



José Corrochano Pardo

ÍNDICE

1. FUNDAMENTOS	2
1.1 CONFIGURANDO EL ENTORNO DE TRABAJO	3
1.2 PRIMEROS PASOS CON VSCODE	3
1.3 EXTENSIONES Y TEMAS	4
1.4 ¿QUÉ ES HTML?	5
1.5 ETIQUETAS HTML	5
1.6 ESTRUCTURA DE UN DOCUMENTO HTML	6
1.7 SNIPPETS, EMMET Y MENÚ CONTEXTUAL	8
1.8 COMENTARIOS	8
1.9 HOLA MUNDO	9
2. TEXTOS	10
2.1 TÍTULOS Y PÁRRAFOS	11
2.2 LISTAS	13
2.3 TABLAS	14
3. MULTIMEDIA	18
3.1 CÓMO CONSTRUIR RUTAS	19
3.2 ENLACES	20
3.3 IMÁGENES	21
3.4 VÍDEOS	22
3.5 AUDIO	23
3.6 IFRAME	24
4. FORMULARIOS	25
4.1 INPUTS DE TEXTO	27
4.2 SELECTS	29
4.3 RADIOBUTTONS	30
4.4 CHECKBOX	31
4.5 RANGOS	31
4.6 TEXTAREA	31
4.7 VALIDACIONES	32
4.8 ENVÍOS	33
4.9 FIELDSET Y LEGEND	33
5. SEMÁNTICA	34
5.1 CABECERA, PRINCIPAL Y PIE	35
5.2 NAVEGACIÓN	36
5.3 SECCIONES, ARTÍCULOS Y AUXILIARES	37
5.4 DIVISORES	38



FUNDAMENTOS

CONFIGURANDO EL ENTORNO DE TRABAJO

Para aprender a usar HTML usaremos “Visual Studio Code”. Para descargarlo, usaremos el siguiente enlace:

DESCARGAR VISUAL STUDIO CODE

<https://code.visualstudio.com/>

Visual Studio Code es un IDE. Esto es un Entorno de desarrollo Integrado, es decir, es una herramienta de Software que nos va a proporcionar un entorno de programación completo para que podamos desarrollar nuestro código sin ningún problema. En esencia, será el editor de texto que nos va a incorporar todas las herramientas necesarias para que podamos trabajar. Una vez instalado nuestro IDE, lo abrimos para empezar a trabajar.

PRIMEROS PASOS CON VSCODE

VSCode es totalmente personalizable. Uno de los primeros cambios que vamos a realizar es ampliar el tamaño, para ello utilizamos “CONTROL +”, y si queremos reducirlo, “CONTROL -”. Lo primero que vamos a utilizar es el menú de la izquierda.



En lo primero que vamos a fijarnos es en el icono del explorador. Si pulsamos sobre él, vamos a poder ver el listado de archivos que tenemos en el directorio en el que vamos a estar trabajando, en otras palabras, la carpeta de trabajo. Por defecto no tenemos ninguna carpeta abierta.



El segundo es el menú de búsqueda, aquí podremos realizar búsquedas de los diferentes archivos que tengamos en el proyecto.



El tercero es el botón del control de código fuente. Este es un concepto más avanzado en el que se ve en cursos más avanzados, pero cuando estamos trabajando con nuestro código, lo normal es tenerlo vinculado a un repositorio, y a través de esta pestaña vamos a poder gestionar el git de nuestro proyecto.




El siguiente botón es el de ejecución y depuración, para poder compilar lenguajes que son compilados, en este caso a nosotros no nos afecta, puesto que vamos a estar trabajando con HTML, pero VSCode está pensado para ser multilenguaje.



En última instancia tenemos el botón de extensiones. A través de este elemento vamos a poder modificar nuestro VSCode, y vamos a poder instalar plugins y nuevos útiles y herramientas que nos ayude a trabajar.

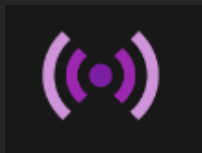
Para crear nuestro primer directorio de trabajo para comenzar a escribir nuestro código, simplemente creamos una nueva carpeta en nuestro escritorio y la llamamos por ejemplo “HTML”. Aquí dentro vamos a ir colocando todo lo que vayamos realizando. Para cargar esta carpeta en VSCode, es tan simple como arrastrarla hacia la propia IDE. Ahora podemos observar en el explorador que ya estamos trabajando dentro del directorio HTML, que es como yo he llamado a mi carpeta. Esta carpeta está totalmente vacía, por lo que vamos a crear nuestro primer archivo HTML.

 Para crear un archivo solo necesitamos pulsar en el primer icono que nos sale a la derecha del nombre de nuestra carpeta.

Después de pulsar en este icono, escribimos como se quiere llamar nuestro archivo, seguido de la extensión “.html”. Este último paso es MUY IMPORTANTE. El nombre que se le suele dar a los archivos principales .html es “index”.

EXTENSIONES Y TEMAS

En esta sección VSCode nos va a permitir añadir plugins que amplíen las funcionalidades de nuestro IDE. Las extensiones más recomendables son:



Live Server. Esta extensión nos va a permitir abrir un archivo simulando su ejecución dentro de un servidor local, es decir, como si estuviese ya subido a internet. Haciendo click derecho sobre el archivo y pulsando “Open with Live Server”, vamos a poder simular la visualización de dicho archivo en un servidor.



Thunder Client. Esta extensión nos va a permitir hacer pruebas de peticiones a una API.

A parte de instalar extensiones, también podemos instalar temas que nos permitan modificar el aspecto de nuestro entorno de desarrollo, usando nuevos esquemas de colores, por ejemplo. Un tema que podemos usar es el “SynthWave”, se instala igual que las extensiones.

También es recomendable instalar el paquete “Material Icon Theme”, muy útil para visualizar mejor los tipos de archivo.

¿QUÉ ES HTML?

HTML es un lenguaje demarcado. Dentro de nuestras construcciones web va a ser el encargado de definir la estructura de cada página web. Con los lenguajes demarcados como HTML, lo que vamos a estar usando son etiquetas. Con estas etiquetas podremos generar toda la estructura de cada una de nuestras páginas, y es que, a la hora de crear el frontal de un sitio web también llamado “FRONT END”, tenemos que tener en cuenta que vamos a usar 3 tecnologías principalmente:

1. HTML. Para crear la estructura de nuestra Página Web en el que definimos todo nuestro contenido.
2. CSS. Este es un lenguaje que usaremos para definir el estilo visual de nuestra web. Se encarga del diseño, de como se van a ver los contenidos que hemos escrito en HTML, su color, su fondo, su tipografía, etc.
3. JAVASCRIPT. Es el lenguaje de programación que nos va ayudar a añadir la lógica y el comportamiento a nuestra experiencia de usuario.

ETIQUETAS HTML

Una etiqueta HTML va a contener siempre la misma estructura.

```
<etiqueta></etiqueta>
```

Dentro de esta etiqueta, irá el contenido que esta etiqueta posee.

```
<etiqueta>  
  Contenido de la etiqueta.  
</etiqueta>
```

Los espacios, o saltos de línea que haya entre las etiquetas de apertura y la de cierre no importan, siempre y cuando el contenido se encuentre dentro de estas dos etiquetas.

También podemos encontrar a veces una abreviatura. Esto se utiliza para aquellas etiquetas que no tienen contenido dentro.

```
<etiqueta />
```

Cuando estemos trabajando con HTML vamos a disponer de una gran cantidad de etiquetas, y cada una de ellas va a estar declarando o definiendo una estructura diferente. Por ejemplo, la etiqueta “<p>” va a estar definiendo un párrafo. En este ejemplo, esta etiqueta será un párrafo que contiene un texto:

```
<p>Contenido de la etiqueta</p>
```

En su contenido, pueden contener a su vez, otras etiquetas. Por ejemplo:

```
<p><strong>Contenido de la etiqueta</strong></p>
```

A la hora de escribir nuestras etiquetas podemos encontrarnos que muchas de ellas van a necesitar de más información que su propio nombre para poder funcionar. Un ejemplo es la etiqueta “<a>”, que es la que utilizamos para hacer un enlace. Esta etiqueta necesitaría saber hacia donde va el enlace, es decir, cual es la dirección a la que nos va a llevar cuando pulsemos en este enlace. Para ello vamos a estar utilizando los atributos.

Los atributos son información adicional que añadimos cuando creamos la etiqueta, en su cabecera inicial, y nos van a ayudar a completar la función de la etiqueta que estamos escribiendo. En el caso de los enlaces podemos usar el atributo “**href**” que es el encargado de indicarnos hacia donde va a ir el enlace.

```
<a href="http://google.com">Enlace a Google</a>
```

Una etiqueta puede tener tantos atributos como necesite, y la nomenclatura será:

```
<etiqueta atributo1="XX" atributo2="YY"></etiqueta>
```

ESTRUCTURA DE UN DOCUMENTO HTML

Todos nuestros documentos HTML repetirán siempre esta estructura:

```
<!DOCTYPE html>
<html>
  <head>

  </head>
  <body>

  </body>
</html>
```

La primera etiqueta que nos encontramos, “<!DOCTYPE html>”, es una etiqueta especial con la que siempre vamos a iniciar nuestros documentos. Su función es indicar al navegador que el presente documento que estamos escribiendo es de tipo HTML.

Posteriormente declaramos nuestra etiqueta “<html>”, y dentro de esta etiqueta tenemos 2 etiquetas más, que son las etiquetas principales de nuestro html. Las etiquetas que se encuentran una dentro de otras las denominamos etiquetas hijo. Es decir, las etiquetas “<head>” y “<body>” son etiquetas hijo de “<html>”. Y “<html>” es la etiqueta padre de “<head>” y “<body>”.

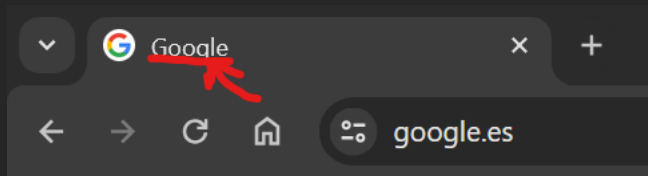
Dentro de “<head>” vamos a escribir todo el contenido que es necesario para nuestra página web, pero que no se muestra en el navegador, es decir, aquí vamos a escribir los metadatos de nuestro documento, como el título de nuestro documento, que tipo de caracteres vamos a usar, quién es el autor del código, el archivo de diseño CSS que va a estar utilizando este HTML, etc.

Posteriormente tenemos el “<body>”. Aquí es donde escribiremos todo el contenido de nuestra página web. Escribiremos todo lo que va a mostrar el navegador cuando se visite esta página: los títulos, los textos, las imágenes, etc.

La etiqueta “<html>” tiene un atributo bastante interesante en el cual podemos especificar para nuestro documento, cuál va a ser el idioma de la página web que estamos viendo. El atributo es “**lang**”.

```
<html lang="es"></html>
```

La primera etiqueta interesante que podemos añadir a “<head>” es “<title>”. Esta etiqueta va a marcar el título de nuestra página web. No se visualiza dentro de la propia página web en sí, pero sí se visualiza en la pestaña del navegador. Por ejemplo, el título de la página web de Google sería el siguiente:



En HTML se vería tal que así:

```
<title>Google</title>
```

Otra de las etiquetas que nos va a resultar muy interesante es la etiqueta “<meta>”. Esta etiqueta nos va a ayudar a añadir diferentes metadatos a nuestra página web. Algunos de los más interesantes son:

- El primero de ellos es el conjunto de caracteres que vamos a estar utilizando. Dentro del conjunto de caracteres que conforman cualquier construcción, cada lenguaje puede tener diferentes símbolos que son especiales de ese lenguaje. Cuando estamos trabajando en español, es interesante marcar que el charset va a ser el de “UTF-8”, que es el conjunto de caracteres que incluye las tildes, los acentos, las eñes, etc.

```
<meta charset="UTF-8">
```

- Otro es “http-equiv”. Los navegadores son los encargados de interpretar el código HTML que nosotros escribimos para que el usuario pueda visualizar correctamente la página web. Con esta instrucción nos estamos asegurando una correcta compatibilidad para que los usuarios de ciertos navegadores como Edge, pueda visualizar nuestra página web tal y como nosotros la hemos escrito.

```
<meta http-equiv="X-UA-Compatible" content="IE-edge">
```

- Con la explosión de la tecnología móvil ha aumentado considerablemente el número de diferentes pantallas a través de las cuales se accede a nuestra página web. Ahora también podemos acceder a través de un dispositivo móvil o una Tablet. Con esta instrucción nos aseguramos de que el ancho de la pantalla es la escala inicial de nuestro contenido.

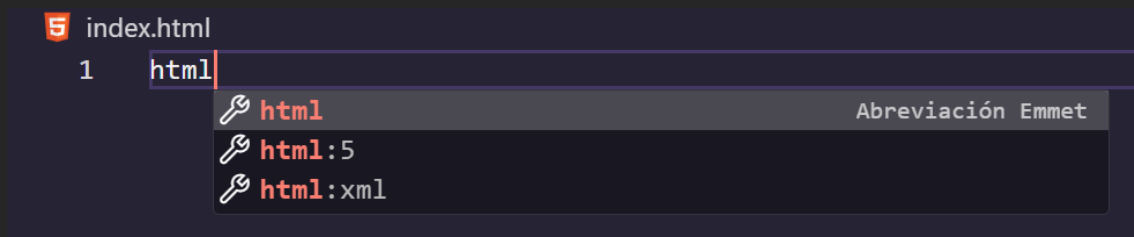
```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```


SNIPPETS, EMMET Y MENÚ CONTEXTUAL

VSCoide nos va a ofrecer diferentes opciones para facilitarnos la escritura de código, los snippets, las abreviaciones emmet y el menú contextual.

Los Snippets son trozos de código ya encapsulados que podemos asociar a una palabra, para que, cuando escribamos dicha palabra, el editor de código nos autocomplete colocando todo ese trozo de código por nosotros.

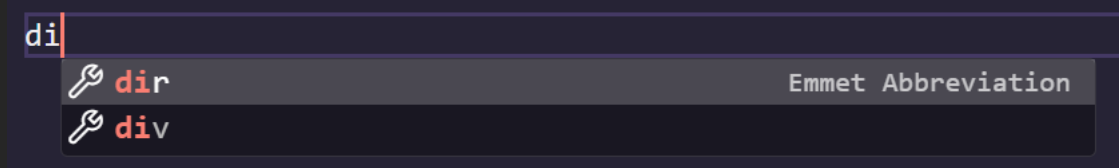
Si escribimos la palabra `html` en un archivo vacío `.html` nos aparecerá lo siguiente:



Si seleccionamos la versión “`html:5`” VSCoide escribirá por nosotros toda la escritura base de nuestro documento HTML tal y como hemos visto en el punto anterior.

Las abreviaciones emmet están pensadas para cuando tengamos que escribir múltiples trozos de código, que se van a repetir, como podría ser el caso de escribir 5 párrafos. Para ello basta con escribir “`p*5`” y se nos crearán 5 párrafos.

El menú contextual nos va a aparecer cada vez que estemos escribiendo código, y nos va a sugerir diferentes autocompletados en función de lo que estemos escribiendo. Por ejemplo, si queremos escribir la etiqueta “`div`”, con solo poner “`di`”, ya podremos autocompletarlo:



Si alguna vez se desactiva el menú contextual, se puede forzar el hecho de que aparezca pulsando “Control + Espacio”

COMENTARIOS

En HTML los comentarios se escriben de la siguiente manera:

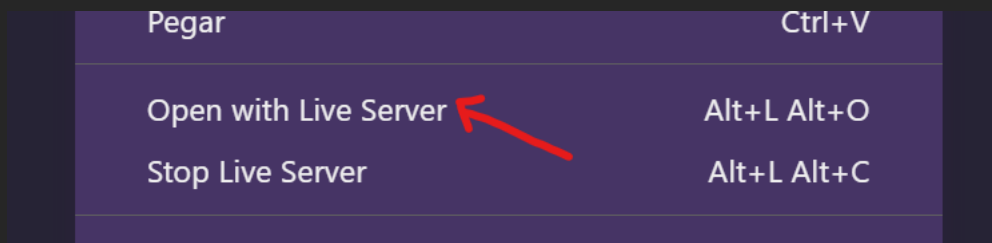
```
<!--Hola, soy un comentario-->
```

HOLA MUNDO

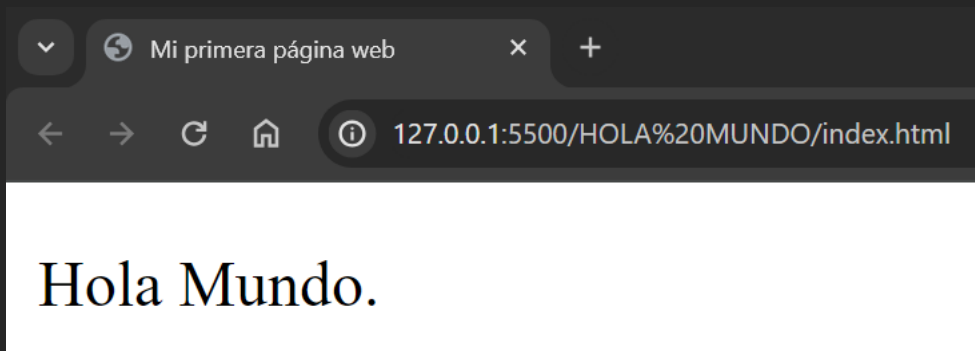
Para enfrentarnos a nuestro primer código, creamos un nuevo proyecto llamado “HOLA MUNDO” y lo cargamos en VSCode. Utilizando el snippet de html:5 colocamos la estructura básica de todo documento. Una vez la tenemos, realizamos algunas modificaciones como el lenguaje y el título del documento.

Dentro de body incluimos un párrafo que diga: “Hola mundo.”

Una vez escrito nuestro código guardamos el documento con “CONTROL + S”. Como ya lo hemos guardado, ya podemos visualizarlo. Para ello damos click derecho sobre nuestro documento y seleccionamos “Open with Live Server”:



Nos aparecerá en nuestro navegador de la siguiente manera:

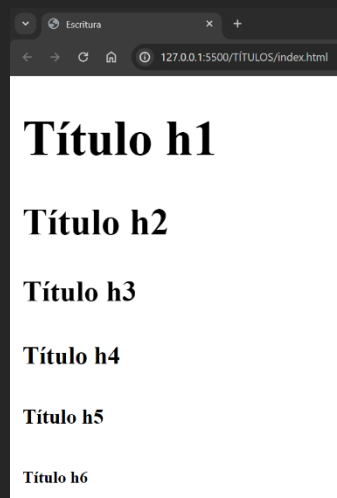


TEXTOS

TÍTULOS Y PÁRRAFOS

Para comenzar a trabajar creamos un nuevo directorio de trabajo llamado “TÍTULOS”. Escribimos la estructura del documento lanzando el snippet “html:5”.

Los títulos van a ser escritos con las etiquetas que van desde el “<h1>” hasta el “<h6>”. El número va a indicar la importancia del título que vamos a estar escribiendo, siendo h1 el título principal y h6 el menos importante. Esto vamos a poder observarlo con el tamaño de la tipografía que por defecto HTML va a ubicar para cada uno de los títulos, siendo h1 el más grande y h6 el más pequeño.



```
<body>
  <h1>Título h1</h1>
  <h2>Título h2</h2>
  <h3>Título h3</h3>
  <h4>Título h4</h4>
  <h5>Título h5</h5>
  <h6>Título h6</h6>
</body>
```

Las etiquetas de título decimos que son etiquetas de bloque, esto quiere decir, que por defecto, al escribirlas, cuando estas terminan, el contenido salta automáticamente a la siguiente línea por defecto, es decir, no tienen contenido en sus laterales.

La etiqueta “<p>” es la que nos va a permitir escribir párrafos. A estos párrafos vamos a poder enriquecerle el texto utilizando o acudiendo a más etiquetas.

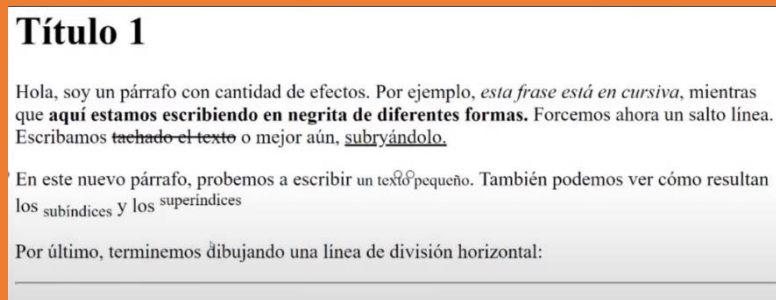
ETIQUETA	FUNCIÓN
<i>	Cursiva
 o 	Negrita
	Tachado
<u>	Subrayado
<sub>	Subíndice
<sup>	Superíndices
<small>	Textos pequeños
 	Salto de línea
<hr/>	Línea división horizontal

```
<body>
  <h1>Título h1</h1>
  <p>Párrafos</p>
  <i>Cursiva</i>
  <b>Negrita1</b> <strong>Negrita 2</strong>
  <del>Tachado</del>
  <u>Subrayado</u>
  <sub>Subíndice</sub>
  <sup>Superíndices</sup>
  <small>Textos pequeños</small>
  <br><!--Salto de Línea-->
  <hr><!--Línea de división horizontal-->
</body>
```



ACTIVIDAD 1

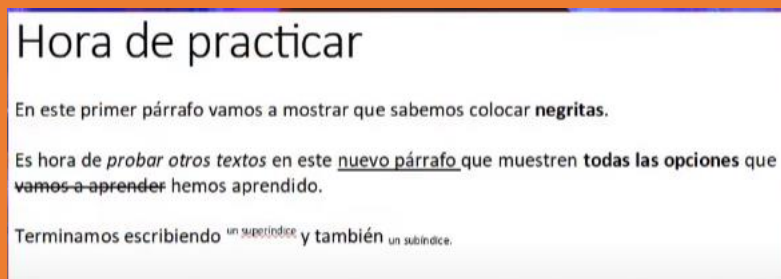
Diseñar la siguiente página web:



..\DIRECTORIOS\TEXTOS\Actividad1_TextosBásicos.html

ACTIVIDAD 2

Diseñar la siguiente página web:



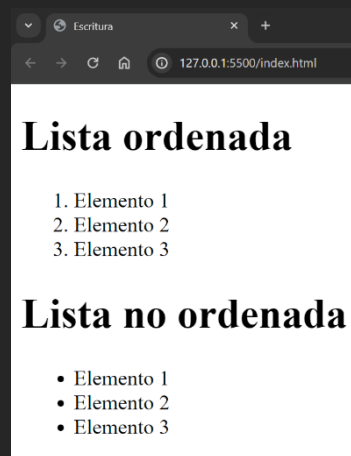
..\DIRECTORIOS\TEXTOS\Actividad2_TextosBásicos2.html

LISTAS

Tenemos principalmente dos tipos de listas. Las listas ordenadas y las listas no ordenadas.

ETIQUETA	FUNCIÓN
<code></code>	Lista ordenada
<code></code>	Lista no ordenada
<code></code>	Elementos de la lista

```
<body>
  <h1>Lista ordenada</h1>
  <ol>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
  </ol>
  <h1>Lista no ordenada</h1>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
  </ul>
</body>
</html>
```



La principal diferencia se basa en el aspecto de sus viñetas, de los bullet points que coloca delante.

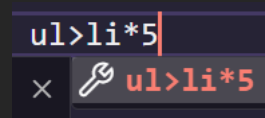
Si deseas escribir una lista muy larga, la tarea de escritura puede hacerse un tanto tediosa, para hacer la tarea más sencilla, disponemos de una abreviación emmet en VSCode para escribirlas más rápido. La abreviación será de la siguiente manera:

`tipo-de-la-lista>li*numero-de-elementos`

Por ejemplo, si queremos escribir una lista desordenada con 5 elementos, con escribir “`ul>li*5`” se nos creará automáticamente.

Abreviación Emmet
16

```
<ul>
  <li>|</li>
  <li>|</li>
  <li>|</li>
  <li>|</li>
  <li>|</li>
</ul>
```



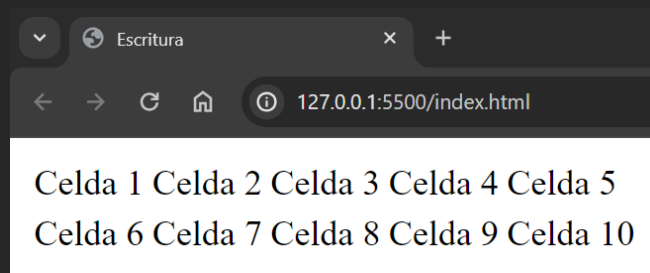
Si observamos, el símbolo “>” lo que nos está indicando es “dentro de este”. Así pues, podríamos traducir la expresión anterior como: Créame una etiqueta ul, y dentro de esta, me creas li multiplicado por 5. Este tipo de abreviaciones se pueden utilizar para cualquier elemento que queramos

TABLAS

Es hora de organizar nuestro contenido en elementos un poco más complejos.

ETIQUETA	FUNCIÓN
<code><table></code>	Tabla
<code><tr></code>	Fila
<code><td></code>	Celdas fila
<code><th></code>	Celda destacada

```
<table>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
    <td>Celda 3</td>
    <td>Celda 4</td>
    <td>Celda 5</td>
  </tr>
  <tr>
    <td>Celda 6</td>
    <td>Celda 7</td>
    <td>Celda 8</td>
    <td>Celda 9</td>
    <td>Celda 10</td>
  </tr>
</table>
```

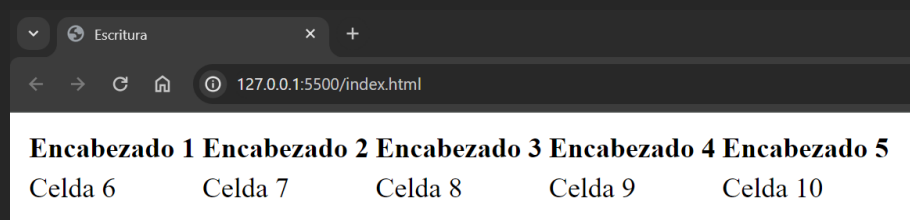


Podemos convertir cada una de las celdas normales “`<td>`”, en una celda de encabezado, es decir, en una celda destacada, para ello usamos “`<th>`”.

```
<tr>
  <th>Celda 1</th>
  <th>Celda 2</th>
  <th>Celda 3</th>
  <th>Celda 4</th>
  <th>Celda 5</th>
</tr>
```

Si en el ejemplo anterior cambiamos los `td` de la primera fila por `th` veremos la diferencia. Para modificar el texto en varias líneas a la vez en VSCode, manteniendo el “ALT”, y haciendo click en las zonas que queremos modificar, se modificarán todas a la vez.

```
<table>
  <tr>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
    <th>Encabezado 4</th>
    <th>Encabezado 5</th>
  </tr>
  <tr>
    <td>Celda 6</td>
    <td>Celda 7</td>
    <td>Celda 8</td>
    <td>Celda 9</td>
    <td>Celda 10</td>
  </tr>
</table>
```

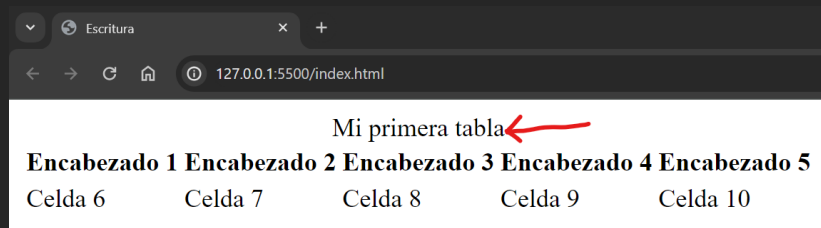


Uno de los añadidos que podemos aplicar a una tabla es agregarle un título a la tabla. Esto lo hacemos con la etiqueta de caption.

ETIQUETA	FUNCIÓN
<code><caption></code>	Título tabla

A la anterior tabla que hemos generado podríamos añadir un título

```
<table>
  <caption>Mi primera tabla</caption>
  <tr>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
    <th>Encabezado 4</th>
    <th>Encabezado 5</th>
  </tr>
  <tr>
    <td>Celda 6</td>
    <td>Celda 7</td>
    <td>Celda 8</td>
    <td>Celda 9</td>
    <td>Celda 10</td>
  </tr>
</table>
```

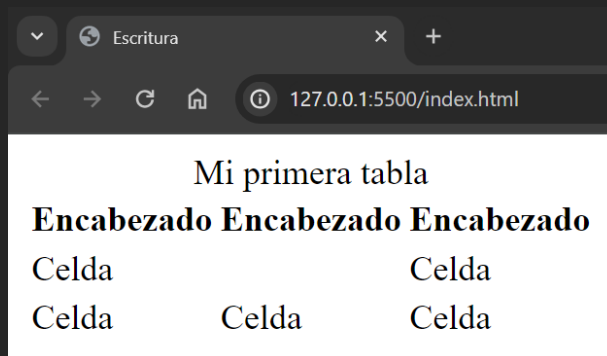


Mi primera tabla				
Encabezado 1	Encabezado 2	Encabezado 3	Encabezado 4	Encabezado 5
Celda 6	Celda 7	Celda 8	Celda 9	Celda 10

Cuando estamos estructurando una tabla, suele ser común hacer que una celda de dicha tabla ocupe más de una celda, es decir, que se expanda vertical u horizontalmente para ocupar más. Sería en otras palabras, unir celdas. Para esto utilizamos los atributos “**colspan**” y “**rowspan**”, para indicar cuantas celdas queremos que se expandan, siendo colspan una expansión horizontal, y rowspan una expansión vertical.

ATRIBUTO	FUNCIÓN
<code>colspan</code>	Expandir horizontalmente (Columnas)
<code>rowspan</code>	Expandir verticalmente (Filas)

```
<table>
  <caption>Mi primera tabla</caption>
  <tr>
    <th>Encabezado</th>
    <th>Encabezado</th>
    <th>Encabezado</th>
  </tr>
  <tr>
    <td colspan="2">Celda</td>
    <td>Celda</td>
  </tr>
  <tr>
    <td>Celda</td>
    <td>Celda</td>
    <td>Celda</td>
  </tr>
</table>
```



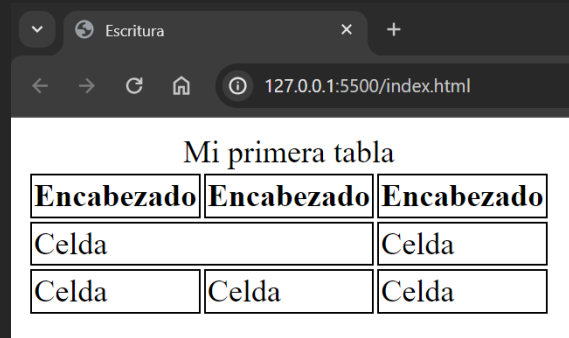
Mi primera tabla		
Encabezado	Encabezado	Encabezado
Celda	Celda	
Celda	Celda	Celda

La primera celda de la segunda fila está ocupando en este caso 2 columnas de la tabla.

De forma excepcional, para que se pueda ver bien este aspecto, vamos a aplicar un borde a cada una de las celdas para que podamos apreciarlo mejor, toda esta parte es CSS, pero es simplemente para que se pueda apreciar mejor

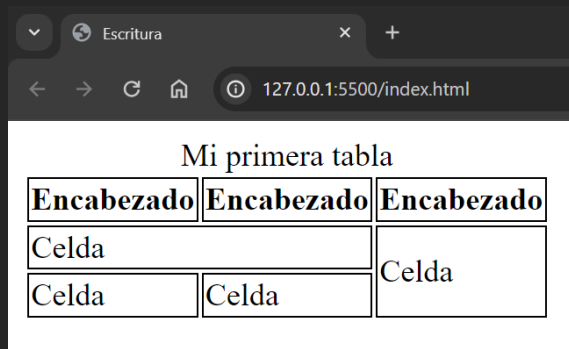
El CSS necesario para que se vean los bordes de las tablas sería el siguiente:

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Escritura</title>
  <style>
    th,td{
      border: 1px solid black;
    }
  </style>
</head>
```



También podemos extender nuestra celda verticalmente con el atributo rowspan.

```
<table>
  <caption>Mi primera tabla</caption>
  <tr>
    <th>Encabezado</th>
    <th>Encabezado</th>
    <th>Encabezado</th>
  </tr>
  <tr>
    <td colspan="2">Celda</td>
    <td rowspan="2">Celda</td>
  </tr>
  <tr>
    <td>Celda</td>
    <td>Celda</td>
  </tr>
</table>
```



A veces puede resultarnos interesante definir diferentes áreas dentro de nuestra tabla, distinguiendo una primera parte de nuestra tabla, la cabecera, después el cuerpo central con todo el contenido, y al final un pie de tabla.

ETIQUETA	FUNCIÓN
<thead>	Cabecera tabla
<tbody>	Cuerpo tabla
<tfoot>	Pie de tabla

```
<table>
  <thead>
    <th>Encabezado 1</th>
    <th>Encabezado 2</th>
    <th>Encabezado 3</th>
  </thead>
  <tbody>
    <tr>
      <th>Celda 1</th>
      <td>Celda 2</td>
      <td>Celda 3</td>
    </tr>
    <tr>
      <th>Celda 4</th>
      <td>Celda 5</td>
      <td>Celda 6</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th colspan="3">Pié de página</th>
    </tr>
  </tfoot>
</table>
```

Encabezado 1	Encabezado 2	Encabezado 3
Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6
Pié de página		

ACTIVIDAD 3

Diseñar la siguiente página web:

Registro de horas								
	Juan		Raúl		Sonia		Pedro	
Día	Entrada	Salida	Entrada	Salida	Entrada	Salida	Entrada	Salida
Lunes	9:00	16:00	8:30	15:30	9:00	16:00	8:30	15:30
Martes	9:00	16:00	8:30	15:30	9:00	16:00	8:30	15:30
Miércoles	9:00	16:00	8:30	15:30	9:00	16:00	8:30	15:30
Jueves	9:00	16:00	8:30	15:30	9:00	16:00	8:30	15:30
Viernes	9:00	16:00	8:30	15:30	9:00	16:00	8:30	15:30
Fecha	18/04/22							

..\DIRECTORIOS\TEXTOS\Actividad3_Tablas.html

MULTIMEDIA

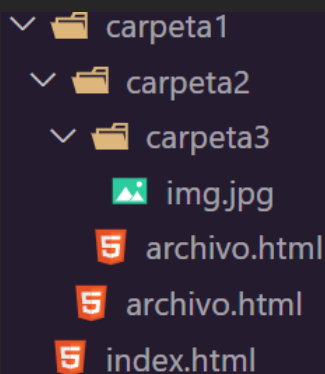
CÓMO CONSTRUIR RUTAS

Vamos a comenzar a añadir contenido multimedia nuestras construcciones, esto significa que vamos a estar añadiendo imágenes, vídeos, audios. Por ello, vamos a necesitar indicar a dichas etiquetas dónde se encuentran los recursos, es decir, vamos a tener que mostrarle a nuestro código la ruta para que el navegador pueda cargar correctamente la información. Y por ello vamos a ver como se construye una ruta, los tipos de rutas y cómo usarlas de forma apropiada.

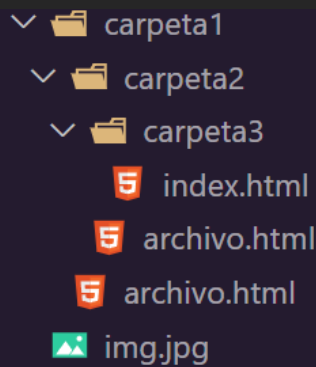
Dentro de los tipos de rutas vamos a distinguir siempre entre dos grupos principales, las rutas absolutas y las rutas relativas.

En cuanto a las absolutas, estas son las más fáciles de escribir, ya que contienen el camino completo hacia el recurso. No importa dónde se encuentre el archivo, que va a hacer la llamada hacia este recurso, ya que, la llamada absoluta siempre se escribe igual. Por ejemplo, si se quiere hacer uso de una imagen que se encuentra en esta ubicación: "<https://picsum.photos/200/300>", su ruta absoluta sería exactamente esa, expresando esta dirección con todos sus apartados. Usaremos este tipo de rutas cuando llamemos a recursos externos que se encontrarán fuera de nuestro sitio web, por ejemplo, cuando queremos una foto alojada en Google. Para el resto de ocasiones, que van a ser la gran mayoría, vamos a estar usando siempre las rutas relativas, ya que las absolutas, en ciertas ocasiones, van a poder conducirnos a error.

En cuanto a las rutas relativas, son aquellas en las cuales no especificamos el camino completo para llegar a ellas, sino que vamos a expresarlo cómo llegar desde el recurso, desde la posición en la que se encuentre el recurso que va a estar llamando a ese elemento. Un ejemplo de ruta relativa sería este de aquí: "img/equipo/miembro1.jpg". En esta ruta estamos indicando que el archivo se encuentra desde la ubicación en la que el documento que estamos escribiendo está, vamos a estar accediendo a una carpeta que se llama "img", que está a este nivel, y una vez dentro, nos encontraremos otra carpeta que se llama "equipo", y accediendo a esta carpeta equipo, dentro, vamos a encontrar ya la imagen. Es decir, en las rutas relativas no vamos a estar refiriéndonos con la dirección completa, sino que desde donde se encuentre el archivo de código que estoy escribiendo, voy a ir indicando cómo se llega desde ese archivo hacia el recurso que busco.



Para adentrarnos en una carpeta, colocamos simplemente el nombre de la carpeta y posteriormente usamos "/". Por ejemplo, en esta estructura, si queremos usar la imagen "img.jpg" que se encuentra en la carpeta 3, desde el archivo index.html, la ruta sería: "carpeta1/carpeta2/carpeta3/img.jpg". Sin embargo, si queremos usar ese mismo recurso, pero desde el documento "archivo.html" que se encuentra en la carpeta 2, la ruta será: "carpeta3/img.jpg".



Si queremos utilizar un recurso que se encuentra fuera de la carpeta desde la cual lo llamamos utilizaremos la expresión “../”. Con esto lo que indicamos es que salimos un nivel desde nos encontramos. Suponiendo esta estructura de archivos, y situándonos en el index que está dentro de la carpeta 3, para construir la ruta correctamente necesitaríamos indicar que queremos salir de varias carpetas. Para ello utilizaríamos la siguiente expresión: “../../../img.jpg”.

La expresión “../” es lo mismo que decir “Esta carpeta”.

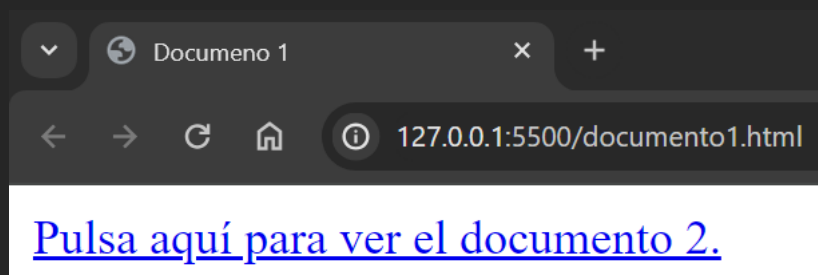
ENLACES

Los enlaces son la base de toda estructura web, ya que, nos van a permitir navegar entre los diferentes documentos. Cuando pulsamos sobre un enlace le navegador saldrá del documento actual y cargará el documento al que referencie dicho enlace. Para realizar un enlace vamos a utilizar la etiqueta “<a>”.

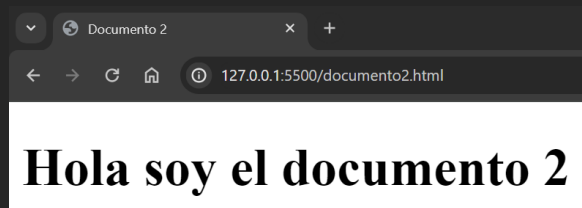
ETIQUETA	FUNCIÓN
<a>	Enlace

Dentro de la etiqueta vamos a colocar el texto que va a ser enlazado. Esta etiqueta va a necesitar el atributo “href”. En este atributo es donde tenemos que indicar la ruta para poder llegar al documento que enlazamos.

```
<a href="documento2.html">Pulsa aquí para ver el documento 2.</a>
```



Si pulso en el enlace nos llevaría al documento 2 referenciado en la ruta:



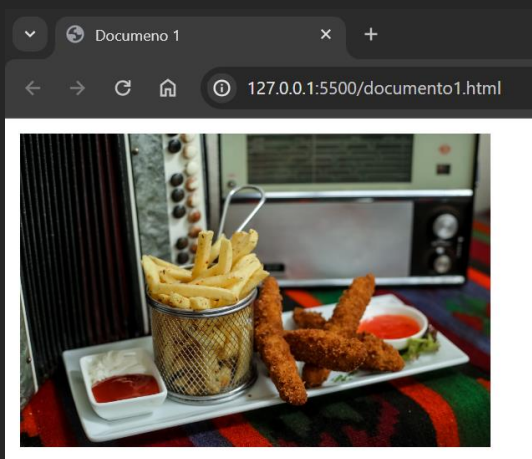
Hemos visto el comportamiento por defecto de nuestro enlace, que es que cuando pulsamos, se nos carga directamente en el navegador el siguiente de los archivos, sin embargo, vamos a poder modificar este comportamiento con el atributo **"target"**. Este atributo nos va a permitir modificar el comportamiento de como queremos abrir ese archivo. Por defecto, el valor que contiene es "self", que significa "Esta misma pestaña". Si no especificamos nada, aplicará self.

ATRIBUTO	FUNCIÓN
target="_self"	En la misma pestaña
target="_blank"	Nueva pestaña
target="_parent "	Carga el archivo en el padre del enlace
target="_top"	Carga el archivo en el body del documento del enlace

IMÁGENES

Para incluir una imagen vamos a estar utilizando la etiqueta **""**, y al igual que pasa en otras etiquetas, necesitará un atributo para indicar la ruta de la imagen en este caso, este atributo es **"src"**.

ETIQUETA	FUNCIÓN
	Imagen



```

```

Uno de los atributos que podríamos estar incorporando dentro de la etiqueta de imagen es el atributo **"alt"**. Este atributo es un texto que explica qué contiene la imagen. No va a ser visualizado en el contenido, pero, es muy útil para robots de búsqueda y el posicionamiento SEO de una página, ya que ayuda a entender a los diferentes motores qué es lo que se muestra en la imagen.

Solo veríamos ese texto si hay algún error en la carga de la imagen.

```

```

Además, a la imagen también podríamos estar especificándole una anchura, usando el atributo “width”, y también una altura con el atributo “height”.

ACTIVIDAD 4

Realiza un proyecto nuevo que tenga una estructura parecida a esta, en la que tengas una página con un enlace a otra página donde aparezca una imagen con dimensiones adecuadas para su visualización, y en esta segunda página, haya otro enlace para volver a la página principal. Además, habrá una tercera página que se abrirá en una nueva pestaña.

Página principal	Segunda página	Tercera página
		

..\DIRECTORIOS\MULTIMEDIA\Actividad4_EnlacesImágenes

VÍDEOS

En la etiqueta de vídeo, al igual que con las imágenes necesitaremos valernos de un atributo “src” mediante el cual indicaremos la ubicación donde se encuentra el archivo. También podemos añadirle un “width”, un “height”.

```
<video src="videos/video.mp4">
  Video no soportado...
</video>
```

Si el navegador que usa el usuario es antiguo y no soporta video podemos colocar un mensaje que se va a mostrar para él dentro de la propia etiqueta.

ETIQUETA	FUNCIÓN
<code><video></code>	Video

Uno de los puntos a tener en cuenta cuando colocamos vídeos en nuestra web son los formatos, los diferentes tipos de vídeo. Si bien el formato .mp4 es el más extendido y usado, existen otro tipo de formatos como el .ogg o el .webm que también son útiles. Si disponemos de nuestro vídeo en diferentes formatos podemos especificar en nuestra etiqueta de video, la ruta a todos ellos, para que el navegador utilice la versión más favorable. Para ello, en lugar de especificar la ruta en el src, dentro de la etiqueta `<video>`, ubicamos otra etiqueta `<source>`. A esta etiqueta le especificamos el atributo “src” en el cual le indicamos la ruta al vídeo, también le vamos a indicar otro atributo más, “type”, que va a indicar qué tipo de vídeo es.

```
<video>
  <source src="videos/video.mp4" type="video/mp4">
  <source src="videos/video.ogg" type="video/ogg">
  <source src="videos/video.webm" type="video/webm">
  Video no soportado...
</video>
```

Una vez insertado el vídeo, existen otros atributos que pueden ser de interés, por ejemplo, el añadir al vídeo un reproductor, unos controles que nos permitan manejarlo. Para ello, en la etiqueta `<video>` añadimos el atributo “controls”. No hace falta especificar ningún valor para este atributo.

Otro atributo interesante es el de “autoplay”, con este atributo lo que hacemos es, que cuando entramos en la página, el vídeo se empieza a reproducir por defecto.

Por último, tenemos otro atributo interesante para nuestra etiqueta `<video>`, el cual es el atributo “loop”. Este atributo hace, que cuando el vídeo haya terminado, este volverá a comenzar por el principio automáticamente en bucle.

AUDIO

Esta etiqueta funciona muy similar a la etiqueta `<video>`. También podemos añadirle un mensaje en caso de que no se pueda reproducir. Exactamente igual a la etiqueta `<video>`. Al igual que nos ocurría anteriormente, tenemos que especificar ciertos controles y herramientas para que podamos reproducirlo correctamente.

A la etiqueta de `<audio>` vamos a poder especificar diferentes recursos, siguiendo el mismo formato que en la etiqueta de `<video>`.


```
<audio controls autoplay loop>
  <source src="audios/audio.mp3" type="audio/mp3">
  <source src="audios/audio.wav" type="audio/wav">
  Audio no soportado...
</audio>
```

El único atributo que no aplicaría para el audio, y sí aplica para el vídeo, son los de anchura y altura, puesto que el audio no tiene dimensiones, solo sonido.

En función del navegador, muchas de las últimas versiones están despreciando el “autoplay” para el audio, de tal forma de que una página web no pueda ser invasiva para el usuario que la está navegando. Entonces es posible que esta función no se encuentre ejecutada, esto no es fallo del código.

IFRAME

El iframe es una etiqueta que debe usarse con cuidado, sin embargo, aún sigue siendo útil en algunos escenarios. El iframe nos permite incrustar un HTML dentro de otro. Por ejemplo, incrustar una página web dentro de la nuestra en una sección determinada. Solo se deben usar iframes para sitios confiables, como por ejemplo cargar desde YouTube o Spotify o cualquiera similar.

Compartir



Insertar



WhatsApp



goo

Dentro de los vídeos de YouTube en la sección de compartir, podemos ver que una de las opciones es la de “Insertar”.

Insertar vídeo

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/
bCKk1TU2780?si=96EwI2NtXYxponeN"
title="YouTube video player"
frameborder="0"
allow="accelerometer; autoplay;
clipboard-write; encrypted-media;
gyroscope; picture-in-picture; web-
share" allowfullscreen></iframe>
```

Cuando pulsamos sobre insertar, YouTube nos va a ofrecer ya una etiqueta <iframe> para que podamos incrustar este vídeo en nuestra página web.

Si insertamos esta etiqueta en nuestra página web lo veremos de la siguiente manera:

```
<iframe width="560" height="315" src="https://www.youtube.com/embed/bCKk1TU2780?si=96EwI2NtXYxponeN"
title="YouTube video player" frameborder="0"
allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share"
allowfullscreen></iframe>
```

FORMULARIOS

FOMULARIOS

Con los formularios, no solo podemos hacer secciones de contacto, sino que además, podemos hacer todo tipo de estructuras para que nuestros usuarios se comuniquen con nosotros.

Cada vez que vayamos a definir una zona de interacción donde vamos a recoger datos donde el usuario va a tener que introducir, vamos a tener que crear un formulario con la etiqueta `<form>`.

ETIQUETA	FUNCIÓN
<code><form></code>	Formulario

Dentro de esta etiqueta `<form>` iremos añadiendo cada uno de los elementos que vamos a posicionar para que nuestro usuario pueda introducir datos, es decir, las cajas de texto, de números, los selectores, etc. Antes de ello necesitamos ubicar algunos de los atributos necesarios para el formulario.

Principalmente, cuando estamos trabajando con un formulario vamos a tener dos atributos de especial importancia. Uno es el atributo `"method"`, y otro es el atributo `"action"`. El `action` es un atributo que va a indicar la ruta donde se encuentra el archivo que va a procesar los datos que el usuario nos envíe. El segundo de los atributos es el `method`, este atributo va a contener un verbo que indica cómo viajarán los datos al fichero que los va a procesar. Principalmente podemos destacar los verbos de `"get"`, `"post"`, `"put"`, `"delete"` y `"path"`. De todos estos nos quedaremos con los más importantes que son el `"get"` y el `"post"`.

Utilizamos `get` cuando estamos enviando datos que no son sensibles, es decir, cuando la información que estamos recibiendo por parte del usuario no es privada. Por ejemplo, una barra de búsqueda, pero no un formulario de login, puesto que vamos a estar enviando la contraseña del usuario.

`Post` va a encriptar la información en la petición y no será visible por terceros. Usaremos esta opción cuando estemos enviando datos que son sensibles. Sería darle un buen uso como ya hemos dicho anteriormente, en un login donde el usuario tenga que introducir la contraseña, o donde tengamos que introducir los datos de nuestra tarjeta de crédito, etc.

```
<form method="post" action="backend/form.php">

</form>
```

INPUTS DE TEXTO

ETIQUETA	FUNCIÓN
<code><input></code>	Caja de texto

Para crear una caja de texto vamos a usar la etiqueta “<input>”, y a dicha etiqueta tenemos que darle un atributo “**type**” para indicar qué tipo de caja de texto va a ser. Además, deberemos de darle una propiedad, un atributo “**name**”. Indicando en este name un nombre identificativo a cada una. Nos vale cualquier nombre siempre y cuando no contenga tildes, ni eñes, ni espacios, es decir, una única palabra identificativa. Solo puede contener textos y números.

```
<input type="" name="" />
```

TYPE	VALOR
text	<input type="text"/>
number	<input type="number" value="0"/>
password	<input type="password" value="....."/>
date	<input type="date" value="dd/mm/aaaa"/>
email	<input type="email" value="email@gmail.com"/>
file	<input type="file" value="Seleccionar archivo"/> Ninguno archivo selec.

Cuando tengamos un formulario, que dentro tiene algún tipo de input que sea file, vamos a tener que indicar como atributo:

```
enctype="multipart/form-data"
```

TYPE	VALOR
time	<input type="time"/>
color	<input type="color"/>
datetime-local	<input type="datetime-local"/>
month	<input type="month"/>
search	<input type="search"/>
url	<input type="url"/>
week	<input type="week"/>

Vamos a tener dos formatos distintos para poder guiar a nuestro usuario sobre qué debe introducir.

ETIQUETA	FUNCIÓN
<code><label></code>	Texto explicativo de la caja de texto

Esta etiqueta es para añadir un texto explicativo de lo que debe colocar el usuario. Si queremos ser más explícitos, con el atributo `for` e indicarle que es para el nombre del input, y al input se le añade un atributo `id` para que el label pueda referirse a él. La construcción es la misma, pero a nivel de información estamos indicando que el label es para un input determinado.

```
<label for="nombre">Nombre: </label>
<input type="text" name="txtNombre" id="nombre">
```

Formularios

127.0.0.1:5500/index.html?

Nombre:

Otro de los formatos que tenemos para especificar o guiar al usuario que puede hacer, es el atributo “**placeholder**”. Nos sirve para escribir un texto como de borrador, que está por debajo, una marca de agua, que vamos a colocar dentro de la caja de texto para guiar al usuario.

```
<input type="text" placeholder="Escribe aquí tu texto">
```

También podemos darle un valor por defecto a el elemento con el atributo “**value**”.

```
<input type="text" value="Hola Mundo">
```

También se pueden modificar las cajas de texto con el atributo “**size**”.

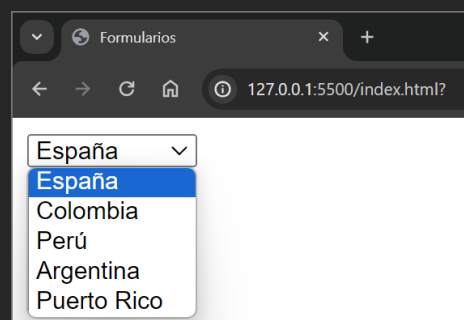
SELECTS

Ahora vamos a conocer la etiqueta “<select>”. Esta etiqueta nos permitirá crear una caja con un menú desplegable donde el usuario tendrá que seleccionar entre las opciones disponibles.

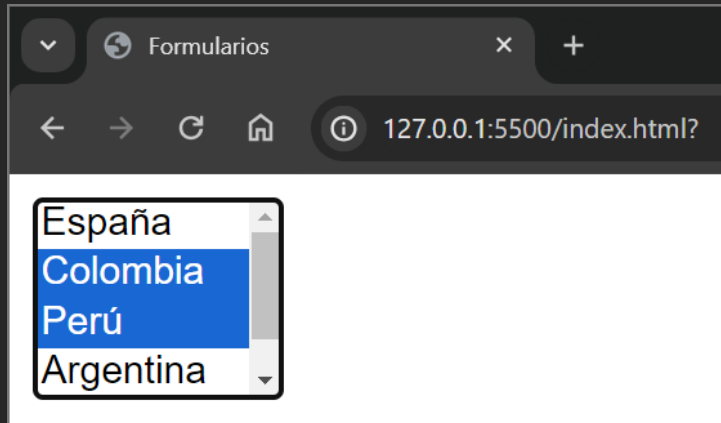
ETIQUETA	FUNCIÓN
<select>	Menú desplegable
<option>	Opciones del desplegable

Al igual que hemos hecho con los inputs, a este select, se le debe dar un “**name**”, para que así el backend pueda obtener la información. Definimos con la etiqueta “<option>” cada una de las opciones que va a tener el desplegable. El texto que se encuentra dentro de la etiqueta de option es información que le pasamos al usuario, pero tenemos que enviarle al backend la información de la opción seleccionada, y para ello vamos a establecer dentro de la etiqueta option un atributo “**value**”, y este va a ser el valor que se va a enviar.

```
<select name="pais">
  <option value="es">España</option>
  <option value="co">Colombia</option>
  <option value="pe">Perú</option>
  <option value="ar">Argentina</option>
  <option value="pr">Puerto Rico</option>
</select>
```



Al campo select podemos añadirle el atributo “multiple”. Si lo indicamos, lo que estamos diciendo es que ahora el usuario puede escoger entre varios valores pulsando la letra CONTROL.



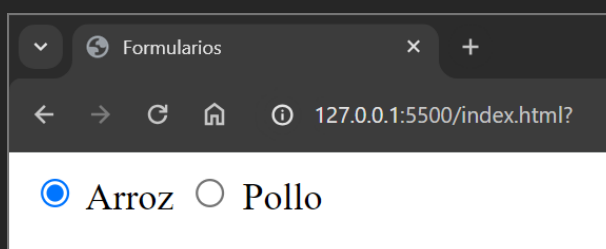
RADIOBUTTONS

Estos son botones circulares, por defecto, que podemos utilizar para marcar o no una selección dentro de un grupo de elementos. Normalmente se usa para darnos a escoger entre una de las diferentes opciones que muestra la página. Los vamos a expresar a través de una etiqueta “<input>” que va a ser de tipo “radio”. A estos inputs, también tenemos que darle un “name” además de un “value” como hemos hecho con los selects. Este input no va a mostrar la información que contiene, simplemente tendríamos un círculo que podríamos seleccionar o no. Para poder modificar esto, vamos a necesitar nosotros añadirle un “<label>” en la cual vamos a indicar el value del input para que el usuario pueda ver qué está seleccionando si marca esa casilla.

```
<input type="radio" name="rbComida" value="arroz" id="arroz" /> <label for="arroz">Arroz</label>
```

☐ Arroz

Si añadimos un elemento más podemos ver como sólo se selecciona una, esto es debido a que ambas son solución para el mismo elemento, que es el elemento de “comida” en este caso. Si queremos que una de las opciones se quede seleccionada de manera predeterminada, podemos añadir el atributo “checked”.



CHECKBOX

El checkbox es esa cajita que marcamos cada vez que pulsamos sobre ese mensaje de “Acepto términos y condiciones”. La forma de construirlo es muy similar a como hacemos con los botones de radio.

```
<input type="checkbox" name="chkTerminos" value="yes" id="terminos"/> <label for="terminos">Acepto</label>
```

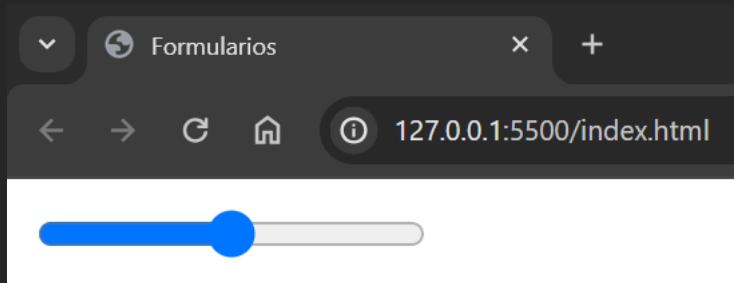


Por defecto, los checkbox son de selección múltiple.

RANGOS

Con los rangos podemos pintar una barra deslizadora para que el usuario ajuste un valor numérico entre dos valores. Para indicar estos valores vamos a utilizar los atributos “min” y “max” de cara a marcar el valor mínimo y el valor máximo del rango respectivamente.

```
<input type="range" name="porcentaje" min="0" max="100" />
```



TEXTAREA

A la hora de escribir textos es posible que una única línea, como nos ofrecen los inputs, no sea suficiente, para esas ocasiones vamos a usar el “<textarea>”

ETIQUETA	FUNCIÓN
<textarea>	Área de texto

Formularios

127.0.0.1:5500/index.html

Escribe aquí tu comentario

Para darle las dimensiones apropiadas a esta caja vamos a poder utilizar algunos atributos como son el atributo de “cols” y de “rows”, columnas y filas respectivamente, que nos van a ayudar a indicar su dimensión. Al igual que ocurre con los inputs, vamos a poder aplicarle un “placeholder” o cualquier otro tipo de elemento. Si además queremos que el texto contenga un mínimo de caracteres o un máximo, vamos a poder utilizar también los atributos de “minlength” y “maxlength” que son longitud mínima y máxima respectivamente.

```
<textarea name="comentario" cols="30" rows="10" minLength="10" maxLength="500" placeholder="Escribe aquí tu comentario"></textarea>
```

VALIDACIONES

Si bien los aspectos de validación, es decir, comprobar que los datos que se han introducido se contienen dentro de lo que estamos esperando, por ejemplo, que la edad no sea un texto o que el nombre no contenga números, corresponde principalmente a el backend, desde HTML podemos hacer unas validaciones iniciales. Ya hemos visto con el textarea que podemos indicar una longitud mínima y máxima. También podemos observar cuando hemos estado utilizando por ejemplo el input de tipo número, que no nos deja introducir textos, solo números. Además de todo esto vamos a poder añadir alguna validación con HTML antes de hacer el envío, y principalmente, la que más destaca es el uso de “required”. A todas las etiquetas que podemos colocar dentro de un formulario, podemos añadirle también “required”. Con este atributo vamos a estar indicando que ese campo es obligatorio. Si no está, no enviará los datos. Por ejemplo, si hacemos un checkbox en el que tiene que marcarlo para aceptar los términos, sería un buen uso este atributo.

Formularios

127.0.0.1:5500/index.html

☐ Aceptar Términos

! Selecciona esta casilla de verificación si quieres continuar

ENVÍOS

Ahora que ya disponemos de un repertorio suficiente para poder construir los formularios, lo único que nos queda, es colocar el botón de “Enviar”. Para ello vamos a utilizar el tipo “**submit**” en la etiqueta input de nuestro formulario. A esta etiqueta input podemos añadirle un “**value**” que es el texto de dicho botón. Si no especificamos el value, ubicará la palabra por defecto en función al lenguaje del navegador en el cual se encuentra.

```
<input type="submit" value="Enviar Datos">
```

Existen muchas otras etiquetas para formularios, pero es prácticamente imposible memorizarse todas de memoria. Pero existen sitios en las que se pueden consultar:

DOCUMENTACIÓN HTML

<https://developer.mozilla.org>

<https://www.w3schools.com/>

FIELDSET Y LEGEND

Hay veces que nos puede interesar comprimir una serie de inputs en una misma área, por eso existe la etiqueta “<fieldset>”. Esta etiqueta crea un borde sobre los inputs que se encuentren dentro de ella. Para ponerle un título a este fieldset se utiliza la etiqueta “<legend>”.

ETIQUETA	FUNCIÓN
<fieldset>	Caja Formularios
<legend>	Título para los fieldset

```
<form action="">
  <fieldset>
    <legend>Título del fieldset</legend>
    <label for="nombre">Nombre: </label>
    <input type="text" name="txtNombre" id="nombre">
  </fieldset>
</form>
```

Título del fieldset

Nombre:

SEMÁNTICA

SOBRE ESTRUCTURAS

La estructura es un aspecto fundamental en el desarrollo web. No solo porque nos ayuda a mantener un código limpio y estructurado, sino también por la importancia de la semántica, y es que, las etiquetas de estructura nos van a ayudar a agrupar parte de nuestro contenido, pero también serán de gran utilidad para los robots que quieran leer nuestra web y quieran saber qué contiene esta. Hay múltiples robots que van a acceder a nuestra web a diario, y van a juzgar el contenido, y a ellos, las etiquetas de estructura les ayudan a entender qué contiene tu página. Un ejemplo de robot sería el de búsqueda de Google, que lee el contenido de la página para saber qué contiene, y así poder mostrarla en su listado de opciones cuando alguien esté buscando alguna información de interés que contiene tu sitio. Si deseas que tu sitio tenga visibilidad debes ayudar al robot de Google con estructuras bien definidas.

Pero existen también otros robots, como por ejemplo el lector de contenido para personas discapacitadas. Hay gente con ciertas discapacidades relacionadas con la visión que navegan por internet, y, para poder navegar usan robots como el lector del navegador que les ayudan a entender qué se está mostrando en el monitor y les lee la información que solicitan.

Es muy importante agrupar las diferentes partes de nuestra página web en este tipo de etiquetas.

CABECERA, PRINCIPAL Y PIE

Vamos a empezar a darle un poco de orden al contenido que escribimos con las etiquetas “<header>”, “<main>” y “<footer>”.

ETIQUETA	FUNCIÓN
<header>	Cabecera de la página
<main>	Principal
<footer>	Pie de página

<header>

Cuando estemos construyendo nuestra página web, dentro del body, es un buen ejercicio pararse a pensar cuáles de mis contenidos van a colocarse en la parte superior de mi página como elementos destacados, o en un lateral como elemento destacado, como puede ser por ejemplo el menú de navegación. Esta parte sería el header.

</header>

<footer>

También tendríamos que preguntarnos qué vamos a colocar abajo del todo de la página, es decir, como pie de nuestra página, como puede ser por ejemplo el contenido de enlaces de refuerzo, el aviso legal, el copyright, y toda esta información. Todo esto lo vamos a agrupar dentro del footer.

</footer>

```
<main>
```

El resto del contenido, dentro de esta primera estructuración, lo escribiríamos en el main, que va a ser siempre el contenido principal.

```
</main>
```

Es muy importante que no vamos a referirnos solo dentro del body, sino dentro de cualquiera elemento. Imaginar que estamos diseñando una tarjeta con una información sobre un artículo en un blog. Pues esa tarjeta a su vez, puede tener header, puede tener main y puede tener footer.

También es importante destacar que toda esta estructuración no es obligatoria, somos nosotros como desarrolladores los que decidimos cómo vamos a colocar cada una de ellas. Por eso es importante tener la cabeza bien formada para saber, a base de experiencia, dónde colocar cada una de las etiquetas.

NAVEGACIÓN

```
<nav>
```

En nuestro sitio web, debemos colocar en diferentes lugares, menús a través de los cuales el usuario pueda navegar a través de las diferentes páginas que conforman nuestro sitio web. A nivel semántico, y sin importar donde coloquemos nuestro menú, este lo encerraremos siempre dentro de la etiqueta "nav".

```
</nav>
```

ETIQUETA	FUNCIÓN
<nav>	Menú navegación

No tiene ningún efecto, es simplemente a nivel de estructura. La estructura típica de un menú de navegación y que se consideraría bien estructurada sería:

```
<header>
  <nav>
    <ul>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
    </ul>
  </nav>
</header>
```

SECCIONES, ARTÍCULOS Y AUXILIARES

Para segmentar el main de nuestra página vamos a conocer 3 nuevas etiquetas de estructura que nos van a ayudar a continuar atomizando nuestro contenido.

ETIQUETA	FUNCIÓN
<code><section></code>	Sección temática
<code><article></code>	Artículo
<code><aside></code>	Contenido auxiliar

`<section>`

`</section>`

En esta etiqueta vamos a agrupar las diferentes secciones en las cuales se divide nuestro contenido principal. Cada una de las secciones suele tratar un tema diferente.

`<article>`

`</article>`

Esta etiqueta va a encerrar el contenido de un artículo dentro de una web. Dentro de este artículo se podrían colocar etiquetas de título, imágenes, etc.

`<aside>`

`</aside>`

Esta última etiqueta vamos a estar utilizándola como contenido secundario. Dentro de esta etiqueta vamos a ubicar toda esa información que está relacionada con la sección o con el artículo, o con el elemento que tenga cerca, pero cuyo material no forma parte de lo importante de esa sección o artículo. Podemos entenderla como un contenido auxiliar.

Un ejemplo en el que podemos implementar estas etiquetas es el siguiente:

```
<main>
  <section>
    <article>
      <h3>Titulo</h3>
      <img src="">
    </article>
    <article>
      <h3>Titulo 2</h3>
      <img src="">
    </article>
  </section>
  <aside>
    <h5>Productos relacionados</h5>
    <ul>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
      <li><a href=""></a></li>
    </ul>
  </aside>
</main>
```

DIVISORES

Puede llegar a ser necesario realizar otras subdivisiones más a parte de las ya vistas anteriormente, que no atienden a ninguno de los casos anteriores. Esto es algo común sobre todo para el CSS, porque a veces nos va a interesar partir este contenido en otras divisiones para así poder trabajarlo. Para ello vamos a ver dos tipos de división en nuestra estructura que podemos aplicar:

ETIQUETA	FUNCIÓN
<code><div></code>	Divisor en bloque
<code></code>	Divisor en línea

Su comportamiento es similar, la principal diferencia que van a contener es que `div` es una división en bloque, mientras que `span` es una división en línea.