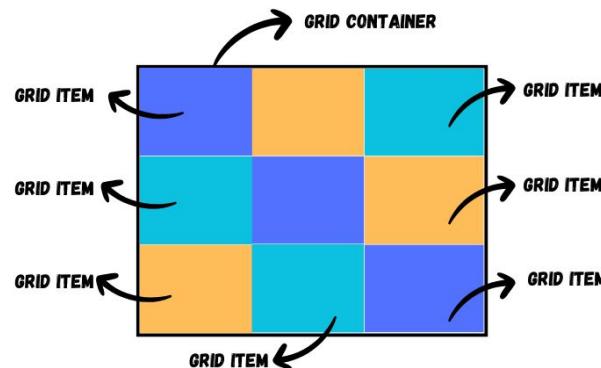
Características principales

Contenedor de Cuadrícula (Grid Container)

- El elemento padre que define el contexto de cuadrícula y contiene los elementos de la cuadrícula (grid items).
- Se establece utilizando `display: grid;`.

Elementos de la Cuadrícula (Grid Items)

- Los elementos hijos directos del contenedor de cuadrícula.
- Pueden ser posicionados y dimensionados de manera precisa dentro de la cuadrícula.



Grid Container

grid-template-columns y grid-template-rows

- Definen el número y tamaño de las columnas y filas de la cuadrícula.

gap

- Define el espacio entre las filas y columnas de la cuadrícula.



```
1 .grid-container {  
2   grid-template-columns: 1fr 2fr; /* Define dos columnas, la primera ocupa 1 fracción y la segunda 2 fracciones */  
3   grid-template-rows: 100px auto; /* Define dos filas, la primera de 100px y la segunda con tamaño automático */  
4   gap: 10px; /* Espacio de 10px entre filas y columnas */  
5 }
```

Grid Items

grid-column

- Define la posición y el tamaño de una columna de un elemento de la cuadrícula.

grid-row

- Define la posición y el tamaño de una fila de un elemento de la cuadrícula.

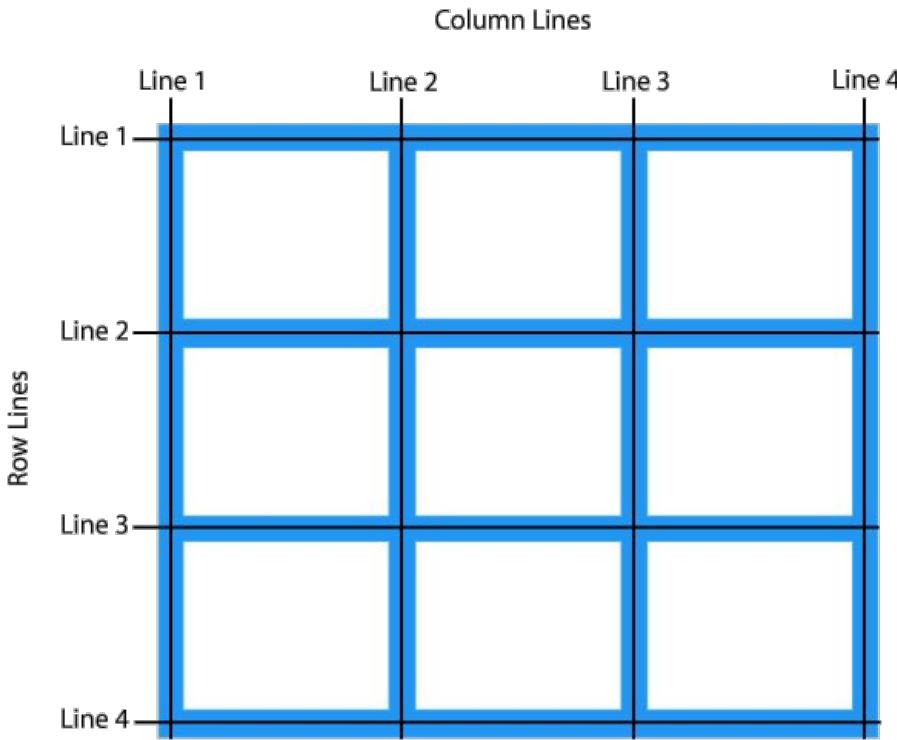


```
1 .grid-item {  
2   grid-column: 1 / 3; /* Abarca desde la columna 1 hasta la 3 */  
3 }
```



```
1 .grid-item {  
2   grid-row: 2 / 4; /* Abarca desde la fila 2 hasta la 4 */  
3 }
```

OJO! El final y el inicio son las líneas



Ejercicio Práctico 5

Aplicar CSS Grid con varios elementos

Ejemplo de solución

Elemento 2

Elemento 1

Elemento 3

Elemento 4

Otras consideraciones

Existen otras propiedades para CSS que sirven para organizar contenido como **CSS Flexbox**.

CSS Grid más moderno, customizable y avanzado.

Con **JavaScript** y **bibliotecas externas** (como bootstrap) también puede organizarse contenido.

Flexbox

One-dimensional layout



Grid

Multi-dimensional layout



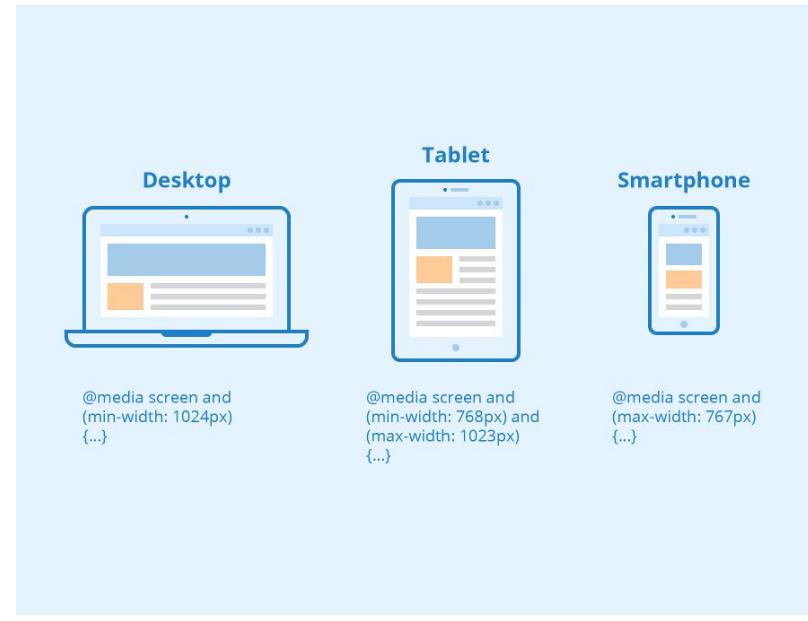
**HTML Semántico e
Introducción a Bootstrap**

Sesión 4

Media Queries - Tamaños de pantalla

Media Queries son una característica de CSS3 que permiten aplicar estilos CSS basados en las características del dispositivo o la ventana del navegador, como el ancho, la altura, la resolución, la orientación y más.

Esto permite crear diseños responsivos que se adaptan a diferentes tamaños de pantalla y dispositivos, mejorando la experiencia del usuario.



Media Queries

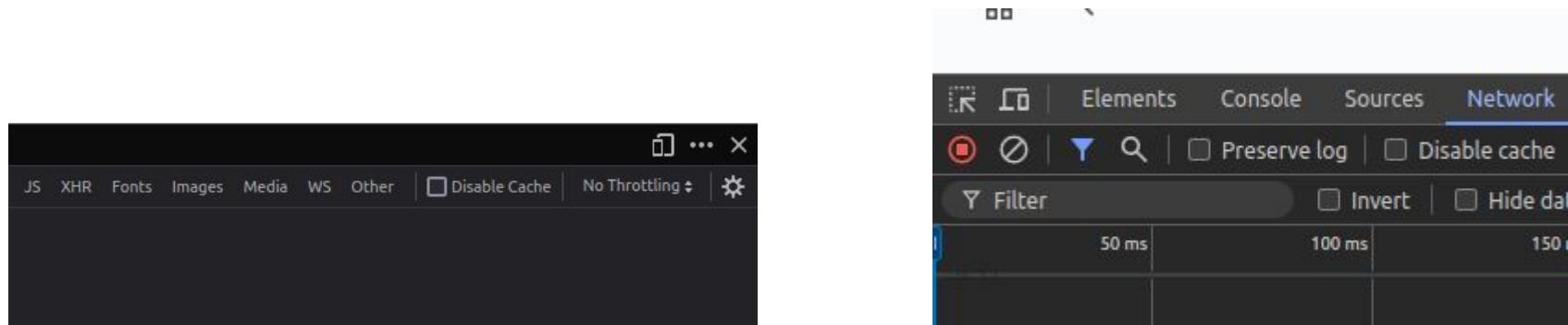
Media Query Básica: Aplica estilos si el ancho de la pantalla es menor o igual a 600px.

Media Query con Múltiples Condiciones: Aplica estilos si se cumplen múltiples condiciones, por ejemplo, si el ancho de la pantalla está entre 601px y 1024px.

Media Query para Orientación: Aplica estilos basados en la orientación del dispositivo, ya sea en modo retrato (portrait) o paisaje (landscape).

```
● ● ●
1 @media (max-width: 600px) {
2   .container {
3     background-color: lightblue; /* Fondo azul claro en pantallas pequeñas */
4   }
5 }
6
7 @media (min-width: 601px) and (max-width: 1024px) {
8   .container {
9     background-color: lightgreen; /* Fondo verde claro en pantallas medianas */
10  }
11 }
12
13 @media (min-width: 1025px) {
14   .container {
15     background-color: lightcoral; /* Fondo coral claro en pantallas grandes */
16   }
17 }
18
19 @media (orientation: landscape) {
20   .container {
21     border-color: blue; /* Cambia el color del borde a azul en modo paisaje */
22   }
23 }
```

Modo pantalla pequeña navegador



Ejercicio Práctico 0

(ejercicio 6 sesión 3)

Haz que los elementos cambien según el tamaño de la pantalla

Redimensiona la ventana para ver los cambios en el fondo.

Status	Method	Domain	File	Initiator	Type	Transferred	Size	0 ms	80 ms	160 ms	240 ms
200	GET	localhost:8080	ejercicio4.html	document	html	1.38 kB	1.07 kB	1 ms			
404	GET	localhost:8080	favicon.ico	Favicon Loader sys.mjs:175 (img)	x-icon	cached	0 B	0 ms			

HTML Semántico

HTML Semántico se refiere al uso de etiquetas HTML que tienen un significado claro y descriptivo sobre el contenido que encierran. Lo vimos en las primeras sesiones!

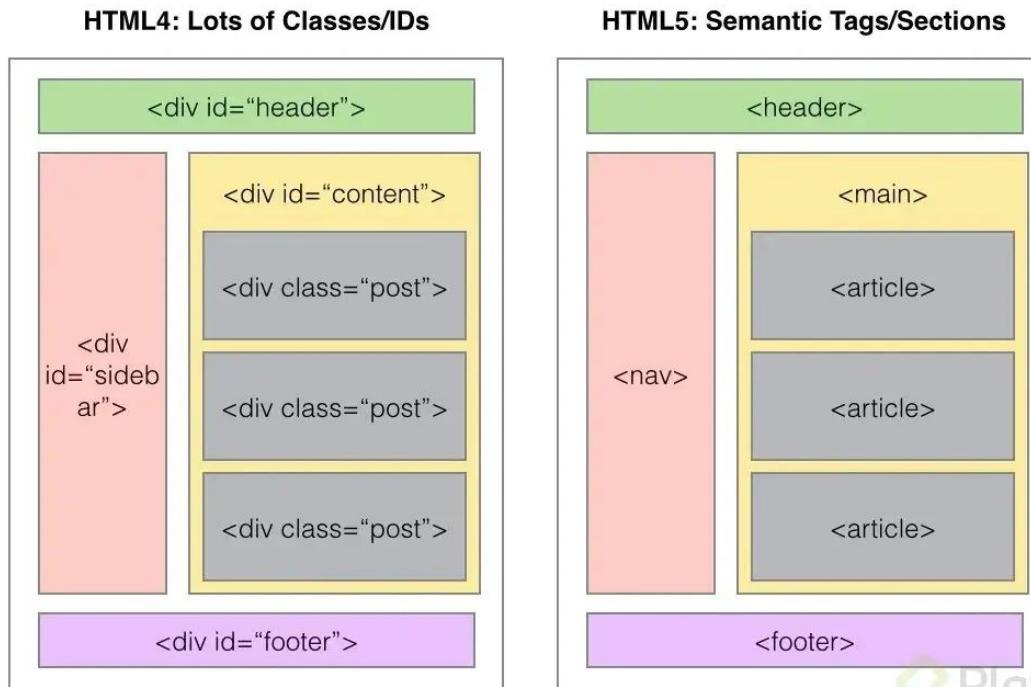
Estas etiquetas no solo describen la apariencia del contenido, sino también su propósito y estructura, lo que mejora la comprensión tanto para los desarrolladores como para los navegadores y motores de búsqueda.

Por qué?

1. **Accesibilidad:**
 - Mejorar la accesibilidad del contenido web para personas con discapacidades, permitiendo a los dispositivos de asistencia, como los lectores de pantalla, interpretar mejor la estructura y el propósito del contenido.
2. **SEO (Search Engine Optimization):**
 - Ayudar a los motores de búsqueda a entender y indexar mejor el contenido de las páginas web, lo que puede mejorar el ranking en los resultados de búsqueda.
3. **Mantenibilidad y Legibilidad:**
 - Facilitar la lectura y el mantenimiento del código HTML, permitiendo a los desarrolladores entender rápidamente la estructura y el propósito del contenido sin tener que interpretar estilos o estructuras complejas.

Etiquetas descriptivas (pero es lo mismo)

HTML4 vs HTML5 Page Structure on a Blog



Ejemplo HTML semántico

Bibliotecas externas

Ya sabemos HTML y CSS, pero por qué no se parecen las páginas que hacemos a las de la web?

Razón 2: le falta comportamiento = le falta JS

Hacer todo el CSS de una página requiere mucho esfuerzo.



Una **Biblioteca externa** es un conjunto de archivos predefinidos (generalmente CSS y JS)

Soluciones para tareas comunes en el desarrollo web.

Ahorro de Tiempo + Consistencia + Comunidad y soporte

Como cargar bibliotecas

Etiquetas “link” y “script”

Dentro de la cabeza del código HTML

```
1 <head>
2   <link rel="stylesheet" href="https://example.com/library.css">
3   <script src="https://example.com/library.js"></script>
4 </head>
```

Urls externas o internas ojo!

CSS es igual que CSS externo!

Introducción a Bootstrap



Bootstrap es una biblioteca de código abierto para el desarrollo de sitios y aplicaciones web responsivas.

Fue creado por el equipo de Twitter y se ha convertido en una de las bibliotecas CSS y JS más populares en el desarrollo web.

Bootstrap proporciona un conjunto de herramientas predefinidas para crear interfaces de usuario atractivas y funcionales de manera rápida y sencilla.

<https://getbootstrap.com/>

Instalando Bootstrap

Hay muchas maneras de instalar Bootstrap dependiendo de nuestro tipo de aplicación.

Nosotros estamos trabajando con HTML **estático = CDN** es nuestra mejor solución! (o **bajar los archivos en local!!!**)

The screenshot shows a dark-themed section of the Bootstrap documentation. At the top left is a globe icon. Below it, the title "Include via CDN" is displayed in white. The text explains that if you only need to include Bootstrap's compiled CSS or JS, you can use jsDelivr. It provides links to a "quick start" and "examples". It also mentions that you can choose to include Popper and our JS separately. Two code snippets are shown in a light-colored box:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/...>
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/...>
```

Ejercicio Práctico 1

Añade Bootstrap vía CDN a uno de los ejercicios anteriores y comprueba si hay algún cambio

Sesión 2 Ejercicio 1

Mi Página Estructurada

- [Inicio](#)
- [Sobre](#)
- [Contacto](#)

Inicio

Bienvenido a mi página web. Aquí encontrarás información sobre diversos temas.

Sobre

Esta es una página web de ejemplo creada para aprender sobre la estructura de HTML.

Contacto

Puedes contactarme a través del correo: ejemplo@correo.com

Sección 1

Contenido de la sección 1.

Mi Página Web

- [Inicio](#)
- [Acerca de](#)

© 2024 Mi Página Web. Todos los derechos reservados.

© 2024 Mi Página Web. Todos los derechos reservados.

Mi Página Estructurada

- [Inicio](#)
- [Sobre](#)
- [Contacto](#)

Inicio

Bienvenido a mi página web. Aquí encontrarás información sobre diversos temas.

Sobre

Esta es una página web de ejemplo creada para aprender sobre la estructura de HTML.

Contacto

Puedes contactarme a través del correo: ejemplo@correo.com

Sección 1

Contenido de la sección 1.

Mi Página Web

- [Inicio](#)
- [Acerca de](#)

© 2024 Mi Página Web. Todos los derechos reservados.

Grid system / Cuadrícula

El sistema de cuadrícula de Bootstrap es una de sus características **más poderosas y versátiles**. Facilita la creación de layouts **responsivos y adaptativos** utilizando una serie de contenedores, **filas y columnas** para alinear y distribuir contenido de manera efectiva.

Contenedores (Containers):

- Los contenedores son elementos de diseño que centran y proporcionan un margen horizontal al contenido dentro de un layout responsivo fijo o fluido.

Filas (Rows):

- Las filas se utilizan para crear un grupo horizontal de columnas.

Columnas (Columns):

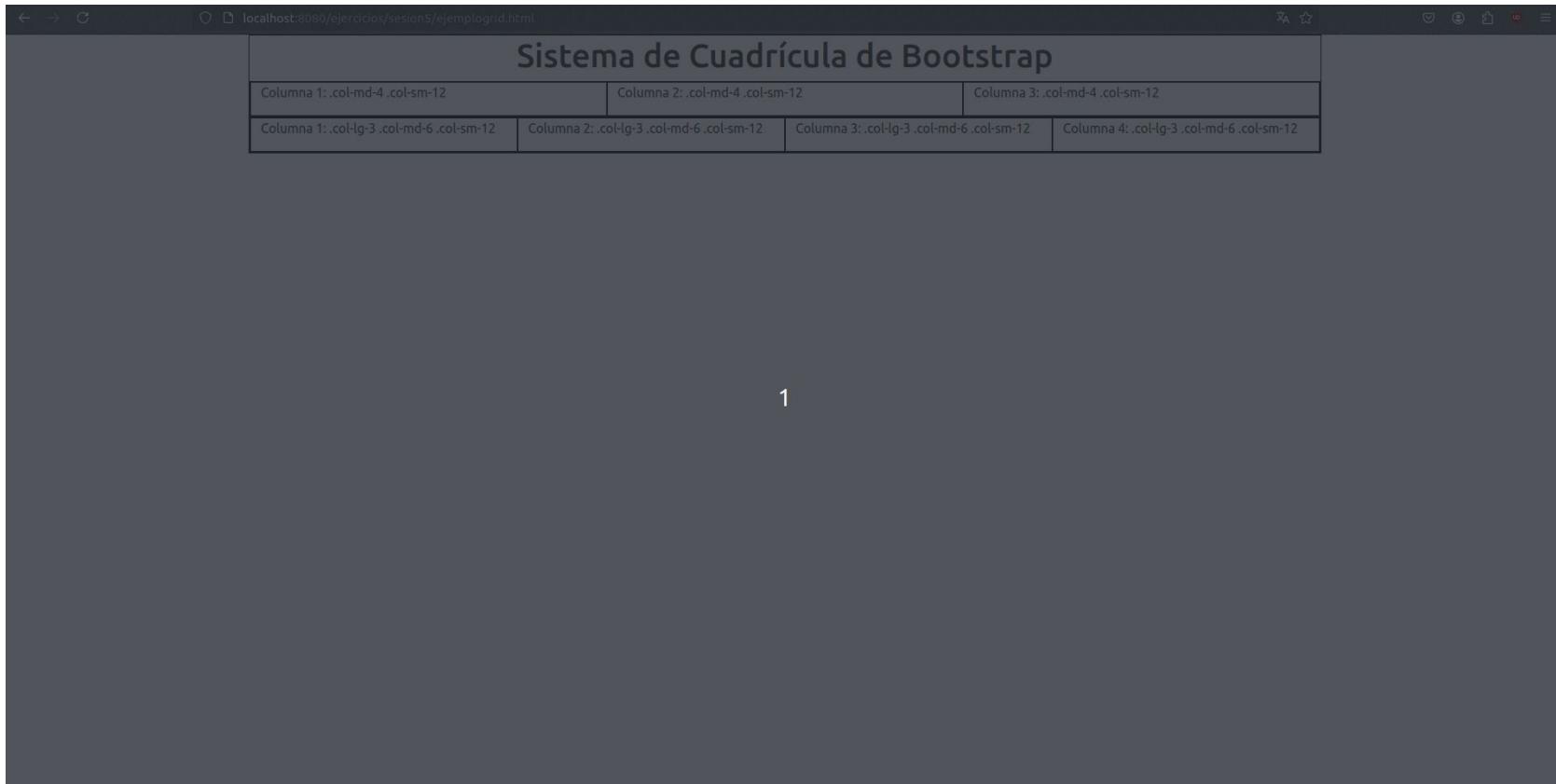
- Las columnas crean una estructura de diseño en 12 partes iguales dentro de una fila. Puedes combinar columnas para crear layouts más complejos.

Breakpoints Responsivos:

- Bootstrap incluye cinco breakpoints para crear layouts responsivos: `xs` (extra pequeño), `sm` (pequeño), `md` (mediano), `lg` (grande) y `xl` (extra grande).

<https://getbootstrap.com/docs/5.3/layout/grid/>

Ejemplos del Sistema de Grid



Componentes básicos de Bootstrap

Vamos a revisar los distintos componentes que tiene bootstrap!

- Navbar (<https://getbootstrap.com/docs/5.3/components/navbar/>)
- Carousel (<https://getbootstrap.com/docs/5.3/components/carousel/>)
- Buttons (<https://getbootstrap.com/docs/5.3/components/buttons/>)
- Tarjetas (<https://getbootstrap.com/docs/5.3/components/card/>)
- Alertas (<https://getbootstrap.com/docs/5.3/components/alerts/>)
- Formularios (<https://getbootstrap.com/docs/5.3/forms/overview/>)
- Acordeón (<https://getbootstrap.com/docs/5.3/components/accordion/>)
- Muchos más...

Hint: Copia y pega el contenido y luego modifica según necesidades!

Ejercicio Práctico 2

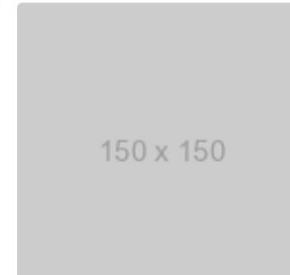
Crear un layout básico usando el sistema de Grid que incluya una barra de navegación, un carrusel de fotos y otro componente (a elección)



Título de la Tarjeta 1

Algun texto de ejemplo rápido para construir el título de la tarjeta y hacer relleno de la tarjeta.

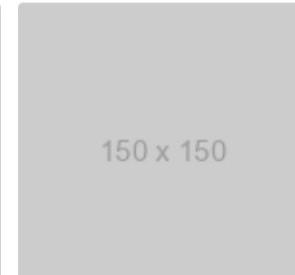
Ir a algún lugar



Título de la Tarjeta 2

Algun texto de ejemplo rápido para construir el título de la tarjeta y hacer relleno de la tarjeta.

Ir a algún lugar



Título de la Tarjeta 3

Algun texto de ejemplo rápido para construir el título de la tarjeta y hacer relleno de la tarjeta.

Ir a algún lugar

Bootstrap templates

Cuidado! Hay algunas páginas que cobran por las templates.

Importante: **LICENCIA**

A veces hay que descargar archivos CSS y JS extra para que funcionen

<https://github.com/awesome-bootstrap-org/awesome-bootstrap?tab=readme-ov-file#single-page-templates>

<https://startbootstrap.com/themes?showPro=false>

Yo las uso! <https://thesis-dlumbrer.gitlab.io>

Ejercicio Práctico 3

Copia una template de bootstrap y cambia su contenido HTML, modifica su CSS también

Página personal

<https://startbootstrap.com/theme/personal>

Descargar todo y cambiar poco a poco.

Despliegue

Otras bibliotecas

Sesión 5

¿Por Qué Necesitamos un Servidor?

Acceso a Recursos Externos

Al utilizar bibliotecas externas, el navegador necesita hacer solicitudes HTTP para obtener estos recursos. Esto no es posible cuando se accede a archivos localmente a través del protocolo `file://`.

Ya no nos vale abrir el archivo con el navegador!!!

CDN (Content Delivery Network) es una red de servidores distribuidos que entregan contenido web a los usuarios en función de su ubicación geográfica. Muchas bibliotecas proporcionan enlaces CDN para incluir fácilmente sus archivos CSS y JS en los proyectos.

Servidores HTTP

Sabemos hacer páginas web, ahora es hora de **compartir!**

Servidor HTTP -> Live Server. Queremos uno público y accesible.

Servicios de despliegue, muchos -> Wordpress,

Dominios

Gratuitos: GitHub, GitLab, servidor propio con un dominio gratis...

Repositorios GitHub

Un **repositorio** remoto (p.e un repositorio de GitHub) es un lugar de almacenamiento de datos (usualmente es código) en el servidor remoto.

Un repositorio **GIT** usa un sistema de **control de versiones**

Cada **COMMIT** es un punto de guardado

Vamos a:

1. Crear cuenta de GitHub si no la tenemos
2. Crear un repositorio
3. Añadir archivos al repositorio

IDE en GitHub

En la carpeta del repositorio pulsamos la tecla “.” o sustituyendo en la url “github.com” por “**github.dev**”

Visual studio code integrado en el navegador.

Vamos a:

1. Crear un archivo HTML
2. Modificar uno que hemos subido antes

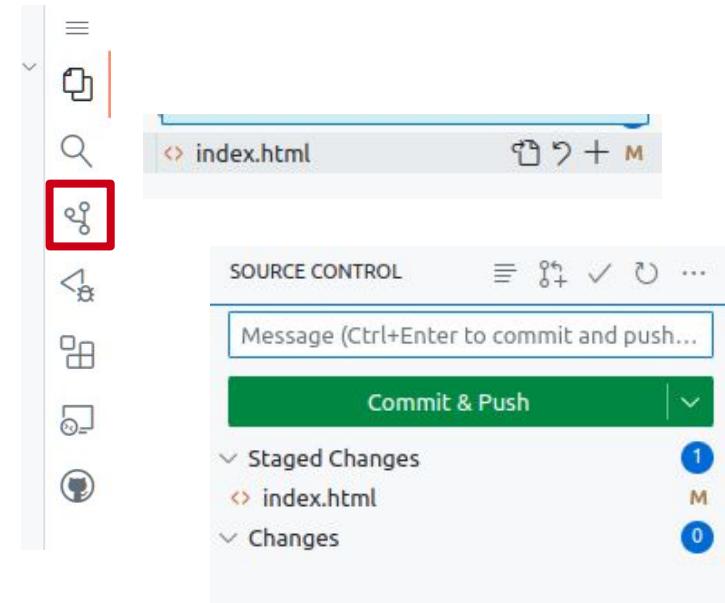
Commit ~ Guardar

Cada vez que hacemos un **Commit** hacemos una “foto” al estado actual del código.

Necesitamos hacerlo para actualizar el **repositorio remoto**.

Vamos a:

1. Icono de “commit”
2. “stage” cambios que queremos actualizar
3. Escribir mensaje
4. Commits and push
5. Comprobar en la página del repositorio que el commit está!



GitHub Pages

Activamos desde el apartado de ajustes de la aplicación.

1. Settings
2. Menú de la Izquierda -> Pages
3. Deploy from a branch -> main

Luego en pestaña **Actions** vemos el proceso de como se está construyendo nuestra web.

Punto verde = OK

Punto rojo = FALLO

Luego nuestros htmls estarán desplegados bajo esta URL:

`https://{usuario}.github.io/{repositorio}/{archivohtml}`

EL CONFIGURAR LAS “PAGES” SE HACE SOLO UNA VEZ POR REPOSITORIO!

Nuevo repositorio que queremos tener desplegado = otra vez estos pasos

GitHub Pages

**CADA VEZ QUE HAGAMOS UN COMMIT LA PÁGINA SE ACTUALIZARÁ
AUTOMÁTICAMENTE**

(PUEDE TARDAR DEPENDIENDO DE LO QUE TARDEN LOS BOTS DE
DESPLIEGUE)

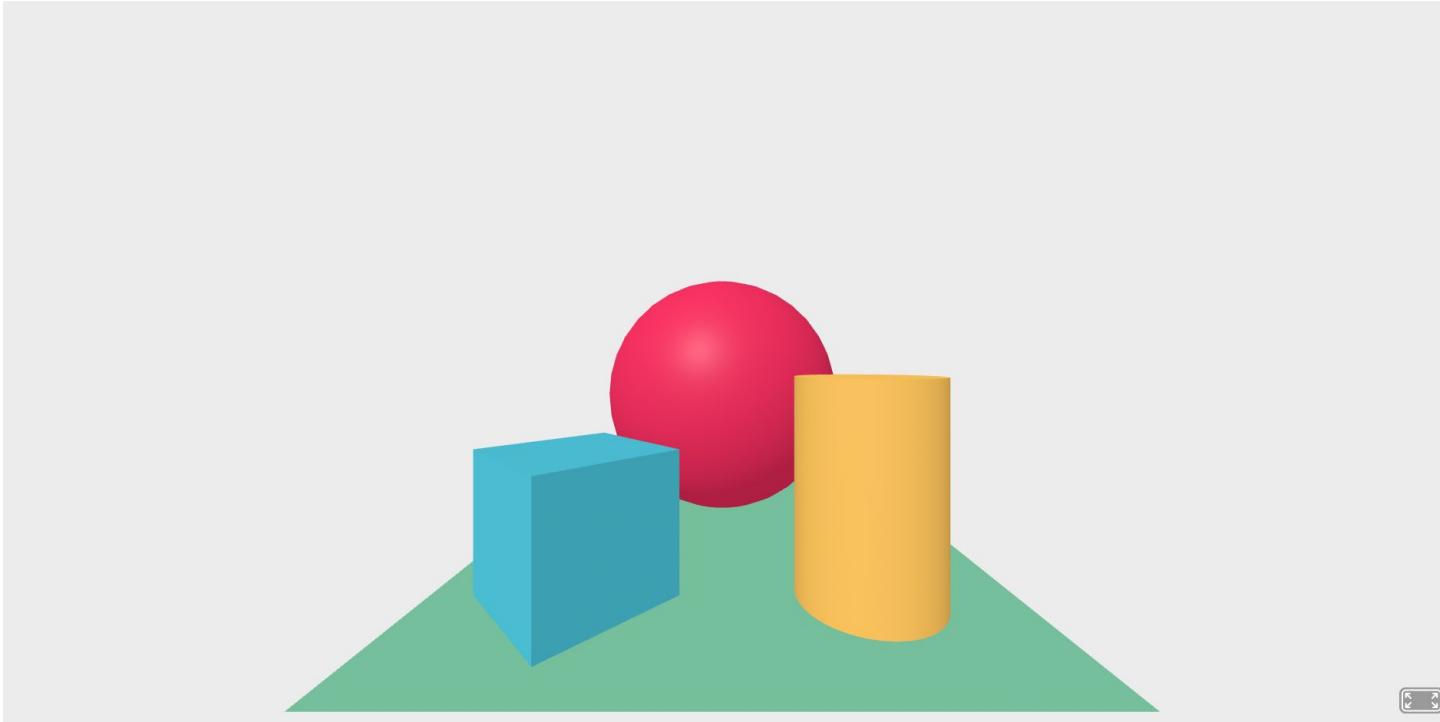
Ejercicio Práctico 1

Despliega una página en GitHub Pages

Ejercicio Práctico 2

Despliega OTRA página en GitHub Pages
y enlazarla a la anterior

A-Frame (escenas en 3D/VR)



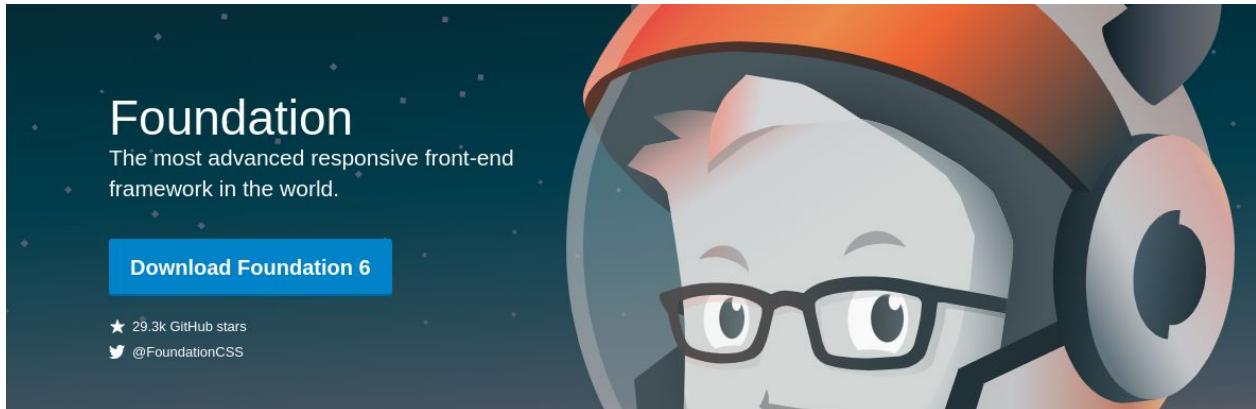
<https://aframe.io/>

Foundation

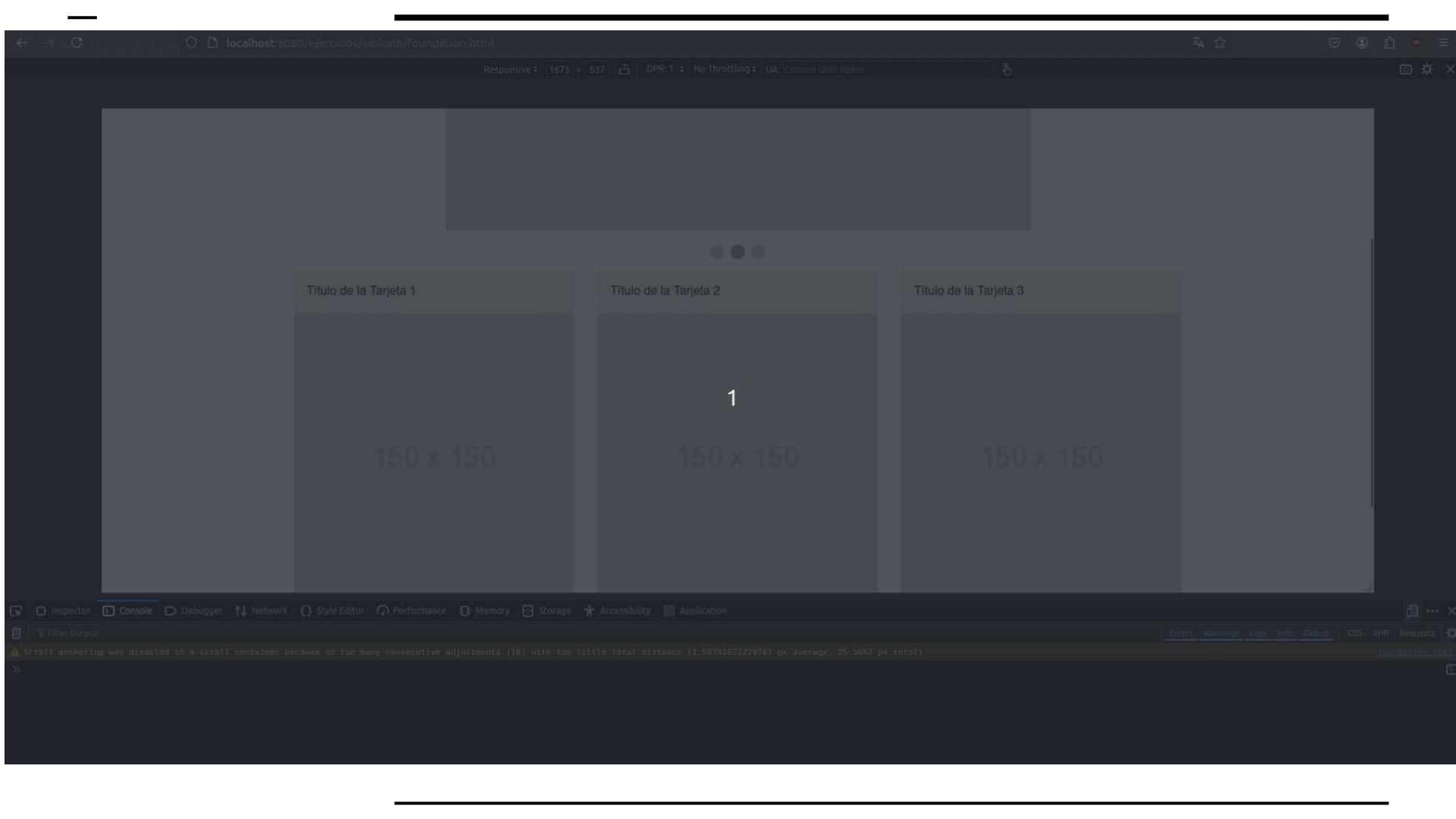
Foundation es una biblioteca front-end de código abierto desarrollada por ZURB para diseñar y construir sitios web y aplicaciones responsivas.

Requiere de JQuery, otra biblioteca (pero esta es solo JS y es para JS).

Se parece a bootstrap, pero no es tan famoso.



<https://get.foundation/index.html>



Foundation vs Bootstrap

Foundation y **Bootstrap** son ambas bibliotecas front-end populares para el desarrollo de interfaces web responsivas, pero tienen enfoques y características diferentes.

Foundation ofrece mayor flexibilidad y personalización avanzada, siendo ideal para proyectos que requieren diseños únicos y detallados. Está completamente basado en **Sass (lenguaje de hoja de estilos en cascada)**, lo que facilita la personalización.

Bootstrap, por otro lado, es más estructurado y proporciona un conjunto de estilos predefinidos que facilitan el desarrollo rápido, aunque puede ser menos adaptable a diseños personalizados sin sobrescribir CSS. Bootstrap es más popular y tiene una **comunidad más grande**, lo que facilita encontrar recursos y soporte.

Bulma

Bulma es una biblioteca de **CSS** moderna basada en Flexbox. Esta es s



Características Principales de Bulma

1. **Flexbox por Defecto:**
2. **Componentes Modulares** (componentes predefinidos, como botones, formularios, tarjetas, menús de navegación)
3. **Sin Dependencias de JavaScript:**
4. **Fácil de Aprender y Usar**

Mi Sitio Inicio Características Precios Contacto

Bienvenido a Mi Sitio

Un ejemplo simple usando Bulma

150 x 150

1
150 x 150

150 x 150

Inspector Console Debugger Network Style Editor Performance Memory Storage Accessibility Application

Filter Output

GET http://localhost:8080/favicon.ico

Errors Warnings Logs Info Debug CSS XHR Requests

[HTTP/1.1] 404 Not Found 2ms

Bulma vs Bootstrap

Bulma está completamente basado en Flexbox y no depende de JavaScript, lo que la hace ligera y fácil de integrar.

Bootstrap ofrece una combinación de Flexbox y otras técnicas de diseño y depende de JavaScript para muchos de sus componentes, proporcionando más funcionalidades pero con mayor complejidad y peso.

Tailwind CSS

Tailwind CSS es una biblioteca de utilidades CSS de bajo nivel que permite a los desarrolladores crear rápidamente interfaces de usuario personalizadas sin tener que escribir CSS personalizado desde cero.

Características Principales de Tailwind CSS

1. Utilidades de Bajo Nivel
2. Alta Personalización
3. Diseño Responsivo
4. Reutilización de Estilos
5. Sin Dependencias de JavaScript



<https://tailwindcss.com/>

Bienvenido a Mi Sitio

Un ejemplo simple usando Tailwind CSS

150 x 150

Título de la Tarjeta 1

Este es un ejemplo de tarjeta usando Tailwind CSS. Tailwind facilita la creación de interfaces modernas y responsivas.

150 x 150

Título de la Tarjeta 2

Este es un ejemplo de tarjeta usando Tailwind CSS. Tailwind facilita la creación de interfaces modernas y responsivas.

150 x 150

Título de la Tarjeta 3

Este es un ejemplo de tarjeta usando Tailwind CSS. Tailwind facilita la creación de interfaces modernas y responsivas.

Tailwind CSS vs Bootstrap

Tailwind CSS se centra en proporcionar utilidades de bajo nivel que permiten un alto grado de personalización directamente en el HTML.

Bootstrap ofrece un conjunto de componentes predefinidos y estilos estructurados que facilitan el desarrollo rápido.

Tailwind CSS es más flexible y adecuado para proyectos que requieren un diseño único y altamente personalizable, mientras que Bootstrap es ideal para proyectos que buscan una implementación rápida y consistente de componentes estándar.

Otras bibliotecas

Hemos visto el potencial de **Bootstrap**

Hay otras que hacen cosas “parecidas” o “mejores” (depende de tu gusto):

- **Aframe:** Crear escenas en 3D/XR.
- **Foundation:** Otra biblioteca popular para diseño responsivo.
- **Bulma:** Biblioteca basada en Flexbox para diseños simples y efectivos.
- **Tailwind CSS:** Biblioteca de utilidades CSS para diseños personalizados.
- <https://github.com/troxler/awesome-css-frameworks>

Pasos para usarla:

1. Leer documentación
2. Identificar CDN/Archivos
3. Insertar en nuestro HTML

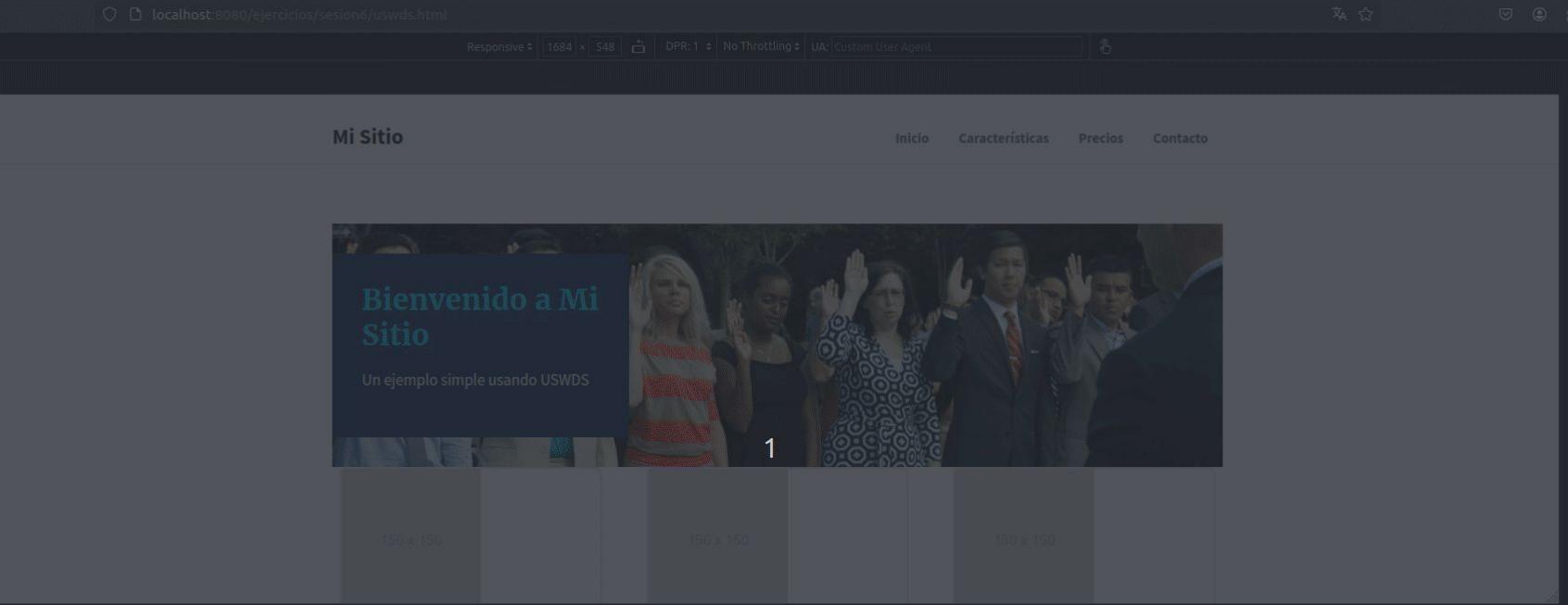
U.S. Web Design System

U.S. Web Design System (USWDS) es una biblioteca de diseño de código abierto creada por el Gobierno de los Estados Unidos para ayudar a los equipos a diseñar y construir sitios web y aplicaciones accesibles, consistentes y fáciles de usar.

The screenshot shows the homepage of the US Web Design System (USWDS). At the top, there's a dark header with the title "U.S. Web Design System (USWDS)" and a search bar with a magnifying glass icon. Below the header, a navigation menu includes links for "How to use USWDS", "Design principles", "Components", "Patterns", "Design tokens", "Utilities", "Templates", and "About". The main content area features a large graphic of various mobile devices and web interfaces. To the left, text reads "A design system for the federal government. We make it easier to build accessible, mobile-friendly government websites." Below this are two buttons: "Introducing USWDS 3.0" (red) and "Migrating to USWDS 3.0" (blue).

OPEN SOURCE!!

<https://designsystem.digital.gov/>



Search HTML

```
<!DOCTYPE html>
<html lang="es"> 事件 scroll
  <head> 事件 contextmenu
    <meta>
    <title>Mi Sitio</title>
  </head>
  <body> 事件 overflow
    <div>Barra de Navegación</div>
    <header class="usa-header usa-header--basic"></header>
    <div>Contenedor Principal</div>
    <main class="usa-section"></main>
    <!-- Incluir las dependencias de JavaScript de USWDS -->
    <script src="https://cdnjs.cloudflare.com/ajax/libs/uswds/2.11.2/js/uswds.min.js"></script>
  </body>
</html>
```

Inspector

Console

Debugger

Network

Style Editor

Performance

Memory

Storage

Accessibility

Application

Layout

Computed

Changes

Compatibility

Filter Styles

element ID ()

body EL (body.scss:1)

color: #1b1b1b;
background-color: #ffff;
overflow-x: hidden;

body EL (normalize.scss:123)

margin: 0;

1. ::after, ::before EL (sizing.scss:2)

box-sizing: inherit;

**El único límite para el éxito es la
imaginación.**

Ejercicio Práctico 3

Añade otra biblioteca y úsala

Presentación de Proyectos
Revisión Final

Sesión 6

Presentación de Proyectos

2 min

Comentarios y Sugerencias

Preguntas

Sugerencias

¿Qué os ha interesado más?

¿Habéis echado en falta algo?

Otras preguntas

JavaScript

Necesario nociones de **programación**, básicamente para la Manipulación del DOM (Document Object Model).

- El DOM representa la estructura del documento HTML como una jerarquía de nodos.
- JavaScript permite acceder y manipular estos nodos para cambiar dinámicamente el contenido y la estructura de la página.
- Etiqueta <script>
- Externo o Interno o en linea



JavaScript

```
● ● ●  
1 <h1 id="titulo">Hola Mundo</h1>  
2 <button onclick="cambiarTitulo()">Haz clic aquí</button>  
3  
4 <script>  
5   function cambiarTitulo() {  
6     document.getElementById('titulo').innerText = '¡Título Cambiado!';  
7   }  
8 </script>
```

Hola Mundo

Haz clic aquí

Traer datos de sitios externos - API

- Con JavaScript también se pueden hacer peticiones a sitios externos y traer datos para luego usarlos/representarlos.
- Estos sitios externos si permiten la petición de datos se denominan **APIs**

Traer datos de sitios externos - API

```
1 <h1>Datos del Usuario</h1>
2 <button onclick="obtenerDatos()">Obtener Datos</button>
3 <div id="datos"></div>
4
5 <script>
6   function obtenerDatos() {
7     fetch('https://jsonplaceholder.typicode.com/users/1')
8       .then(response => response.json())
9       .then(data => {
10         document.getElementById('datos').innerText = `Nombre: ${data.name}, Email: ${data.email}`;
11       });
12     }
13 </script>
```

Datos del Usuario

Obtener Datos

Para cerrar

Puntos clave del curso.

- Fundamentos de **HTML** y **CSS**.
- Uso de bibliotecas como **Bootstrap**, Tailwind, Foundation, Bulma, y USWDS.
- Creación de proyectos web **responsivos** y **accesibles**.

Próximos pasos en el aprendizaje:

- Profundizar en JavaScript y otros lenguajes de programación web.
- Explorar frameworks front-end como React, Angular, o Vue.js.
- Continuar practicando y creando proyectos web.

HTML Pro

Extra

Domina HTML en el IDE (VSCode)

1. Escritura automática de cabeceras: ! + Intro
2. Escritura de div más una clase: .hola + Intro = <div class="hola"></div>
3. Crea items y repitelos X veces:
 - a. circle.miclase*3 + Intro
 - b. <circle class="miclase"><circle>
 - c. <circle class="miclase"><circle>
 - d. <circle class="miclase"><circle>
4. Crea items dentro de items:
 - a. cicrle.miclase>.contenedor+button
 - b. <circle class="miclase"><div class="contenedor"></div><button></button><circle>

Aprende HTML

Iniciación para principiantes

David Moreno Lumbrales
david.morenolu@urjc.es

<https://dlumbrer.gitlab.io/curso-html>

CTIF Madrid-Sur