

Guía rápida HTML

Contenido

Guía rápida HTML.....	1
Introducción	2
Anatomía de una etiqueta	3
Comillas dobles o comillas simples	4
Espacios en blanco	5
Caracteres de escape	5
Comentarios en el código.....	6
Contenidos de la cabecera	6
Fundamentos del texto HTML.....	9
Párrafos y títulos	9
Definir valor semántico, etiqueta	11
Creación de listas	11
Resaltar texto	12
Hipervínculos.....	13
Formato avanzado de texto	15
Estructura web	20
Tablas HTML	26
Conceptos avanzados en tablas	28
Formularios Web.....	30
Estructura de un formulario.....	33
Etiquetas de entrada	34
Atributos de la etiqueta <input>	44
Atributos de la etiqueta <form>.....	47
Otros controles, medidor y barra de progreso.	48
Etiqueta Medidor <meter>	48
Etiqueta Barra de progreso <progress>	49
Cajas de dialogo	49
Referencias	51
Herramientas.....	52

Introducción

HTML hace referencia a Hyper Text Markup Language o “lenguaje de marcas de hiper texto” y se emplea para la elaboración de páginas web. Una página web contiene una estructura básica compuesta por código HTML, texto, imágenes y otros elementos.

Existen múltiples versiones:

- 1989 Tim Berners-Lee inventa www
- 1991 Tim Berners-Lee inventa HTML
- 1995 el grupo de trabajo HTML define HTML 2.0
- 1997 W3C presenta la recomendación de HTML 3.2
- 1999 W3C presenta la recomendación de HTML 4.01
- 2000 W3C presenta la recomendación de XHTML 1.0
- 2008 WHATWG presenta primer borrador de HTML5
- 2012 WHATWG presenta estándar vivo de HTML5
- 2014 W3C presenta la recomendación de HTML5
- 2017 W3C presenta la recomendación de HTML5.1
- 2017 W3C presenta la recomendación de HTML5.2

El W3C o “World Wide Web Consortium” es un consorcio fundado y dirigido por Tim Berners-Lee que genera recomendaciones y estándares para la web.

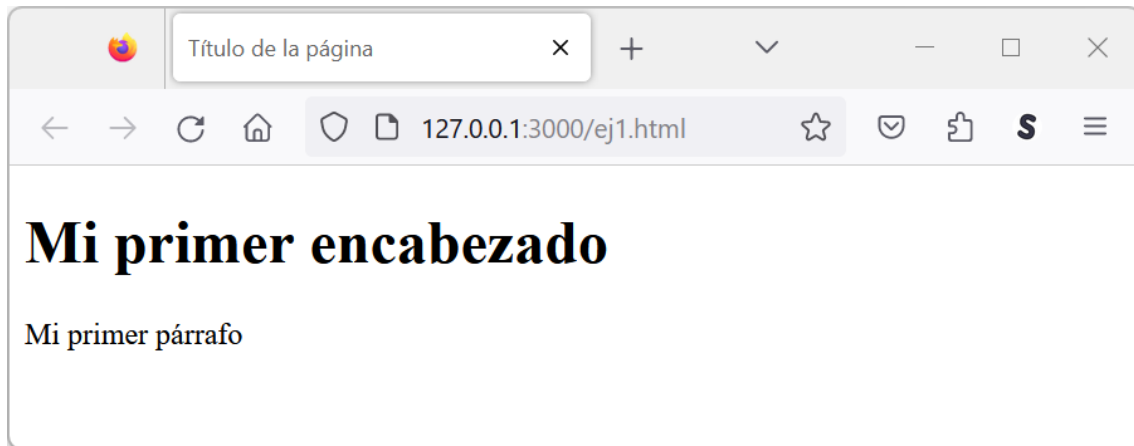
En esta guía vamos a emplear HTML5 lo que es necesario comprender es que esta última versión incluye la mayoría de los elementos de las versiones previas (ojo, algunos elementos se han declarado obsoletos y ya no deberíamos emplearlos).

Hola mundo

Crea un nuevo documento con la extensión “.html” y escribe el siguiente código.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Título de la página</title>
  </head>
  <body>
    <h1>Mi primer encabezado</h1>
    <p>Mi primer párrafo</p>
  </body>
</html>
```

Una vez guardado, ábrelo con un navegador e identifica la posición de cada etiqueta.



La etiqueta `<!DOCTYPE html>` indica que el documento es HTML5, esta etiqueta siempre es la primera del documento.

Seguido tenemos la etiqueta `<html>` la cual delimita el contenido del documento HTML, fíjate que tiene una etiqueta de cierre `</html>`.

Dentro de la etiqueta `<html>` tenemos dos elementos hijos, la cabecera `<head>` y el cuerpo `<body>`.

La cabecera aparte de indicar el título de la página nos va a permitir definir otros metadatos de la página, volveremos más adelante a ella.

El cuerpo contiene todos los elementos visibles de la página pudiéndose anidar o poner en paralelo para realizar composiciones más o menos elaboradas.

Anatomía de una etiqueta

Una etiqueta HTML tienes las siguientes partes

- **Etiqueta de apertura.** Consiste en el nombre del elemento encerrado entre < >
- **Contenido.** Lo que se va a mostrar, según el caso puede ser texto plano o más etiquetas HTML.
- **Etiqueta de cierre.** Lo mismo que la etiqueta de apertura, pero incluye la barra de dividir </ >

Por ejemplo

The diagram illustrates the structure of an HTML element. It starts with 'Apertura' (Opening) represented by a horizontal line, followed by 'Cierre' (Closing) represented by another horizontal line. Below these is the text '<p>Mi primer párrafo</p>', where the opening and closing tags are highlighted in red. A horizontal line below the text is labeled 'Contenido' (Content). Finally, a long horizontal line at the bottom is labeled 'Elemento' (Element), encompassing the entire structure.

Hay que tener en cuenta que algunas etiquetas pueden no tener contenido, en estos casos la propia etiqueta de apertura incluye la / para indicar la ausencia de etiqueta de cierre. Por ejemplo, las imágenes.

```

```

Otra característica de las etiquetas HTML es que pueden incluir atributos, un atributo añade información que no aparece en el contenido. Por ejemplo, el alto, el ancho, el estilo, la descripción alternativa, etc...

Atributo

```
<p class="nota-importante">Mi primer párrafo</p>
```

Un atributo consta se compone de los siguientes partes:

- **Nombre del atributo.** Seguido de un "=" sin espacios.
- **Valor del atributo.** El valor entre comillas dobles "". Es posible que un atributo admita más de un valor, en este caso los distintos valores se separan mediante espacios " ".

Existe un tipo especial de atributo que es el atributo binario, es decir aquel atributo con un valor si o no. En este caso la notación recomendada es que el valor del atributo sea el propio nombre del atributo para el valor cierto. Por ejemplo.

```
<input type="text" disabled="disabled" />
```

En estos casos también es aceptable omitir el apartado del valor. Por ejemplo.

```
<input type="text" disabled />
```

Comillas dobles o comillas simples

Te habrás dado cuenta que siempre que le damos valor a un atributo empleamos las comillas dobles "". El estándar de HTML admite indistintamente las comillas dobles o las comillas simples ' '.

```
<a href='https://www.example.com'>Un enlace de ejemplo.</a>  
<a href="https://www.example.com">Un enlace de ejemplo.</a>
```

Lo que no se admite es mezclar comillas dobles y simples.

```
<a href="https://www.example.com">Un enlace de ejemplo.</a>
```

El punto importante es dentro de las comillas dobles puedo emplear las comillas sencillas y viceversa.

```
<a href="https://www.example.com" alt="Esto es un 'ejemplo' de enlace">Un  
enlace de ejemplo.</a>
```

Si quisiéramos mostrar en el contenido las comillas dobles debemos emplear la secuencia de escape "

```
<a href="https://www.example.com" alt="Esto es un &quot;ejemplo&quot; de enlace">Un enlace de ejemplo.</a>
```

Mi recomendación. **Emplear siempre en el HTML las comillas dobles y en javascript las comillas simples**, de esta manera cuando generéis marcado HTML dinámicamente no tendréis problemas al mezclar en la misma cadena comilla doble y comilla simple.

Espacios en blanco

Una peculiaridad de HTML es que suprime los espacios en blanco y saltos de línea adicionales. Los siguientes fragmentos de código general la misma salida en el navegador.

```
<p>El lenguaje HTML tiene sus cosas.</p>
<p>El      lenguaje      HTML
tiene sus

cosas.</p>
```

Muestra en el navegador



El lenguaje HTML tiene sus cosas.

El lenguaje HTML tiene sus cosas.

En los casos en los que necesitemos representar espacios o saltos de línea en el contenido deberemos emplear código HTML para indicarlos, ` ` para el espacio en blanco y `
` para el salto de línea.

Caracteres de escape

En HTML se reservan ciertos caracteres para la definición del propio lenguaje, en los casos en que necesitemos incluir alguno de estos caracteres en nuestro contenido deberemos emplear una codificación especial.

Los siguientes caracteres son de uso frecuente por lo que conviene memorizarlos:

Literal	Equivalente HTML
<	<
>	>
"	"
'	'
&	&
X (carácter de cierre ventana)	×

Existen muchos más caracteres de escape, se puede consultar la lista completa en el siguiente enlace https://en.wikipedia.org/wiki/List_of_XML_and_HTML_character_entity_references

Comentarios en el código

Para incluir un comentario dentro de código HTML debemos emplear los marcadores `<!--` y `-->`. Por ejemplo.

```
<p>Esto es contenido visible</p>
<!-- <p>Este párrafo es un comentario</p> -->
<p>Esto también es contenido visible</p>
```

Contenidos de la cabecera

La cabecera incluye los elementos de la página que no se muestran en el navegador. A modo de resumen incluye el título, enlaces a ficheros CSS, enlaces a ficheros JS, el enlace al icono de la página y metadatos (autor, descripción, tipo de seguimiento por los navegadores, etc...).

Veamos el contenido de una web sencilla.

```
<!DOCTYPE html>
<html lang="es">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Mi página de pruebas</title>
</head>

<body>
  <p>Contenido de mi página de pruebas</p>
</body>

</html>
```

Lo primero, y no es parte del HEAD, la etiqueta `<html lang="es">` el atributo "Lang" le dice al navegador que idioma por defecto emplea la página facilitando su procesamiento.

Al entrar en el head encontramos varios metadatos que DEBERÍAS INCLUIR SIEMPRE en tus páginas. Veamos cuál es su función:

- `<meta charset="UTF-8">`, permite indicar el juego de caracteres empleado en la codificación del contenido. UTF-8 es un estándar que te permitirá representar la mayoría de los lenguajes del mundo.
 - En caso de indicar un juego de caracteres incorrectos verás cuadraditos en el navegador en vez del texto esperado.

- NOTA: El navegador Chrome corrige automáticamente este atributo cuando detecta problemas.
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`, ajusta automáticamente el tamaño de la página al tamaño del navegador. Esto soluciona muchos problemas de accesibilidad.

La etiqueta meta con los atributos “name” y “content” permiten definir metadatos en formato clave valor, “name” define el nombre del metadato o clave y “content” su valor asociado. Veamos las combinaciones más frecuentes:

Podemos indicar el autor de la página

```
<meta name="author" content="Alejandro López" />
```

Podemos indicar una descripción de los contenidos de la página

```
<meta name="description" content="Esta página incluye maquetado html de prueba" />
```

Podemos indicar como queremos que nos rastreen los buscadores

```
<meta name="robots" content="index, follow">
```

En este caso podemos tomar múltiples valores:

- index, la página debe indexarse.
- noindex, la página no debe indexarse.
- follow, deben seguirse los enlaces en la página.
- nofollow, no deben seguirse los enlaces en la página.
- none, lo mismo que poner noindex y nofollow.
- noarchive, la página no debe cachearse.
- Etc....

RECUERDA que, aunque nosotros demos indicaciones de cómo deben tratarnos los buscadores es muy posible que algún buscador haga lo que quiera.

Palabras relevantes por las que debo ser encontrado

```
<meta name="keywords" content="ejemplo metadatos, aprender, html">
```

Algunos buscadores ya no usan esta etiqueta porque se le ha dado un uso ilícito para atraer usuarios al sitio por lo que se recomienda no incluirla.

La etiqueta `<title>Mi página de pruebas</title>` contiene el título a mostrar en la barra de título de la ventana del navegador, este texto también es el que se emplea al añadir la página a los marcadores.

Definir un icono para la página

Las páginas pueden definir un icono o en este contexto “favicon” que se mostrará en la barra de navegación y en la lista de marcadores. Aunque el tipo recomendado para definir el icono es “.ico” se pueden emplear otros formatos como “png”.

```
<link rel="icon" href="favicon.ico" type="image/x-icon" />
```

```
<link rel="icon" href="favicon-96x96.png" type="image/png" />
```

Referenciar archivos CSS y JavaScript

Aunque podamos incluir nuestros estilos y scripts directamente en nuestra página por diversos motivos es común dividir el código de nuestra página en 3 partes:

- El contenido HTML, en un fichero .html
- El contenido de estilos CSS, en un fichero .css
- El contenido de los guiones JavaScript, en un fichero .js

Los motivos principales por los que se hace esto es mejorar la legibilidad, reutilizar código y optimizar la carga entre otros. Dicho esto, descubrirás que es necesario conocer tanto la codificación inline (todos los contenidos en el propio html) como el uso de elementos externos.

Veamos un ejemplo.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Ejemplo ficheros externos</title>
  <link rel="stylesheet" href="mi-hoja-estilos.css" />
  <script src="mi-codigo-javascript.js"></script>
</head>
<body>
  <p>Este párrafo debería mostrarse con estilo</p>
</body>
</html>
```

El fichero externo "mi-hoja-estilos.css" contiene el siguiente estilo.

```
p {
  background-color: aquamarine;
}
```

Para cargar hojas de estilo externas empleamos la etiqueta `link`, indicamos que es una hoja de estilo mediante `rel="stylesheet"` y la ruta al fichero mediante `href="mi-hoja-estilos.css"`.

```
<link rel="stylesheet" href="mi-hoja-estilos.css" />
```

Para cargar un script externo empleamos la etiqueta `script` con la ruta al fichero mediante `src="mi-codigo-javascript.js"`.


```
<script src="mi-codigo-javascript.js"></script>
```

NOTA: la etiqueta “script” nunca puede ser un elemento vacío por lo que SIEMPRE va acompañada de una etiqueta de cierres “</script>”

Indicar que también se pueden definir estilos y bloques de código JavaScript en el propio head mediante las etiquetas <style> y <script>. Por ejemplo.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <title>Ejemplo ficheros externos</title>
  <link rel="stylesheet" href="mi-hoja-estilos.css" />

  <style>
    p {
      font-size: large;
    }
  </style>

  <script src="mi-codigo-javascript.js" defer></script>
  <script>
    alert("¿cuándo me ejecuto?")
  </script>
</head>
<body>
  <p>Este párrafo debería mostrarse con estilo</p>
</body>
</html>
```

Fundamentos del texto HTML

Es importante que nuestro contenido no sea un texto sin más y que presente una estructura que facilite su lectura. Piensa en un periódico impreso o en un libro, tenemos títulos, párrafos, listas de información, etc...

A continuación, vamos a presentar los elementos básicos para el formato del texto.

Párrafos y títulos

Los títulos permiten indicar apartados o capítulos en una historia, HTML permite definir hasta 6 tipos de títulos mediante las etiquetas <h1>, <h2>, <h3>, <h4>, <h5> y <h6>. Siendo <h1> el título más relevante y el resto de etiquetas sub títulos siendo el menos importante <h6>.

Junto con los títulos empleamos párrafos. Un párrafo se representa con la etiqueta `<p>`. Los párrafos nos van a permitir agrupar varias frases en un bloque, el navegador automáticamente diferenciará los párrafos incluyendo un espacio entre ellos.

Por ejemplo:

```
<h1>Esto es un título h1</h1>
<h2>Esto es un subtítulo h2</h2>
<p>Vemos que podemos dar formato fácilmente a un texto.</p>
<p>Es conveniente que todas las páginas tengan un título h1.</p>
<h2>Otro subtítulo h2</h2>
<h3>Esto es un subtítulo h3</h3>
<p>Por lo general no deberíamos emplear más de 3 tipos de títulos en
la página</p>
<p>Y también deberíamos respetar el orden jerárquico desde h1 hasta
h6</p>
```

El código anterior genera la siguiente salida.

Esto es un título h1

Esto es un subtítulo h2

Vemos que podemos dar formato fácilmente a un texto.

Es conveniente que todas las páginas tengan un título h1.

Otro subtítulo h2

Esto es un subtítulo h3

Por lo general no deberíamos emplear más de 3 tipos de títulos en la página

Y también deberíamos respetar el orden jerárquico desde h1 hasta h6

Al estructurar tu página recuerda lo siguiente:

- La estructura facilita encontrar la información en la página, con lo que se retiene al usuario más tiempo.
- Los motores de búsqueda indexan los títulos por lo que emplear títulos mejoras el posicionamiento SEO.
- Los usuarios que emplean lectores de pantalla usan los títulos para navegar rápidamente por el contenido.

- La estructura al añadir etiquetas que envuelven el contenido facilita el maquetado CSS y la interacción con JavaScript.

Definir valor semántico, etiqueta ``.

Se puede dar el caso de que necesitemos dar un valor semántico a un texto dentro de nuestro contexto, por ejemplo, marcar un mensaje como error, o resaltar una información sobre el resto de contenidos.

En estos casos empleamos la etiqueta ``, que por defecto no aporta ningún significado ni modifica el aspecto visual del elemento envuelto. Su utilidad radica en que admite propiedades para modificar el aspecto del contenido.

Esta etiqueta tendrá sentido cuando veamos CSS, de momento veamos un ejemplo en el que hacemos que un texto se parezca a un título sin serlo.

```
<h1>Esto es un título</h1>
<span style="font-size: 32px; font-weight: bold; margin: 21px 0; display: block;"> Esto NO es un título</span>
```

Creación de listas

Una lista nos permite representar un conjunto de datos, estos datos pueden estar ordenados o no ordenados.

Una lista no ordenada muestra una viñeta delante de cada elemento de la lista. En cambio, una lista ordenada muestra un número incremental delante de cada elemento de la lista.

Las etiquetas que nos permiten definir una lista son:

- ``. "unordered list", define una lista NO ordenada.
- ``. "ordered list", define una lista ordenada.
- ``. "list item", cada elemento de la lista.

Por ejemplo, el siguiente código muestra la misma lista de manera no ordenada y ordenada.

```
<ul>
  <li>HTML</li>
  <li>CSS</li>
  <li>JS</li>
</ul>

<ol>
  <li>HTML</li>
  <li>CSS</li>
  <li>JS</li>
</ol>
```

Genera la siguiente salida.

- HTML
- CSS
- JS

1. HTML
2. CSS
3. JS

Ten en cuenta que las listas se pueden anidar unas dentro de otras para formar la jerarquía deseada.

Resaltar texto

A la hora de escribir empleamos diversos artificios visuales para remarcar o enfatizar el texto, estos artificios son el texto en negrita, el subrayado y el texto en itálica o cursiva.

En HTML actual tenemos 3 etiquetas para resaltar el texto: ``, `` y `<mark>`. Estas etiquetas tienen valor semántico, esto significa que los lectores de pantalla cambiarán el tono de voz al leer estos contenidos.

- La etiqueta `` enfatiza el texto. Visualmente el navegador muestra el contenido en cursiva, este comportamiento puede ser modificado a través de CSS.
- La etiqueta `` pone en negrita (aunque la descripción oficial dice que aporta valor semántico resaltando el contenido...).
- La etiqueta `<mark>` remarca el texto en amarillo al estilo de un rotulador subrayador.
- La etiqueta `<small>` empequeñece un texto, se emplea para copyright o comentarios en el contenido (Se puede conseguir un efecto similar con CSS).

Por ejemplo

```
<p>Este es un texto de ejemplo sin ningún adorno</p>
<p>Este es un texto de ejemplo <em>con énfasis</em></p>
<p>Este es un texto de ejemplo <strong>en negrita</strong></p>
<p>Este es un texto de ejemplo <mark>rotulado</mark></p>
<p>Este es un texto de ejemplo <small>empequeñecido</small></p>
```

Genera la siguiente salida.

Este es un texto de ejemplo sin ningún adorno

Este es un texto de ejemplo *con énfasis*

Este es un texto de ejemplo **en negrita**

Este es un texto de ejemplo rotulado

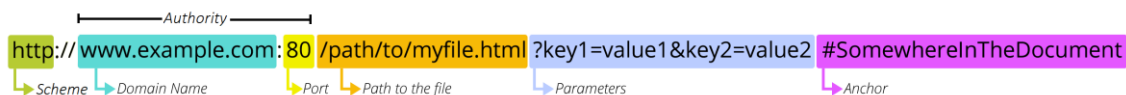
Este es un texto de ejemplo empequeñecido

Existen etiquetas para indicar la negrita ``, la cursiva `<i>` y el subrayado `<u>` pero su uso no se recomienda por no aportar valor semántico. Su origen está en la época en la que las posibilidades del CSS eran limitadas siendo posible actualmente su sustitución por otro marcado con valor semántico. De cualquier manera, es importante remarcar que el subrayado confunde a muchos usuarios haciéndoles pensar en un enlace por lo que no deberías subrayar contenido.

Hipervínculos

Los hipervínculos o enlaces son un elemento esencial en la web, nos permiten navegar entre los distintos documentos que forman la web.

La dirección web a la que apunta un enlace se denomina URL “Uniform Resource Locator”. En su formato más extenso una URL tiene los siguientes apartados:



- Esquema. Indica la manera en la que se va a intercambiar la información. Normalmente http, https, mailto, y alguno más para usos más específicos.
- Nombre de dominio. Indica el servidor al que accedemos, es equivalente a poner la IP de dicho servidor.
- Puerto. Indica el puerto en el servidor por el que se establecerá la conexión. Los puertos para HTTP (80) y HTTPS (443) se pueden omitir, el resto hay que indicarlos obligatoriamente.
- Ruta al recurso. Indica la ruta dentro del servidor a la que queremos acceder. Al principio de la WEB representaba una ruta a un fichero físico, actualmente lo común es que sea una abstracción a un contenido dinámico.
- Parámetros. Indica uno o varios parámetros con el formato clave valor. Comienzan por el símbolo “?” y después las tuplas de clave valor separadas por el símbolo “&”. Por ejemplo: “?clave1=valor1&clave2=valor2&clave3=valor3”.
- Ancla. Indica una referencia dentro de la página actual, es decir, me permite navegar dentro de la página que se muestra sin la necesidad de recargarla.

NOTA: Ten en cuenta que un enlace puede hacer referencia a un archivo con una extensión cualquiera. En este caso cuando el navegador trata de abrirlo si el formato es soportado el

navegador lo muestra, en caso contrario lo descarga y le deja a la aplicación por defecto del sistema que lo abra.

Un hipervínculo consta de la etiqueta `<a>` el atributo `"href"` con la URL a la que apunta y como contenido el texto o código HTML a mostrar en el enlace, nadie te impide mostrar una imagen o un div como un enlace.

Un ejemplo de enlace básico.

```
<p>Aquí podrás ver un  
  <a href="https://www.w3schools.com/html/default.asp">tutorial  
HTML</a>  
</p>
```

Muestra lo siguiente:

Aquí podrás ver un [tutorial HTML](https://www.w3schools.com/html/default.asp)

Puedes añadir el atributo `"title"` para incluir un comentario sobre el contenido del enlace, este comentario sólo se mostrará si el usuario deja el ratón un tiempo sobre el enlace.

```
<p>Aquí podrás ver un  
  <a href="https://www.w3schools.com/html/default.asp"  
    title="Encontraras una guía detallada sobre HTML">tutorial HTML</a>  
</p>
```

Tipos de rutas

Veamos en detalle los distintos tipos de rutas.

- Rutas absolutas.
- Rutas relativas.
 - A la raíz del servidor
 - A la carpeta actual del servidor
- A un fragmento dentro del documento (ancla).

Para ver cada una de ellas en detalle supongamos la siguiente estructura en mi sitio web

- Padre.html
- Mi sitio web\index.html
- Mi sitio web\documentos\listado.html
- Mi sitio web\ejemplo.html

Ruta absoluta.

Este formato indica la dirección completa de un recurso. Su uso es recomendado cuando referenciamos documentos externos a nuestro sitio. Por ejemplo:

```
<a href="https://www.google.com">Ruta absoluta</a>
```

Ruta relativa.

Este es el formato que debes emplear por defecto ya que en caso de cambiar la dirección de nuestro servidor no deberemos modificar el código.

Las rutas relativas presentan varias notaciones que debemos conocer.

La ruta relativa a la raíz del servidor. Esta ruta se indica comenzando con el carácter “/”

```
<a href="/index.html">Ruta relativa, raíz del sitio web</a>
```

La ruta relativa a la carpeta actual del servidor. A diferencia de la relativa al servidor se indica directamente el recurso o ruta al recurso desde la ubicación actual.

```
<a href="documentos/listado.html">Ruta relativa, a la carpeta actual</a>
```

Fíjate que en este caso podemos referenciar la carpeta padre mediante “..”.

```
<a href="../padre.html">Ruta relativa, al padre de la carpeta actual</a>
```

Ancla.

Las anclas nos permiten navegar dentro de la página actual, es decir, son marcadores dentro de la misma página. Para poder usarlos debemos indicar el marcador mediante un identificador.

```
<h2 id="inicio">Ejemplos de rutas</h2>
```

...

...

```
<a href="#inicio">Anclar, volver al inicio</a>
```

Para ampliar la etiqueta `<a>`:

- Propiedad “**target**”, igual que el atributo “target” de la etiqueta `<form>`, puedes consultarlo en el apartado de “Atributos de la etiqueta `<form>`”.
- Propiedad “**download**”, permite descargar un archivo sin intentar abrirlo.

Formato avanzado de texto

Lista de definiciones

Se pueden crear listas de definiciones o simplemente una definición. Las etiquetas son:

- `<dl>`, “Description List” o descripción de lista.
- `<dt>`, “Description Term” o termino de descripción.

- `<dd>`, “Description” o descripción.

Por ejemplo

```
<dl>
  <dt>HTML</dt>
  <dd>HyperText Markup Language, el lenguaje estándar para crear
  páginas web.</dd>

  <dt>CSS</dt>
  <dd>Cascading Style Sheets, se utiliza para dar estilo y formato a
  las páginas web.</dd>

  <dt>JavaScript</dt>
  <dd>Un lenguaje de programación utilizado para crear interactividad
  en sitios web.</dd>
</dl>
```

Genera la siguiente salida.

```
HTML
  HyperText Markup Language, el lenguaje estándar para crear páginas web.
CSS
  Cascading Style Sheets, se utiliza para dar estilo y formato a las páginas web.
JavaScript
  Un lenguaje de programación utilizado para crear interactividad en sitios web.
```

Citar texto

Podemos resaltar un comentario mediante una cita, las citas se resaltan sobre el resto del texto en un nuevo bloque.

```
<p>El profesor dijo:</p>
<blockquote cite="https://www.w3schools.com/html/">
  <p>
    El HTML tiene muchos recursos de aprendizaje.
  </p>
</blockquote>
```

Genera la siguiente salida.

El atributo `cite` permite indicar una referencia, aunque en la realidad no los navegadores no hacen nada con él.

El profesor dijo:

El HTML tiene muchos recursos de aprendizaje.

Citar texto en línea

Podemos también indicar una cita en línea, en este caso la cita no se separa en un nuevo bloque y se muestra seguido de manera entrecomillada.

```
<p>El profesor dijo:
  <q cite="https://www.w3schools.com/html/">
    El HTML tiene muchos recursos de aprendizaje.
  </q>
</p>
```

Genera la siguiente salida.

El profesor dijo: “ El HTML tiene muchos recursos de aprendizaje. ”

Abreviaturas

Las abreviaturas se emplean para incluir la explicación de un acrónimo. La etiqueta `<abbr>` envuelve al acrónimo o abreviatura y propiedad su `title` descripción. La descripción se mostrará automáticamente si se detiene un tiempo el ratón sobre la abreviatura.

```
<p>
  Usamos <abbr title="Lenguaje de marcado de hipertexto">HTML</abbr>
para estructurar nuestros documentos web.
</p>
```

Genera la siguiente salida.

Usamos HTML para estructurar nuestros documentos web.

Dirección de contacto

La etiqueta `<address>` envuelve los datos de un contacto. Podemos incluir en su interior lo que queramos. Por ejemplo:

```
<address>
  <p>
    Chris Mills<br />
    Manchester<br />
    The Grim North<br />
    Reino Unido
  </p>

  <ul>
    <li>Tel: 01234 567 890</li>
    <li>Email: me@grim-north.co.uk</li>
  </ul>
</address>
```

Genera la salida.

Chris Mills
Manchester
The Grim North
Reino Unido

- Tel: 01234 567 890
- Email: me@grim-north.co.uk

Superíndice y subíndice

En las fórmulas o fechas suele ser necesario emplear superíndices `<sup>` y subíndices `<sub>`.
Por ejemplo:

```
<p>Nací el 25<sup>th</sup> de mayo de 2001.</p>
<p>
  La fórmula química de la cafeína es
  C<sub>8</sub>H<sub>10</sub>N<sub>4</sub>O<sub>2</sub>.
</p>
<p>If x<sup>2</sup> es 9, x debe ser igual 3 o -3.</p>
```

Genera la salida.

Nací el 25th de mayo de 2001.

La fórmula química de la cafeína es C₈H₁₀N₄O₂.

If x² es 9, x debe ser igual 3 o -3.

Representar código informático

Para poder presentar código fuente e instrucciones de entrada / salida en la web disponemos de etiquetas diversas etiquetas que deberemos combinar.

- Etiqueta `<pre>`, muestra el texto respetando los espacios y los saltos de línea. Este es el truco para poder mostrar el código formateado.
- Etiqueta `<code>`, marca fragmentos de código informático.
- Etiqueta `<var>`, para indicar el nombre de una variable dentro de un texto explicativo, si usas `<code>` no te hace falta.
- Etiqueta `<kbd>`, para resaltar un texto como entrada o combinación de teclado.
- Etiqueta `<samp>`, para resaltar un texto como salida de consola.

Por ejemplo.

```
<pre><code>var para = document.querySelector('p');

  para.onclick = function() {
    alert('¡Guau!, ¡deja de apretar!');
  }</code></pre>
```

<p>

En el ejemplo de JavaScript anterior, `<var>para</var>` representa un elemento de párrafo.

</p>

<p>

Selecciona todo el texto con `<kbd>Ctrl</kbd>/<kbd>Cmd</kbd> + <kbd>A</kbd>`.

</p>

```
<pre>$ <kbd>ping mozilla.org</kbd>
    <samp>PING mozilla.org (63.245.215.20): 56 bytes de datos
    64 bytes de 63.245.215.20: icmp_seq=0 ttl=40
    time=158.233ms</samp></pre>
```

Genera la salida.

```
var para = document.querySelector('p');

para.onclick = function() {
    alert('¡Guau!, ¡deja de apretar!');
}
```

En el ejemplo de JavaScript anterior, *para* representa un elemento de párrafo.

Selecciona todo el texto con Ctrl/Cmd + A.

```
$ ping mozilla.org
  PING mozilla.org (63.245.215.20): 56 bytes de datos
  64 bytes de 63.245.215.20: icmp_seq=0 ttl=40 time=158.233 ms
```

Mostrar fechas y horas

Uno de los problemas de la web es que a nivel mundial no está clara la hora de referencia y/o su formato. Para solucionar esto la etiqueta `<time>` separa la fecha mostrada al usuario de fecha legible por la máquina. El atributo “datetime” recoge el valor para la máquina.

```
<!-- Fecha simple estándar -->
<time datetime="2016-01-20">20 Enero 2016</time><br/>
<!-- Solo año y mes -->
<time datetime="2016-01">Enero 2016</time><br/>
<!-- Solo mes y día -->
<time datetime="01-20">20 Enero 2016</time><br/>
<!-- Solo tiempo, horas y minutos -->
<time datetime="19:30">19:30</time><br/>
<!-- ¡También puedes hacer segundos y milisegundos! -->
<time datetime="19:30:01.856">19:30:01.856</time><br/>
<!-- Fecha y hora -->
<time datetime="2016-01-20T19:30">7.30pm, 20 Enero 2016</time><br/>
<!-- Fecha y hora con desplazamiento de zona horaria -->
<time datetime="2016-01-20T19:30+01:00">7.30pm, 20 Enero 2016 es 8.30pm
en Francia</time><br/>
```

Genera la siguiente salida.

```
20 Enero 2016
Enero 2016
20 Enero 2016
19:30
19:30:01.856
7.30pm, 20 Enero 2016
7.30pm, 20 Enero 2016 es 8.30pm en Francia
```

Estructura web

Hasta este punto hemos visto como formatear el texto o mejor dicho el contenido de nuestras páginas, pero una página web además de contenido tiene una estructura que le aporta consistencia y usabilidad al sitio. A continuación, vamos a ver los elementos principales que dan estructura a un sitio web. Evidentemente no todos los sitios tienen las mismas necesidades ni la misma estructura, por lo que es tu obligación como desarrollador seleccionar y configurar los elementos que definan esta estructura.

Elementos estructurales.

- **Encabezado**, en la parte superior, normalmente permanece invariable mientras navegas y suele contener un logotipo y un gran título.
- **Barra de navegación**, contiene los enlaces a las secciones del sitio, normalmente se compone de botones, enlaces o pestañas. Al igual que el encabezado suele permanecer invariable mientras se navega.
 - NOTA: Aunque muchos diseñadores lo incluyen en el encabezado es recomendable que sea un elemento separado ya que de esta manera se facilita la accesibilidad en los lectores de pantalla.
- **Contenido principal**, incluye el contenido de mi sitio web. Esta sección es la que debe cambiar al navegar.
- **Barra lateral**, suele contener información adicional relacionada con el contenido principal. Como, por ejemplo, información del autor del contenido o enlaces relacionados. Otro uso frecuente es como menú de navegación secundario.
- **Pie de página**, es la parte inferior de la página, que generalmente contiene el copyright y la información de contacto. También es el sitio en el que colocar información común que no tiene la relevancia suficiente como para ser colocada en el encabezado. Otro uso común del pie es el posicionamiento SEO incluyendo accesos directos al contenido más popular.



Las etiquetas que establecen estas secciones son:

- Encabezado: `<header>`.
- Menú de navegación: `<nav>`.
- Contenido principal: `<main>`, con varias subsecciones (además de la barra lateral) representadas por los elementos `<article>`, `<section>`, y `<div>`.
- Barra lateral: `<aside>`, a menudo colocada dentro de `<main>`.
- Pie de página: `<footer>`.

Dedica un rato a comprender el código que ha generado la página de ejemplo.

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8" />

    <title>El título de mi página</title>
    <link
      href="https://fonts.googleapis.com/css?family=Open+Sans+Condensed:300|Sonsie+One"
      rel="stylesheet"
      type="text/css" />
    <link rel="stylesheet" href="style.css" />

  </head>

  <body>
    <!-- Aquí empieza el encabezado principal que se mantendrá en todas
    las páginas del sitio web -->
```

```

<header>
  <h1>Encabezado</h1>
</header>

<nav>
  <ul>
    <li><a href="#">Inicio</a></li>
    <li><a href="#">Nuestro equipo</a></li>
    <li><a href="#">Proyectos</a></li>
    <li><a href="#">Contacto</a></li>
  </ul>

  <!-- Un formulario de búsqueda es una forma no-lineal de hacer
búsquedas en un sitio web. -->
  <form>
    <input type="search" name="q" placeholder="Buscar" />
    <input type="submit" value="¡Vamos!" />
  </form>
</nav>

<!-- Aquí está el contenido principal de nuestra página -->
<main>
  <!-- Contiene un artículo -->
  <article>
    <h2>Título del artículo</h2>

    <p>
      Lorem ipsum dolor sit amet, consectetur adipisicing elit. Donec
a diam
      lectus. Set sit amet ipsum mauris. Maecenas congue ligula as
quam
      viverra nec consectetur ant hendrerit. Donec et mollis dolor.
Praesent
      et diam eget libero egestas mattis sit amet vitae augue. Nam
tincidunt
      congue enim, ut porta lorem lacinia consectetur.
    </p>

    <h3>Subsección</h3>

    <p>
      Donec ut librero sed accu vehicula ultricies a non tortor.
Lorem ipsum
      dolor sit amet, consectetur adipisicing elit. Aenean ut gravida
lorem.
      Ut turpis felis, pulvinar a semper sed, adipiscing id dolor.
    </p>

    <p>

```

Pelientesque auctor nisi id magna consequat sagittis. Curabitur dapibus, enim sit amet elit pharetra tincidunt feugiat nist imperdiet.

Ut convallis libero in urna ultrices accumsan. Donec sed odio eros.

</p>

<h3>Otra subsección</h3>

<p>

Cum Donec viverra mi quis quam pulvinar at malesuada arcu rhoncus.

soclis natoque penatibus et manis dis parturient montes, nascetur

ridiculus mus. In rutrum accumsan ultricies. Mauris vitae nisi at sem

facilisis semper ac in est.

</p>

<p>

Vivamus fermentum semper porta. Nunc diam velit, adipscing ut tristique vitae sagittis vel odio. Maecenas convallis ullamcorper

ultricies. Curabitur ornare, ligula semper consectetur sagittis, nisi

diam iaculis velit, is fringille sem nunc vet mi.

</p>

</article>

<!-- el contenido aparte también se puede anidar dentro del contenido principal -->

<aside>

<h2>Relacionado</h2>

Oh, me gusta estar junto al mar

Oh, me gusta estar junto al mar

Aunque en el norte de Inglaterra

Nunca deja de llover

Oh, bueno...

</aside>

</main>

<!-- Y aquí está nuestro pie de página principal que se utiliza en todas las páginas de nuestro sitio web -->

<footer>

<p>©Copyright 2050 de nadie. Todos los derechos revertidos.</p>

</footer>

```
</body>
</html>
```

Puedes encontrar la hoja de estilos “style.css” en la URL:

https://github.com/mdn/learning-area/blob/main/html/introduction-to-html/document_and_website_structure/style.css

Recomendaciones sobre las etiquetas de diseño.

- `<main>`, debe aparecer una única vez dentro del `<body>`. Es mejor que no lo anidemos con otros elementos.
- `<article>`, encuadra un bloque de contenido que tiene sentido por sí mismo. Piensa que en el artículo como un elemento que podríamos sacar de la página y al insertarlo en otra se mostraría correctamente.
- `<section>`, es parecido al `<article>` pero se emplea como contenedor artículos. Es buena práctica incluir un encabezado en la sección. Por ejemplo, en un periódico podríamos tener secciones de noticias locales, nacionales y deportes, y dentro de cada sección sus artículos.
- `<aside>`, incluye información relacionada con el contenido principal.
- `<header>`, representa el encabezado principal del sitio cuando es hijo del elemento `<body>`. También se puede emplear como encabezado de `<section>` y `<article>` en vez de un encabezado común.
- `<nav>`, contiene los enlaces principales de navegación del sitio.
- `<footer>`, representa el contenido final de una página.

Envolturas no semánticas

Como ya hemos comentado las etiquetas `` y `<div>` permiten definir contenedores que por defecto no aportan valor semántico. La gracia de estas etiquetas es que vayan siempre acompañadas de un atributo “class” para darles valor semántico.

La etiqueta `` es un elemento en línea, en el siguiente ejemplo resaltamos contenido relevante para el “editor” de una obra de teatro.

```
<p>
  El rey volvió ebrio a su habitación alrededor de la 01:00, y sin duda
  la cerveza no le ayudaba cuando cruzó tambaleante la puerta
  <span class="editor-note">[nota del editor: en este instante de la
  representación, deberían atenuarse las luces]</span>.
</p>
```

La etiqueta `<div>` es un elemento de bloque, que se emplea para dar valor semántico cuando no tenemos otro elemento específico. Por ejemplo, imagina que tenemos una tienda online y queremos representar un carrito de la compra.


```

<div class="shopping-cart">
  <h2>Carrito de compras</h2>
  <ul>
    <li>
      <p>
        <a href=""><strong>Pendientes de plata</strong></a>:
$99.95.
      </p>
      
    </li>
    <li>...</li>
  </ul>
  <p>Importe total: $237.89</p>
</div>

```

Este no es un elemento lateral porque no necesariamente está relacionado con el contenido principal (lo podrías ver desde cualquier página). Tampoco es una sección porque no es el contenido principal de la página (son tus compras). Fíjate que hemos incluido un encabezado para los lectores de pantalla.

NOTA: los `<div>` son muy fáciles de usar, el problema de esta facilidad es que se tienden a usar sin necesidad añadiendo complejidad al documento por lo que deberías de tratar de emplearlos sólo cuando no haya una solución mejor.

Salto de línea y líneas horizontales

Como hemos comentado, el texto HTML elimina los espacios en blanco y los saltos de línea superfluos. Si queremos indicar forzar un salto de línea debemos emplear la etiqueta `
`, que es el acrónimo de “break”.

Para incluir una línea horizontal podemos emplear la etiqueta `<hr />`, que es el acrónimo de “horizontal rule”.

¿Cómo planificar una web sencilla?

Vamos a proponer una serie de pasos para la definición de un sitio web compuesto por unas pocas páginas. En un sitio grande necesitaras una mayor planificación.

1. Identifica los elementos comunes en la mayoría de las páginas, como son el menú de navegación y el contenido del pie de página.
2. Esboza un esquema sencillo de cuál podría ser la estructura deseada de cada página identificando los bloques principales. Header, Main, Footer, Aside1, Aside2, no Aside...
3. Elabora una lista con el contenido que no sea común a todas las páginas.
4. Ordena o agrupa el contenido no común para hacerte una idea de los contenidos que podrían ir en la misma página. Esto es similar al card sorting.

5. Esquematiza el mapa del sitio web. Dibuja un globo por cada página y la navegación entre ellas con una flecha, comienza por la página principal. Si te hace falta incluye un comentario en el globo de la página.

Tablas HTML

Las tablas nos permiten mostrar datos tabulados organizados en filas y columnas, los datos tabulados son aquellos en lo que normalmente cada fila representa una entidad y cada columna una propiedad de la entidad. En este caso las tablas tienen son útiles para presentar y trabajar con mucha información de manera fácil.

Las tablas HTML NO SE DEBEN EMPLEAR PARA ESTRUCTURAR nuestra web. Las principales razones son:

1. Las tablas reducen la accesibilidad para los usuarios con discapacidad visual. Los lectores de pantalla generan salidas confusas.
2. Las tablas generan estructuras incorrectas que dificultan el mantenimiento del código.
3. Las tablas no tienen respuesta adaptativa automática. Por ejemplo `<header>`, `<section>`, `<article>`, `<div>` adaptan su ancho predeterminado al 100% del elemento padre, en cambio una tabla se dimensiona de forma predeterminada según su contenido.

Las etiquetas básicas para trabajar con tablas son:

- `<table>`, define una tabla.
- `<tr>`, define una fila.
- `<th>`, define una celda de cabecera.
- `<td>`, define una celda de datos, celda de contenido.

La etiqueta `<td>` dispone de los siguientes atributos:

- `colspan`, fusiona el número de celdas indicado en el sentido de la columna.
- `rowspan`, fusiona el número de celdas indicado en el sentido de la fila.

Para facilitar la visualización de la tabla vamos a incluir en nuestros ejemplos el siguiente estilo.

```
<style>
  table{
    border-style: solid;
    border-collapse: collapse;
  }
  td, th{
    border-style: solid;
    text-align: center;
  }
</style>
```

El código de una tabla sencilla sería el siguiente.

```
<table>
```

```

<tr>
  <th>&nbsp;</th>
  <th>Knocky</th>
  <th>Flor</th>
  <th>Ella</th>
  <th>Juan</th>
</tr>
<tr>
  <td>Raza</td>
  <td>Jack Russell</td>
  <td>Caniche</td>
  <td>Perro callejero</td>
  <td>Cocker Spaniel</td>
</tr>
<tr>
  <td>Edad</td>
  <td>16</td>
  <td>9</td>
  <td>10</td>
  <td>5</td>
</tr>
<tr>
  <td>Propietario</td>
  <td>Suegra</td>
  <td>Yo</td>
  <td>Yo</td>
  <td>Cuñada</td>
</tr>
<tr>
  <td>Hábitos alimentarios</td>
  <td>Come las sobras de todos</td>
  <td>Mordisquea la comida</td>
  <td>Come en abundancia</td>
  <td>Come hasta que revienta</td>
</tr>
</table>

```

Genera la siguiente salida.

	Knocky	Flor	Ella	Juan
Raza	Jack Russell	Caniche	Perro callejero	Cocker Spaniel
Edad	16	9	10	5
Propietario	Suegra	Yo	Yo	Cuñada
Hábitos alimentarios	Come las sobras de todos	Mordisquea la comida	Come en abundancia	Come hasta que revienta

Puede ser útil aplicarle un estilo personalizado a una o varias columnas. Se podría pensar inicialmente en aplicar un estilo a cada celda de la columna, pero HTML contiene una etiqueta para agrupar todas las celdas de una columna llamada `<colgroup>`. Esta etiqueta debe ir posicionada siempre después de la etiqueta `<table>`.

Por cada columna de nuestra tabla debemos definir una etiqueta `<col />` de izquierda a derecha, aunque no vayamos a modificar el estilo de la columna. Veamos un ejemplo.

```
<table>
  <colgroup>
    <col />
    <col style="background-color: yellow" />
  </colgroup>
  <tr>
    <th>Dato 1</th>
    <th>Dato 2</th>
  </tr>
  <tr>
    <td>Calcuta</td>
    <td>Pizza</td>
  </tr>
  <tr>
    <td>Robots</td>
    <td>Jazz</td>
  </tr>
</table>
```

Genera la siguiente salida.

Dato 1	Dato 2
Calcuta	Pizza
Robots	Jazz

Conceptos avanzados en tablas

Título de tabla

Podemos añadir un título a la tabla mediante la etiqueta `<caption>`, esta etiqueta debe ir justo debajo de la etiqueta de apertura `<table>`.

```
<table>
  <caption>
    Dinosaurios en el período Jurásico
  </caption>
  ...
</table>
```

El título de una tabla mejora la accesibilidad al dar la posibilidad a los lectores de pantalla de conocer el contenido de la tabla sin tener que entrar en ella.

Estructurar la tabla

Se puede dar estructura a una tabla mediante las etiquetas `<thead>`, `<tfoot>` y `<tbody>`. Aunque visualmente puedan parecer lo mismo, al darle estructura a la tabla podemos hacer que la cabecera y el pie siempre se repitan en cada página impresa.

- `<thead>`, envuelve la fila de cabecera.
- `<tfoot>`, incluye la fila o filas de pie de tabla. Al tener valor semántico, aunque la definamos antes del `<tbody>` se mostrará al final.
- `<thead>`, delimita el contenido de la tabla. Se puede omitir porque el navegador lo añade automáticamente al detectar las otras etiquetas, se recomienda incluirlo para facilitar el mantenimiento.

Anidar tablas

Es posible incluir una tabla anidada dentro de una celda de otra tabla, en algunos casos esto es necesario pero su uso no se recomienda porque dificulta mucho su accesibilidad. Por ejemplo.

```
<table id="tabla1">
  <tr>
    <th>título1</th>
    <th>título2</th>
    <th>título3</th>
  </tr>
  <tr>
    <td id="nested">
      <table id="tabla2">
        <tr>
          <td>celda1</td>
          <td>celda2</td>
          <td>celda3</td>
        </tr>
      </table>
    </td>
    <td>celda2</td>
    <td>celda3</td>
  </tr>
  <tr>
    <td>celda4</td>
    <td>celda5</td>
    <td>celda6</td>
  </tr>
</table>
```

Genera la siguiente salida.

título1			título2	título3
celda1	celda2	celda3	celda2	celda3
celda4			celda5	celda6

Tablas para usuarios con discapacidad visual

Ya hemos comentado que el uso de celdas de cabecera en las columnas y en las filas facilita el procesamiento de la información a los lectores de pantalla.

La etiqueta `<th>` incluye un atributo **“scope”** que permite indicar en qué sentido actúa el encabezado, para la columna valor **“col”** o para la fila valor **“row”**. Por ejemplo.

```
<tr>
  <th scope="row">Corte de pelo</th>
  <td>Peluquería</td>
  <td>12/09</td>
  <td>Gran idea</td>
  <td>30</td>
</tr>
```

NOTA: Existen otros atributos para mejorar la accesibilidad (id, header) pero se desaconsejan debido a la sobrecarga de código que significan.

Formularios Web

Los formularios web son el principal elemento de interacción entre el usuario y el sitio web. Un formulario permite enviar datos desde la página al servidor o un caso menos común solicitar datos para modificar la interfaz (por ejemplo, añadir un nuevo elemento a una lista).

Un formulario se compone de controles de diversos tipos, etiquetas y botones, el principal elemento para la introducción de datos es el `<input>`.

El elemento `<form>` define un formulario. Todos sus atributos son opcionales, aunque lo normal es contar siempre con al menos estos dos:

- **“action”**, define la URL a la que se mandarán los datos.
- **“method”**, define el método HTTP con que se envían los datos (generalmente **get** o **post**).

Un ejemplo de formulario.

```
<form action="/pagina_manejadora" method="post"></form>
```

El ejemplo anterior no es funcional, no se solicita ninguna información. Veamos como añadirle algunos campos, supongamos un campo para solicitar un nombre, un campo para solicitar un email y campo de texto multilínea para dejar un mensaje.

Fíjate en el uso de las etiquetas `<label>`, `<input>` y `<textarea>` del siguiente ejemplo.

```
<form action="/pagina_manejadora" method="post">
  <ul>
    <li>
      <label for="name">Nombre:</label>
      <input type="text" id="name" name="user_name" />
    </li>
    <li>
      <label for="mail">Correo electrónico:</label>
      <input type="email" id="mail" name="user_mail" />
    </li>
    <li>
      <label for="msg">Mensaje:</label>
      <textarea id="msg" name="user_message"></textarea>
    </li>
  </ul>
</form>
```

NOTA: la lista `` la empleamos para dar un formato básico sin introducir CSS.

Genera la siguiente salida.

- Nombre:
- Correo electrónico:
- Mensaje:

La etiqueta `<label>` añade una etiqueta explícita a cada control del formulario y se asocia al control a través del atributo `"for"` que incluye el valor del `"id"` del control de entrada.

El uso de `<label>` mejora la accesibilidad y usabilidad del formulario ya que al pulsar sobre el `<label>` automáticamente se selecciona el control asociado.

Empleamos dos tipos de `<input>`. En el caso de `type="text"` acepta cualquier entrada de texto. En el caso de `type="email"` sólo acepta cadenas con formato de email.

En el caso de la entrada de tipo `<textarea></textarea>` que nos permite introducir un texto multilínea fíjate que no es un elemento vacío como `<input>`. Esto implica que para acceder a su valor o para dar un valor por defecto la codificación es diferente.

En el caso del `<input>` debemos emplear el atributo `"value"`.

```
<input type="text" value="por defecto este elemento se llena con este texto" />
```

En el caso del `<textarea></textarea>` debemos indicar su contenido.

```
<textarea>
```

Por defecto, este elemento contiene este texto

</textarea>

Etiqueta <button>

Nos queda añadir un botón para poder enviar el formulario. Completamos nuestro ejemplo con el siguiente código.

```
<li class="button">
  <button type="submit">Envíe su mensaje</button>
</li>
```

A estas alturas tu formulario debería tener el siguiente aspecto:

- Nombre:
- Correo electrónico:
- Mensaje:
-

Fíjate en el atributo “type” del botón ya que define el comportamiento del control. Admite los siguientes valores:

- “submit”, envía los datos a la dirección definida en el “action” del formulario. Es el valor por defecto.
- “reset”, restablece los valores del formulario.
- “button”, que no hace nada, su utilidad radica en poder ejecutar una función javascript.

Los formularios admiten dos tipos de validaciones:

- Validación interna, dentro del propio formulario.
- Validación externa, mediante javascript al pulsar el botón de envío.

En este ejemplo hemos visto un campo que requería un correo electrónico, pero a la hora introducir valores erróneos en formulario no se queja, no vemos ninguna diferencia visual. Aunque no lo veamos internamente el estado del formulario es de datos inválidos, para poder probar esto vamos a añadir un poco de estilo.

```
<style>
  input:invalid {
    border: 2px dashed red;
  }

  input:valid {
    border: 2px solid black;
  }
</style>
```


Fíjate como se ven los errores.

- Nombre:
- Correo electrónico:
- Mensaje:
-

Estructura de un formulario.

Hemos visto que podemos darle estructura al formulario mediante listas no ordenadas ``, aunque también es común emplear las etiquetas `<p>` y `<div>`. Según sea el caso deberemos emplear lo que más nos convenga.

En el caso de tener un formulario muy grande también es posible emplear secciones `<section>`.

En algunos casos puede ser necesario crear agrupaciones, por ejemplo, cuando tenemos varias selecciones excluyentes en el mismo formulario. Para estos casos contamos con las `<fieldset>` y `<legend>`.

Etiquetas `<fieldset>` y `<legend>`

Permiten agrupar controles `<fieldset>` e incluir un título `<legend>` para el grupo. La etiqueta `<legend>` debe ir inmediatamente después de la etiqueta `<fieldset>`. Por ejemplo:

```
<form>
  <fieldset>
    <legend>Tamaño del zumo de fruta</legend>
    <p>
      <input type="radio" name="size" id="size_1" value="small" />
      <label for="size_1">Pequeño</label>
    </p>
    <p>
      <input type="radio" name="size" id="size_2" value="medium" />
      <label for="size_2">Mediano</label>
    </p>
    <p>
      <input type="radio" name="size" id="size_3" value="large" />
      <label for="size_3">Grande</label>
    </p>
  </fieldset>
</form>
```

Genera la siguiente imagen.

Tamaño del zumo de fruta

☐ Pequeño

☐ Mediano

☐ Grande

Etiquetas de entrada

Hemos visto el funcionamiento básico del formulario a continuación listamos los distintos tipos de entradas que admite.

Etiqueta `<select>`

No es propiamente un `<input>` pero es un elemento de entrada frecuente. Esta etiqueta nos permite definir una lista desplegable en la que elegir un valor. La etiqueta `<select>` va acompañada de una colección de etiquetas `<option>` para especificar cada uno de los posibles valores.

```
<label for="coches">Elige un coche:</label>
<select name="coches" id="coches">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="mercedes" selected>Mercedes</option>
  <option value="audi">Audi</option>
</select>
```

Genera la siguiente imagen.

Elige un coche: Mercedes ▼

- Volvo
- Saab
- Mercedes**
- Audi

Adicionalmente podemos crear grupos dentro de la lista de opciones con `<optgroup>`. La etiqueta del nombre del grupo no es seleccionable.

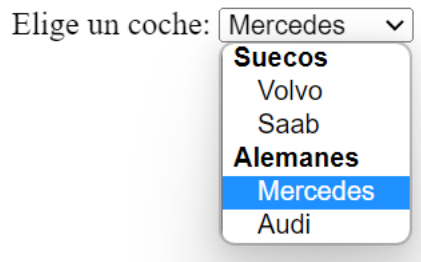
```
<label for="coches">Elige un coche:</label>
<select name="coches" id="coches">
  <optgroup label="Suecos">
    <option value="volvo">Volvo</option>
    <option value="saab">Saab</option>
  </optgroup>
```

```

<optgroup label="Alemanes">
  <option value="mercedes" selected>Mercedes</option>
  <option value="audi">Audi</option>
</optgroup>
</select>

```

Genera la siguiente imagen.



La etiqueta `<select>` permite seleccionar múltiples valores para ellos debemos emplear un par de propiedades:

- **multiple**, propiedad binaria, activa la selección múltiple.
- **size**, indica el número de elementos máximos que podemos seleccionar.

Por ejemplo.

```

<select name="coches" id="coches" multiple size="2">

```

Entrada `<input type="text">`

Permite introducir un texto plano de una línea.

Es el valor por defecto.

```

<form>
  <label for="fname">Nombre:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Apellido:</label><br>
  <input type="text" id="lname" name="lname">
</form>

```

Genera la siguiente imagen.

Nombre:

Apellido:

Entrada `<input type="password">`

Enmascara la entrada con asteriscos. Evidentemente, al enviar se envía el valor real.

```
<form>
  <label for="username">Usuario:</label><br>
  <input type="text" id="username" name="username"><br>
  <label for="pwd">Password:</label><br>
  <input type="password" id="pwd" name="pwd">
</form>
```

Genera la siguiente imagen.

Visualización de un formulario de login. Se muestra la etiqueta 'Usuario:' seguida de un campo de texto con el valor 'Alex'. Debajo, se muestra la etiqueta 'Password:' seguida de un campo de contraseña con seis puntos para enmascarar el texto.

Entrada `<input type="submit">`

Envía los datos del formulario a la URL indicada en el atributo "action".

```
<form action="/action_page.php">
  <label for="fname">Nombre:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Apellido:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Enviar">
</form>
```

Genera la siguiente imagen.

Visualización de un formulario de registro. Se muestra la etiqueta 'Nombre:' seguida de un campo de texto con el valor 'John'. Debajo, se muestra la etiqueta 'Apellido:' seguida de un campo de texto con el valor 'Doe'. En la parte inferior, hay un botón rectangular con el texto 'Enviar'.

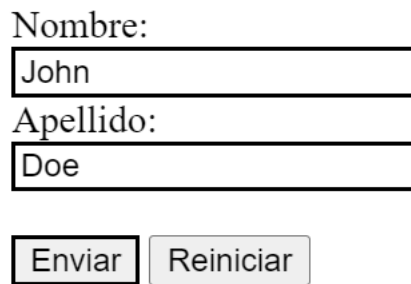
Entrada `<input type="reset">`

Reinicia los datos del formulario a su valor inicial.

```
<form action="/action_page.php">
  <label for="fname">Nombre:</label><br>
  <input type="text" id="fname" name="fname" value="John"><br>
  <label for="lname">Apellido:</label><br>
  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Enviar">
  <input type="reset" value="Reiniciar">
</form>
```

</form>

Genera la siguiente imagen.



Nombre:

Apellido:

Entrada <input type="radio">

Permite seleccionar UNA ÚNICA opción entre varias.

```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" checked id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

Genera la siguiente imagen.

Elige tu lenguaje Web favorito:



☐ HTML
☒ CSS
☐ JavaScript

Entrada <input type="checkbox">

Permite seleccionar 0 o más opciones.

```
<form>
  <input type="checkbox" id="vehicle1" name="vehicle1" value="Bike">
  <label for="vehicle1"> Tengo una moto</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> Tengo un coche</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3" value="Boat">
  <label for="vehicle3"> Tengo un barco</label>
</form>
```

Genera la siguiente imagen.

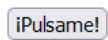
- ☐ Tengo una moto
- ☐ Tengo un coche
- ☐ Tengo un barco

Entrada `<input type="button">`

No realiza ninguna operación por defecto, su funcionalidad debe implementarse en el evento "onclick" asociado.

```
<input type="button" onclick="alert('¡Hola caracola!')"
value="¡Pulsame!">
```

Genera la siguiente imagen.

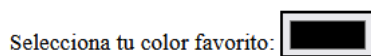


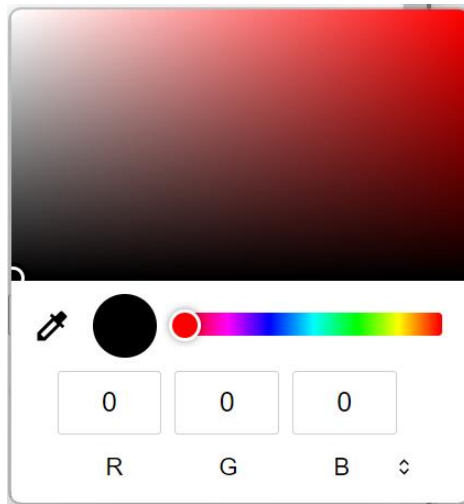
Entrada `<input type="color">`

Muestra un selector de color.

```
<form>
  <label for="favcolor">Selecciona tu color favorito:</label>
  <input type="color" id="favcolor" name="favcolor">
</form>
```

Genera la siguiente imagen.





Entrada `<input type="date">`

Permite introducir una fecha con el formato “dd/mm/yyyy”.

```
<form>
  <label for="birthday">Cumpleaños:</label>
  <input type="date" id="birthday" name="birthday">
</form>

<p>Admite valores máximo y mínimo</p>
<form>
  <label for="datemax">Introduce una fecha anterior 1980-01-01:</label>
  <input type="date" id="datemax" name="datemax" max="1979-12-31"><br><br>
  <label for="datemin">Introduce una fecha posterior 2000-01-01:</label>
  <input type="date" id="datemin" name="datemin" min="2000-01-02">
</form>
```

Genera la siguiente imagen.

Cumpleaños:

Admite valores máximo y mínimo

Introduce una fecha anterior 1980-01-01:

Introduce una fecha posterior 2000-01-01:

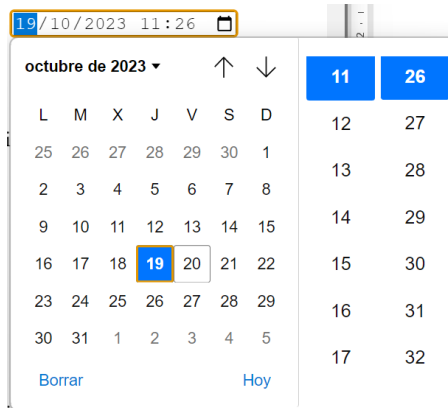
Entrada `<input type="datetime-local">`

Permite introducir una fecha y una hora sin zona horaria.

```
<form>
  <label for="birthdaytime">Cumpleaños (fecha y hora):</label>
  <input type="datetime-local" id="birthdaytime" name="birthdaytime">
</form>
```

Genera la siguiente imagen.

Cumpleaños (fecha y hora):



Entrada `<input type="email">`

Permite introducir un email.

```
<form>
  <label for="email">Introduce tu correo electrónico:</label>
  <input type="email" id="email" name="email">
</form>
```

Genera la siguiente salida.

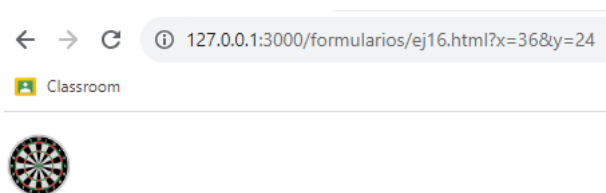
Introduce tu correo electrónico:

Entrada `<input type="image">`

Permite utilizar una imagen como botón "submit", además devuelve las coordenadas en la imagen donde se ha pulsado el ratón.

```
<form>
  <input type="image" src="diana.jpg" alt="Submit" width="48"
height="48">
</form>
```

Genera la siguiente imagen.



Entrada `<input type="file">`

Permite seleccionar un fichero para ser cargado en el servidor. POSIBLEMENTE debas ajustar parámetros adicionales en el formulario para soportar la carga (enctype).

```
<form>
  <label for="myfile">Selecciona un archivo a subir:</label>
```



```
<input type="file" id="myfile" name="myfile">
</form>
```

Genera la siguiente imagen.

Selecciona un archivo a subir:

Esta entrada admite el atributo “accept” que permite indicar el tipo de fichero admitido. Admite los siguientes valores:

- Una lista de extensiones separados por coma, por ejemplo “.jpg, .png, .doc”
- “audio/*”, que representa archivos de audio.
- “video/*”, que representa archivos de video.
- “image/*”, que representa archivos de imagen.

Entrada `<input type="hidden">`

Permite definir un campo no visible, ni modificable, por el usuario. Permite ocultar información que será enviada con el resto de datos.

OJO: este valor se puede modificar con las herramientas de desarrollo, cuidado con la veracidad de la información.

```
<form>
  <label for="fname">Nombre:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <input type="hidden" id="custId" name="custId" value="3487">
  <input type="submit" value="Submit">
</form>
```

Genera la siguiente imagen.

Nombre:

Entrada `<input type="month">`

Permite seleccionar un mes y un año.

```
<form>
  <label for="bdaymonth">Cumpleaños (mes y año):</label>
  <input type="month" id="bdaymonth" name="bdaymonth">
</form>
```

Genera la siguiente imagen.

Cumpleaños (mes y año):



Entrada `<input type="number">`

Permite seleccionar un número. Este campo admite restricciones.

```
<form>
  <label for="quantity">Cantidad (entre 1 y 5):</label>
  <input type="number" id="quantity" name="quantity" min="1" max="5">
</form>
```

Genera la siguiente imagen.

Cantidad (entre 1 y 5):

Entrada `<input type="range">`

Permite introducir un valor numérico mediante un control deslizante. Los valores por defecto van del 0 al 100.

```
<form>
  <label for="vol">Volumen (entre 0 y 50):</label>
  <input type="range" id="vol" name="vol" min="0" max="50">
</form>
```

Genera la siguiente imagen.

Volumen (entre 0 y 50):

Entrada `<input type="search">`

Se emplea para introducir términos de búsqueda. Su comportamiento es similar a un campo de texto común.

```
<form>
  <label for="gsearch">Search Google:</label>
  <input type="search" id="gsearch" name="gsearch">
</form>
```

Genera la siguiente imagen.

Search Google:

Entrada `<input type="tel">`

Permite introducir un número de teléfono. El formato puede validarse mediante un patrón.

```

<form>
  <label for="phone">Introduce tu teléfono:</label>
  <input type="tel" id="phone" name="phone" pattern="[0-9]{3}-[0-9]{3}-[0-9]{3}">
</form>

```

Genera la siguiente imagen.

Introduce tu teléfono:

Entrada <input type="time">

Permite introducir una hora sin franja horaria.

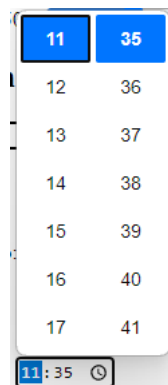
```

<form>
  <label for="appt">Selecciona una hora:</label>
  <input type="time" id="appt" name="appt">
</form>

```

Genera la siguiente imagen.

Selecciona una hora:



Entrada <input type="url">

Permite introducir una dirección URL.

```

<form>
  <label for="homepage">Añade tu página web:</label>
  <input type="url" id="homepage" name="homepage">
</form>

```

Genera la siguiente imagen.

Añade tu página web:

Entrada <input type="week">

Permite introducir una semana y un año.

```

<form>
  <label for="week">Selecciona una semana:</label>

```

```
<input type="week" id="week" name="week">
</form>
```

Genera la siguiente imagen.



Atributos de la etiqueta `<input>`

Podemos ajustar el comportamiento de la etiqueta `<input>` mediante atributos. Los atributos pueden afectar al comportamiento del control, a su aspecto o añadir validaciones. A continuación, se incluye la lista de atributos disponibles ten en cuenta que su utilidad puede ser mayor o menor en función del “`type`” empleado.

Atributo “`value`”

Permite darle un valor inicial al campo de entrada.

```
<label for="fname">Nombre:</label><br>
<input type="text" id="fname" name="fname" value="John">
```

Atributo “`readonly`”

Permite especificar que un campo de entrada es de sólo lectura. El valor se envía con el submit.

```
<label for="fname">Nombre:</label><br>
<input type="text" id="fname" name="fname" value="John" readonly>
```

Atributo “`disabled`”

Permite deshabilitar un campo. El valor NO SE ENVÍA con el submit.

```
<label for="fname">Nombre:</label><br>
<input type="text" id="fname" name="fname" value="John" disabled>
```

Atributo “`size`”

Establece el ancho visible indicando el número de caracteres que deben verse.

```
<label for="fname">Nombre:</label><br>
```

```
<input type="text" id="fname" name="fname" value="John" size="50">
```

Atributo "maxlength"

Establece el número máximo de caracteres admitido en el campo de entrada. Alcanzado el valor no se podrá continuar escribiendo.

```
<label for="pin">PIN:</label><br>
<input type="text" id="pin" name="pin" maxlength="4" size="4">
```

Atributos "min" y "max"

Establecen el valor mínimo y máximo para un campo de entrada. Funciona con los siguientes tipos de entrada: "number", "range", "date", "datetime-local", "month", "time" y "week".

```
<label for="datemin">Introduce una fecha posterior 2000-01-01:</label>
<input type="date" id="datemin" name="datemin" min="2000-01-02"><br><br>
```

```
<label for="quantity">Cantidad (entre 1 y 5):</label>
<input type="number" id="quantity" name="quantity" min="1" max="5">
```

Atributo "multiple"

Permite al usuario introducir más de un valor en el campo. Funciona con los siguientes tipos de entrada: "email" y "file".

```
<label for="files">Selecciona los archivos:</label>
<input type="file" id="files" name="files" multiple>
```

Atributo "pattern"

Permite especificar una expresión regular que debe ser cumplida al enviar el formulario. Funciona con los siguientes tipos de entrada: "text", "date", "search", "url", "tel", "email", y "password".

NOTA: puedes emplear el atributo "title" para describir el patrón y ayudar al usuario.

```
<label for="country_code">Código de país:</label>
<input type="text" id="country_code" name="country_code"
  pattern="[A-Za-z]{3}" title="Código de país 3 letras">
```

Atributo "placeholder"

Permite mostrar una pista que ayude al usuario a rellenar el campo. Esta pista se muestra antes de que el usuario introduzca un valor. Funciona con los siguientes tipos de entrada: "text", "search", "url", "tel", "email", y "password".

```
<label for="phone">Introduce un número de teléfono:</label>
<input type="tel" id="phone" name="phone"
  placeholder="123-45-678"
  pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}">
```

Atributo "required"

Especifica que el campo de entrada debe ser rellenado antes de enviar el formulario. Funciona con los siguientes tipos de entrada: "text", "search", "url", "tel", "email", "password", "date pickers", "number", "checkbox", "radio", y "file".

```
<label for="username">Nombre:</label>
<input type="text" id="username" name="username" required>
```

Atributo "step"

Especifica los intervalos numéricos permitidos. Se puede combinar con "min" y "max" para definir rangos de valores legales. Funciona con los siguientes controles: "number", "range", "date", "datetime-local", "month", "time" y "week".

```
<label for="points">Puntos:</label>
<input type="number" id="points" name="points" step="3">
```

Atributo "autofocus"

Permite definir el campo de entrada que estará activo cuando la página se cargue.

```
<form>
  <label for="fname">Nombre:</label><br>
  <input type="text" id="fname" name="fname" autofocus><br>
  <label for="lname">Apellido:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

Atributos "height" y "width"

Permite indicar el alto y ancho de una imagen al cargar la página. No es necesario, pero si recomendable ya que evita que la pantalla se redimensione durante la carga al conocer los valores iniciales.

```
<input type="image" src="diana.jpg" alt="Submit" width="48" height="48">
```

Atributo "list"

Permite referenciar un elemento `<datalist>` con valores predefinidos para el elemento de entrada. Estos valores se muestran como sugerencias mientras el usuario edita el campo.

```
<form>
  <input list="browsers">
  <datalist id="browsers">
    <option value="Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

Atributo "autocomplete"

Permite al navegador predecir los valores de entrada en el formulario en base al historial de datos previos tecleados. Funciona con los siguientes controles: "text", "search", "url", "tel", "email", "password", "datepickers", "range", y "color".

```
<form action="/action_page.php" autocomplete="on">  
...  
</form>
```

Atributos de la etiqueta <form>

Aunque hay más atributos a continuación vamos a resumir los que debes conocer y sus valores más frecuentes.

Atributo "action"

Permite indicar la URL a la que se envían los datos del formulario.

Atributo "method"

Permite indicar el método HTTP empleado para enviar los datos del formulario.

Admite dos posibles valores:

- **"GET"**, valor por defecto, la información se añade a la URL como parámetros. Esto implica que los datos son visibles. Es apropiado para solicitudes de búsqueda o para recuperar información. El tamaño de una URL está limitado a 2048 caracteres y únicamente permite transmitir texto.
- **"POST"**, incluye la información en la cabecera de la petición por lo que no son visibles en la URL. No tiene un límite de tamaño. Son apropiados para enviar datos sensibles y se emplean en operaciones de tipo inserción, actualización o eliminación. Permite enviar archivos.

Atributo "enctype"

Permite indicar la manera en la que se van a codificar los datos a enviar en las peticiones de tipo POST.

Admite tres posibles valores:

- **"application/x-www-form-urlencoded"**, valor por defecto, codifica los caracteres especiales antes de ser enviados.
- **"multipart/form-data"**, valor NECESARIO PARA ENVIAR ARCHIVOS.
- **"text/plain"**, envía los datos en texto plano.

El siguiente ejemplo recoge los atributos previos

```
<h2>Subir un Archivo</h2>
```

```
<form action="procesar_archivo.php" method="post"
enctype="multipart/form-data">
  <label for="archivo">Selecciona un archivo:</label>
  <input type="file" name="archivo" id="archivo" accept=".pdf, .doc,
.txt" required>
  <br>
  <input type="submit" value="Subir Archivo">
</form>
```

Atributo "target"

Permite indicar donde se va a mostrar la respuesta.

Admite cuatro posibles valores:

- **"_blank"**, muestra la respuesta en una nueva ventana o pestaña.
- **"_self"**, valor por defecto, muestra la respuesta en el "frame" actual, que normalmente al no haberse definido es la ventana actual del navegador.
- **"_parent"**, muestra la respuesta en el "frame" padre del frame actual.
- **"_top"**, muestra la respuesta en la ventana actual del navegador.

Es decir, salvo codificación horrorosa **"_parent"** y **"_top"** tienen el mismo resultado que **"_self"**, estos dos valores son de la época en la que se codificaba con frames.

```
<form action="procesar_form.php" method="post" target="_blank"></form>
```

Otros controles, medidor y barra de progreso.

Etiqueta Medidor <meter>

Un medidor **<meter>** nos permite representar una barra mediante un rango, valor mínimo y máximo. Su principal característica es que en base al valor alcanzado **modifica su aspecto visual pasando de verde a amarillo, y de amarillo a rojo al alcanzar valores altos.**

Esta etiqueta de múltiples propiedades para cumplir con su objetivo.

- **"min"**, valor mínimo.
- **"max"**, valor máximo.
- **"low"**, permite establecer el rango de la parte baja entre **"min"** y **"low"**.
- **"high"**, permite establecer el rango de la parte alta entre **"high"** y **"max"**.
- **"optimum"**, valor optimo o valor deseado. Salvo codificación adicional sólo aporta valor semántico.
- **"value"**, valor actual que representa medidor.

Es decir, los valores tendrían que quedar así **"min" ≤ "low" ≤ "high" ≤ "max"**.

```
<meter min="0" max="100" value="60" low="33" high="66"
optimum="0">60</meter>
```

Genera la siguiente salida.



Fíjate que hemos rellenado el contenido de la etiqueta `<meter>` con el mismo valor del atributo `“value”` para mejorar la accesibilidad, los lectores de pantalla van a leer el contenido.

Etiqueta Barra de progreso `<progress>`

La barra de progreso `<progress>` representa un valor que cambia a lo largo del tiempo. Sus atributos son `“max”` para indicar el valor máximo y `“value”` para indicar el actual.

```
<progress max="100" value="75">75/100</progress>
```

Genera la siguiente salida.



Rellenamos el contenido para que pueda ser leído por los lectores de pantalla, fíjate que le hemos dado formato 75 de 100.

Cajas de dialogo

Las cajas de dialogo nos permiten crear ventanas emergentes. Normalmente se emplean para crear popups o para crear formularios modales.

La etiqueta `<dialog>` no permite definir una caja de dialogo que contendrá en su interior todo el maquetado HTML que necesitemos.

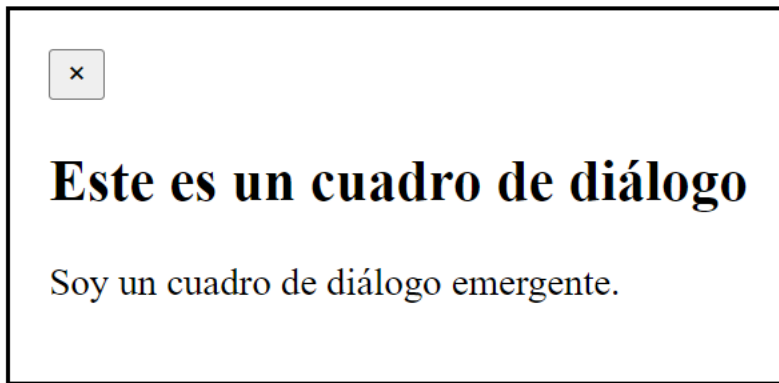
La etiqueta `<dialog>` tiene una propiedad `“open”` de tipo binaria que indica si caja de dialogo debe mostrarse o no.

Un **popup** normalmente dispone de una cabecera con un botón (x) que cierra la caja de dialogo seguido del contenido del popup.

Por ejemplo

```
<dialog id="miPopup" open>
  <header><button id="cerrarPopup" aria-label="Cerrar">
    &times;</button></header>
    <h2>Este es un cuadro de diálogo</h2>
    <p>Soy un cuadro de diálogo emergente.</p>
</dialog>
```

Genera la siguiente salida.



Un formulario modal normalmente incluye el contenido del formulario seguido de un pie con los botones necesarios. Siempre un botón para cerrar el dialogo y si se puede enviar datos también el botón “submit”.

Por ejemplo

```
<button id="abrirModal" type="button">Pulsa para abrir</button>
```

```
<dialog id="miModal">
  <h2>Formulario modal</h2>
  <p>Soy un formulario emergente</p>
  <footer>
    <button id="cancelar">Cancelar</button>
    <button id="aceptar" type="submit">Aceptar</button>
  </footer>
</dialog>
```

Genera la siguiente salida.



El código que hace funcionar los ejemplos anteriores es el siguiente.

```

<script>
  const $ = selector => document.querySelector(selector);
  const $modalBotonX = $('#cerrarPopup');
  const $modalDialogo = $('#miPopup');

  $modalBotonX.addEventListener('click', () => {
    $modalDialogo.close();
  });

  const $botonAbrirModal = $('#abrirModal');
  const $miModal = $('#miModal');
  const $botonCancelar = $('#cancelar');

  $botonAbrirModal.addEventListener('click', () => {
    $miModal.showModal();
  });

  $botonCancelar.addEventListener('click', () => {
    $miModal.close();
  } )

</script>

```

Referencias

Tutorial HTML de w3schools

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

Tutorial de CSS de w3schools

<https://www.w3schools.com/css/default.asp>

Tutorial de JavaScript de w3schools

<https://www.w3schools.com/js/default.asp>

¿Cómo hacer...? De w3schools

<https://www.w3schools.com/howto/default.asp>

Atributos comunes

https://www.w3schools.com/html/html_attributes.asp

MDN. Aprender HTML

<https://developer.mozilla.org/en-US/docs/Learn/HTML>

MDN. Aprende CSS

<https://developer.mozilla.org/en-US/docs/Learn/CSS>

MDN. Aprende JavaScript

<https://developer.mozilla.org/en-US/docs/Learn/JavaScript>

MDN. Como convertirse en un desarrollador web

https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer

Formularios WEB

<https://developer.mozilla.org/es/docs/Learn/Forms>

Listado de elementos HTML, incluye obsoletos

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

Herramientas

Extensión para visual studio code para ejecutar webs locales sin instalar un servidor web

<https://marketplace.visualstudio.com/items?itemName=ms-vscode.live-server>