

HTML

Ángel Villalba Fdez-Paniagua

Table of Contents

1. Historia del HTML	1
2. Introducción	1
3. Estructura	2
4. Etiquetas principales de HTML	3
4.1. Etiqueta DOCTYPE	4
4.2. Etiqueta <html>	4
4.3. Etiqueta <head>	4
4.4. Etiqueta <body>	4
5. Etiquetas que se usan en <head>	5
5.1. Etiqueta <title>	5
5.2. Etiqueta <style>	5
5.3. Etiqueta <script>	6
6. Etiqueta <meta> (PAGINA 191)	8
7. Elementos estructurales	8
7.1. Etiqueta <header>	11
7.2. Etiqueta <footer>	11
7.3. Etiqueta <nav>	12
7.4. Etiqueta <article>	12
7.5. Etiqueta <aside>	13
7.6. Etiqueta <section>	13
7.7. Ejercicio	14
8. Elementos de texto	14
8.1. Elementos de título: <h1> - <h6>	14
8.2. Etiqueta <p>	14
8.3. Etiqueta 	15
8.4. Etiqueta <i>	15
8.5. Etiqueta <sup>	15
8.6. Etiqueta <sub>	16
8.7. Etiqueta 	16
8.8. Etiqueta <hr>	16
8.9. Espacios en blanco	16
9. Etiqueta de enlace: <a>	17
9.1. Enlaces a páginas externas	17
9.2. Enlaces a distintas páginas dentro del mismo sitio	18
9.3. Enlaces a distintas secciones de la misma página	18
9.4. Enlaces que abren la página en otra ventana	19
9.5. Enlaces a correos	19
9.6. Ejercicio	20

9.7. Ejercicio	20
10. Elementos de listas	20
10.1. Etiqueta 	20
10.2. Etiqueta 	20
10.3. Etiqueta 	20
10.4. Etiqueta <dl>	21
10.5. Etiqueta <dt>	21
10.6. Etiqueta <dd>	21
10.7. Listas anidadas	22
10.8. Ejercicio	22
11. Elementos de Tablas	22
11.1. Etiqueta <table>	22
11.2. Etiqueta <tr>	23
11.3. Etiqueta <td>	23
11.4. Etiqueta <th>	23
11.5. Etiqueta <thead>	24
11.6. Etiqueta <tbody>	24
11.7. Etiqueta <tfoot>	24
11.8. Combinando filas y columnas	24
11.9. Ejercicio	25
12. Elementos de Formularios	25
12.1. Etiqueta <form>	26
12.2. Etiqueta <input>	27
12.3. Etiqueta <label>	31
12.4. Etiqueta <fieldset>	31
12.5. Etiqueta <legend>	32
12.6. Etiqueta <button>	32
12.7. Etiqueta <select>	32
12.8. Etiqueta <option>	33
12.9. Etiqueta <textarea>	33
12.10. Atributo placeholder	33
12.11. Atributo autofocus	34
12.12. Atributo pattern	34
12.13. Atributo required	34
12.14. Ejercicio	34
12.15. Ejercicio	34
12.16. Ejercicio	35
13. Imagenes	35
13.1. Etiqueta 	35
13.2. Etiqueta <figure>	36
13.3. Etiqueta <figcaption>	36

13.4. Transparencia en las imagenes	37
13.5. Ejercicio	37
14. Elementos semánticos	37
14.1. Etiqueta 	37
14.2. Etiqueta 	38
14.3. Etiqueta <blockquote>	38
14.4. Etiqueta <q>	38
14.5. Etiqueta <abbr>	38
14.6. Etiqueta <cite>	39
14.7. Etiqueta <dfn>	39
14.8. Etiqueta <address>	39
14.9. Etiqueta <ins>	40
14.10. Etiqueta 	40
14.11. Etiqueta <s>	40
14.12. Etiqueta <meter>	40
14.13. Etiqueta <progress>	41
14.14. Ejercicio	41
15. Atributos globales	41
15.1. id	41
15.2. class	42
15.3. title	42
15.4. hidden	42
15.5. data	43
15.6. dir	43
15.7. style	43
15.8. spellcheck	44
16. Etiquetas <meta>	44
16.1. Ejercicio	45
17. Comentarios	45
18. Escapar caracteres	46
18.1. Ejercicio	46

1. Historia del HTML

HTML surge por el año 1980 cuando un físico **Tim Berners-Lee** que trabajaba en el CERN, propuso un nuevo sistema para compartir los documentos.

Tim se asocia con un ingeniero de sistemas **Robert Cailliau** para desarrollar su sistema de hipertexto, y lo presentaron a una convocatoria que había en aquel momento y que resultó ser la ganadora. La llamaron la **WorldWideWeb** (W3).

El primer documento formal con la descripción de HTML se publicó en 1991 con el nombre de *HTML Tags* y consistía en unas cuantas etiquetas para definir listas, títulos, párrafos, secciones, enlaces, y alguna etiqueta más.

En 1993 se realiza una propuesta formal para convertirlo en un estándar, y aunque habían logrado avances significativos (habían añadido tablas, imágenes y formularios), ninguna de las propuestas presentadas (**HTML** y **HTML+**) consiguieron convertirse en un estándar oficial.

En 1995 el organismo IETF, organiza un grupo de trabajo de HTML y consigue publicar el estándar **HTML 2.0**, que era el primer estándar oficial de HTML.

A partir de 1996 los estándares de HTML los publica la **W3C**. La versión **HTML 3.2** se publicó en 1997 e incorporaba los últimos avances de las páginas web desarrolladas hasta 1996 como texto que se ponía alrededor de las imágenes.

En 1998 se publicó la versión **HTML 4.0** y suponía un gran salto respecto a las versiones anteriores. Entre las novedades más destacadas que traía consigo, se encontraban las **hojas de estilos CSS**, la posibilidad de incluir pequeños **scripts**, tablas y formularios más complejos...

A partir de la publicación de **HTML 4.01** la actividad de estandarización de HTML se detuvo y la W3C se centró en el desarrollo del estándar de XHTML. Por este motivo en el año 2004, algunas empresas como Apple, Mozilla y Opera muestran su preocupación ante la falta de interés del W3C en HTML, y deciden crear una nueva organización para ello, la WHATWG.

Esta nueva organización se centra en el futuro del estándar de HTML5, cuyo primer borrador oficial se publicó en 2008, y debido a la fuerza de las empresas que forman esta organización y a la publicación del borrador de HTML5, la W3C decidió retomar la actividad de estandarización de HTML.

HTML5 es la última versión que ha salido de HTML y trae consigo muchas mejoras con respecto a las anteriores versiones, entre ellas se encuentran todas las APIs nuevas que nos permiten acceder al hardware del dispositivo, usar la memoria del navegador, acceder a la ubicación del dispositivo, pintar usando el canvas...

2. Introducción

HTML (*HyperText Markup Language*) es el lenguaje de marcado que nos permite definir como va a ser la estructura de una página web para que los navegadores entiendan como tienen que mostrar el contenido a los usuarios.

Cuando se habla de **hipertexto**, se refiere a que nuestro documento HTML nos va a permitir crear enlaces para visitar otras páginas, o movernos entre las distintas secciones de la nuestra.

Y cuando decimos que es de **marcado** nos estamos refiriendo a que vamos a envolver el contenido de la página entre etiquetas que le van a dar significado a ese contenido, y permitirá al navegador mostrarlo correctamente.

Para crear un documento HTML, tenemos que crear un archivo con la extensión **.html**. Y si lo abrimos con el navegador podremos ver el contenido de este.

3. Estructura

En cualquier clase de documentos, es importante estructurar bien el contenido para ayudar a los lectores a entender el mensaje que se quiere transmitir. Por tanto, es importante saber como estructurar bien los documentos.

Por ejemplo, los formularios en una web se pueden estructurar igual que cualquier formulario en papel. El formulario tendrá una serie de preguntas con unos campos para rellenar y estas preguntas pueden estar separadas en secciones por títulos.

¿Algún otro ejemplo?

Usar los títulos y subtítulos en un documento es una forma de reflejar la estructura de la información.

Por ejemplo, en la noticia de un periodico podemos encontrar un título principal, un resumen de la noticia, alguna imagen y la información dividida en párrafos. Estos párrafos a veces pueden estar separados en secciones con un subtítulo para describir el contenido de la sección de la noticia.

Para estructurar una página web, tenemos que meter el contenido que queremos mostrar entre unas **etiquetas** HTML. Los **elementos** de HTML están compuestos por dos etiquetas HTML, una que abre y otra que cierra.

Estas etiquetas son palabras que van entre los signos de *menor* y *mayor* (<p>). Y son las encargadas de decirle al navegador que es lo que va entre esas etiquetas.

Esas etiquetas actuan como contenedores, e indican que tipo de información es lo que hay entre esas etiquetas.

Entonces, las etiquetas de apertura se componen de:

- Un signo de menor <
- Una palabra que nos dice el proposito que tiene la información que se encuentra entre las etiquetas.
- Un signo de mayor >

Mientras que las etiquetas de cierre se componen de:

- Un signo de menor <

- El signo de división /
- Una palabra que nos dice el propósito que tiene la información que se encuentra entre las etiquetas.
- Un signo de mayor >

```
<!DOCTYPE html>
<html lang="es">
<body>
  <h1>Título principal</h1>
  <p>Esto es un párrafo</p>
  <p>Esto es otro párrafo</p>
</body>
</html>
```

Las etiquetas pueden contener **atributos**. Los atributos proporcionan información adicional sobre el contenido de un elemento.

Los atributos se añaden dentro de la etiqueta de apertura y se componen de dos partes: el **nombre** del atributo y el **valor** que le vamos a dar.

```
<!DOCTYPE html>
<html lang="es">
<body>
  <h1>Título principal</h1>
  <p>Esto es un párrafo</p>
  <p>Esto es otro párrafo</p>
</body>
</html>
```

El *nombre del atributo* indica que tipo de información extra le estamos añadiendo al elemento. El nombre debe de ir en minúsculas, aunque en HTML5 se permite ponerlos en mayúsculas.

El *valor del atributo* debe de ir entre comillas, pero usando HTML5 no hace falta ponerlas. Los valores de los atributos ya están definidos, por tanto no nos los podemos inventar.

La mayoría de los atributos pueden usarse solamente en algunos elementos, mientras que algunos pocos pueden usarse con cualquier elemento.

A continuación vamos a ver las distintas etiquetas que nos podemos encontrar en HTML.

4. Etiquetas principales de HTML

En esta sección vamos a ver las principales etiquetas de HTML, aquellas que tienen que aparecer en cualquier documento, tanto para mostrar contenido, como para proporcionar información adicional sobre la página.

4.1. Etiqueta DOCTYPE

La etiqueta **DOCTYPE** (*Document Type Declaration*) indica al navegador el tipo y versión del marcado en que se va a escribir el documento.

Esta etiqueta siempre aparece la primera de todas, y de esta forma le hacemos saber al navegador el tipo de documento que va a tener que renderizar.

```
<!DOCTYPE html>
```

Esta etiqueta le indica al navegador que el documento está escrito en HTML, y se va a usar la última versión, HTML5.

4.2. Etiqueta <html>

La etiqueta **html** es un contenedor donde se van a añadir todos los elementos HTML que van a definir como va a ser nuestra página web.

4.3. Etiqueta <head>

La etiqueta **head** contiene todas las propiedades y metadatos del documento. Aquí también se añaden el título de la página, el juego de caracteres, los estilos y los scripts.

Esta etiqueta siempre se encuentra justo después de la etiqueta de apertura **html**.

4.4. Etiqueta <body>

En la etiqueta **body** se encuentran todos los elementos que queremos que se muestren al abrir la página en el navegador.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Título de la página</title>
  </head>
  <body>

  </body>
</html>
```

Esta etiqueta se añade justo después de cerrar la etiqueta **head**.

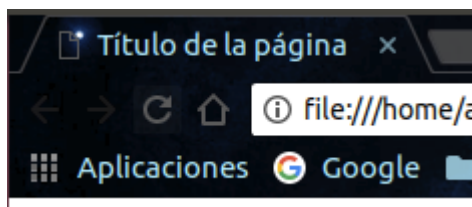
5. Etiquetas que se usan en `<head>`

En esta sección se van a ver las distintas etiquetas que nos podemos encontrar dentro de las etiquetas **head** de una página HTML.

5.1. Etiqueta `<title>`

La etiqueta **title** indica al navegador el contenido que tiene que mostrar en la pestaña de la página.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Título de la página</title>
  </head>
</html>
```



Esta etiqueta siempre se añade entre las etiquetas **head**.

5.2. Etiqueta `<style>`

La etiqueta **style** nos va a permitir añadir los estilos de la página para mejorar la apariencia que tiene.

A esta etiqueta se le debería de añadir el atributo **type** para indicar que los estilos que vamos a usar están escritos en CSS.

```

<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Título de la página</title>
    <style type="text/css">
      p {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1>Título principal</h1>
    <p>Esto es un párrafo</p>
    <p>Esto es otro párrafo</p>
  </body>
</html>

```

Los estilos se añaden entres las dos etiquetas de **style** y estos estilos se aplicarán solo a los elementos de la página en la que se encuentran.

En las antiguas versiones de HTML se podía añadir un atributo **style** en la mayoría de las etiquetas para añadir unos estilos que se iban a aplicar solamente a ese elemento. Esto sigue funcionando en la última versión de HTML pero se recomienda no usarlo.

```

<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Título de la página</title>
    <style type="text/css">
      p {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1 style="color:red">Título principal</h1>
    <p>Esto es un párrafo</p>
    <p>Esto es otro párrafo</p>
  </body>
</html>

```

Esta etiqueta se añade dentro de la etiqueta de **head**.

5.3. Etiqueta **<script>**

La etiqueta **script** nos va a permitir ejecutar código JavaScript en la página para hacerla más dinámica.

A esta etiqueta se le debería de añadir el atributo **type** para indicar que el código que se encuentra entre esas etiquetas es código JavaScript.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Título de la página</title>
    <script type="text/javascript">
      console.log('Un mensaje...');
    </script>
    <style type="text/css">
      p {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1 style="color:red">Título principal</h1>
    <p>Esto es un párrafo</p>
    <p>Esto es otro párrafo</p>
  </body>
</html>
```

En este caso para separar el código JavaScript del código HTML, podemos crear un archivo aparte con todo el JavaScript, e importarlo a través de estas etiquetas para que se use en el documento HTML. Para ello, hay que añadirle el atributo **src** a la etiqueta e indicarle la ruta donde se encuentra el archivo con el JavaScript.

```

<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Título de la página</title>
    <script type="text/javascript" src="main.js"></script>
    <script type="text/javascript">
      console.log('Un mensaje...');
    </script>
    <style type="text/css">
      p {
        color: blue;
      }
    </style>
  </head>
  <body>
    <h1 style="color:red">Título principal</h1>
    <p>Esto es un párrafo</p>
    <p>Esto es otro párrafo</p>
  </body>
</html>

```

Esta etiqueta se puede poner en cualquier sitio del documento HTML, aunque se recomienda ponerlas en el **head**.

6. Etiqueta `<meta>` (PAGINA 191)

Los elementos **meta** contienen información sobre la página. Estos elementos no se muestran en la página, sino que sirven para darle información a los motores de búsqueda, como por ejemplo, quien ha hecho la página, o para indicarles mediante unas palabras clave cual es el contenido de la página...

Estas etiquetas son *etiquetas vacías*, por lo que no tienen una etiqueta de cierre, y se añaden en el **head**.

La información que tienen sobre la página, se añade usando los atributos.

7. Elementos estructurales

En las anteriores versiones de HTML, los desarrolladores de páginas web separaban el contenido similar en secciones usando la etiqueta `<div>`.

Usaban estos elementos para tener en la página web una cabecera, un pie de página, el contenido importante, una barra adicional... Y la forma usada para diferenciar todas estas partes de la página era añadiéndoles atributos **id** y **class** a los elementos.

La estructura más usada en aquella época era la siguiente:

- Una cabecera donde se ponía el logo y un menú de navegación para la página web
- Una sección con el contenido a mostrar en la página (posts, noticias...)
- Una sección lateral donde se podía poner links a páginas similares o a contenido reciente, un buscador, publicidad...
- Un pie de página para poner las preguntas frecuentes, información de contacto,

```
<!DOCTYPE html>
<html>
  <!-- ... -->
  <body>
    <div id="header">
      <div id="navbar">
        <!-- ... -->
      </div>
    </div>
    <div id="section">
      <div class="article">
        <!-- ... -->
      </div>
      <div class="article">
        <!-- ... -->
      </div>
    </div>
    <div id="sidebar">
      <!-- ... -->
    </div>
    <div id="footer">
      <!-- ... -->
    </div>
  </body>
</html>
```

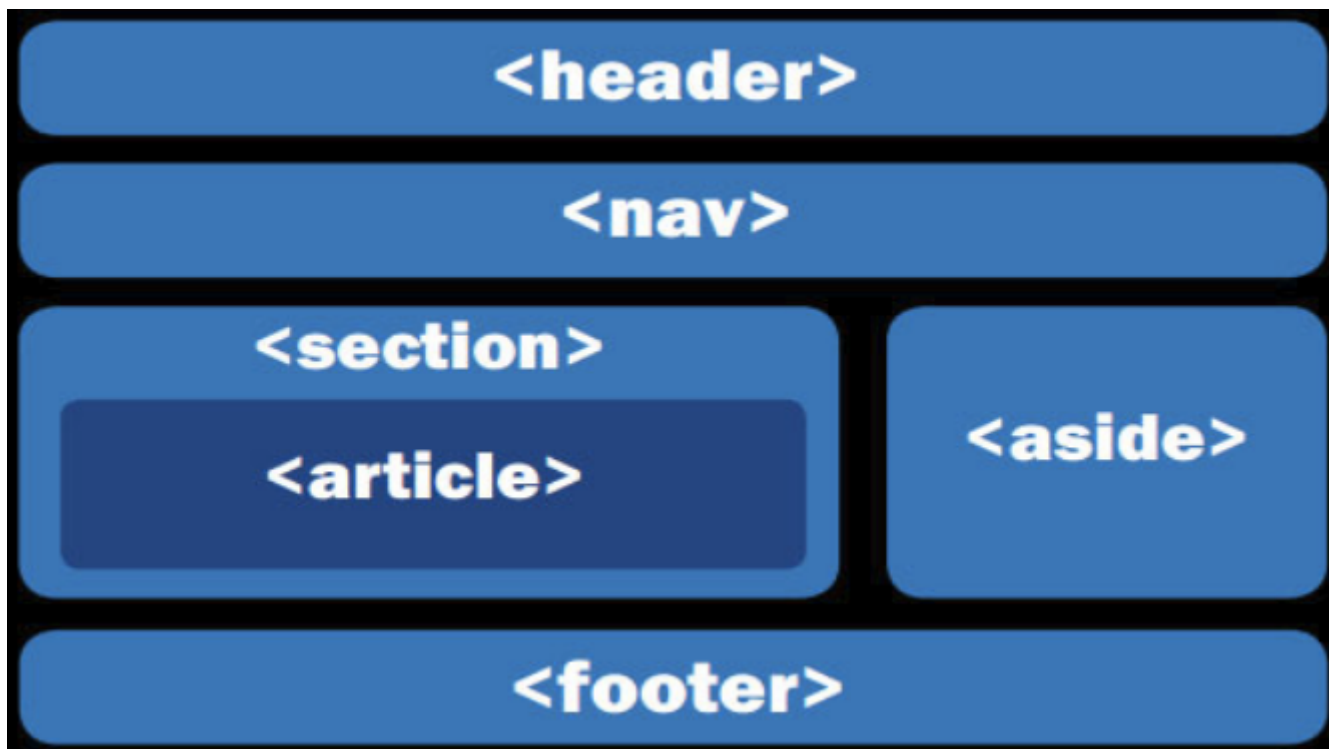
La versión de HTML5 ha introducido un conjunto de etiquetas nuevas que permiten dividir la página en distintas partes. El nombre de estas nuevas etiquetas, indican el tipo de contenido que se va a encontrar dentro de ellas.

El anterior código se vería de la siguiente forma al usar estas nuevas etiquetas.

```
<!DOCTYPE html>
<html>
  <!-- ... -->
  <body>
    <header>
      <nav>
        <!-- ... -->
      </nav>
    </header>
    <section>
      <article>
        <!-- ... -->
      </article>
      <article>
        <!-- ... -->
      </article>
    </section>
    <aside>
      <!-- ... -->
    </aside>
    <footer>
      <!-- ... -->
    </footer>
  </body>
</html>
```

Estas etiquetas surgen porque la mayor parte de las páginas web usaban la misma estructura de elementos **div** con los mismos atributos *id* y *class*. Se decidió estudiar cuales eran los valores de los atributos más usados para cada una de las secciones y se crearon estas nuevas etiquetas.

De esta forma se reducía el uso de elementos *div* y al leer el código HTML es más fácil de ver cual es el contenido de cada etiqueta.



7.1. Etiqueta `<header>`

La etiqueta **header** representa la sección donde vamos a poner el nombre de la página web, empresa, aplicación..., un logo, el menú de navegación...

```
<header>
  
  <h1>Mi página web</h1>
  <nav>
    <ul>
      <li><a href="inicio.html">Inicio</a></li>
      <li><a href="clases.html">Clases</a></li>
      <li><a href="sobre-mi.html">Sobre mi</a></li>
      <li><a href="contacto.html">Contacto</a></li>
    </ul>
  </nav>
</header>
```

Esta etiqueta también puede usarse dentro de las etiquetas **article** y **section** para añadir por ejemplo un título y fecha del contenido que muestran estos contenedores.

7.2. Etiqueta `<footer>`

La etiqueta **footer** representa la sección donde vamos a poner el copyright, los enlaces a las políticas de privacidad, terminos y condiciones, preguntas frecuentes...

```

<footer>
  <ul>
    <li><a href="politicas-privacidad.html">Políticas de privacidad</a></li>
    <li><a href="politicas-privacidad.html"></a></li>
    <li><a href="faq.html">Preguntas frecuentes</a></li>
  </ul>
  Copyright: 2010 Mi página web
</footer>

```

Esta etiqueta también puede usarse dentro de las etiquetas **article** y **section** para añadir por ejemplo enlaces para compartir el contenido en las redes sociales, o el autor del contenido que se está mostrando...

7.3. Etiqueta `<nav>`

El elemento **nav** se usa para añadir el menú de navegación principal de la página.

En este elemento se suele añadir enlaces que nos van a permitir navegar entre las distintas secciones que tiene nuestra web.

```

<nav>
  <ul>
    <li><a href="inicio.html">Inicio</a></li>
    <li><a href="clases.html">Clases</a></li>
    <li><a href="sobre-mi.html">Sobre mi</a></li>
    <li><a href="contacto.html">Contacto</a></li>
  </ul>
</nav>

```

Esta etiqueta no tiene porque ponerse unicamente dentro de la etiqueta **header**. A veces también es usada para añadir los enlaces que se encuentran al pie de la página, aunque esto es menos frecuente.

7.4. Etiqueta `<article>`

La etiqueta **article** actua como un contenedor para cualquier pieza de contenido que pueda ser independiente, como un comentario, un post, una entrada de un blog...

```

<article>
  <header>
    <h3>El Valencia sigue invicto</h3>
  </header>
  <p>Una noticia ...</p>
</article>

```

Los elementos *article* se pueden anidar unos dentro de otros, es decir que podemos tener una

entrada de un blog (elemento *article*) y dentro de este una serie de comentarios (cada uno de ellos, un elemento *article*).

7.5. Etiqueta `<aside>`

La etiqueta **aside** puede tener distintos propósitos dependiendo de donde se encuentre contenida.

En caso de estar dentro de un elemento *article*, entonces debería de contener información relacionada con el contenido del *article* pero sin ser demasiado importante para el contenido de este. Por ejemplo, un glosario.

Mientras que cuando se encuentra fuera de una etiqueta *article*, entonces el contenido del *aside* debería de ser contenido que esté relacionado con la página entera, como por ejemplo, enlaces a páginas de contenido similar, enlaces a los últimos contenidos de nuestra página, una caja de búsqueda...

```
<aside>
  <h3>Últimos artículos</h3>
  <a href="tutorial-jquery.html">Tutorial jQuery</a>
  <a href="action-cable-ror.html">Action Cable en Ruby on Rails</a>
  <a href="redis.html">Redis</a>
</aside>
```

7.6. Etiqueta `<section>`

Los elementos **section** se encargan de agrupar contenido que se encuentra relacionado y normalmente estas secciones tienen un título. Por ejemplo, una sección podría ser el contenedor donde se muestran los productos más vendidos del momento, o las últimas noticias.

Los sections se suelen usar con los elementos *article*. Un *section* puede usarse para agrupar distintos elementos *article* cuyo contenido trata de un mismo tema. Y también pueden servir para separar el contenido de un *article* extenso en distintas secciones.

```
<section class="futbol">
  <article>
    <header>
      <h3>El Valencia sigue invicto</h3>
    </header>
    <p>Una noticia ...</p>
  </article>
  <article>
    <header>
      <h3>Bakambu deja el Villarreal</h3>
    </header>
    <p>Otra noticia...</p>
  </article>
</section>
```

Este elemento no debería de usarse como contenedor de toda una página, ese trabajo es del elemento *div*.

7.7. Ejercicio

- Crear la estructura de la página 'www..com' usando estas etiquetas estructurales

8. Elementos de texto

8.1. Elementos de título: `<h1>` - `<h6>`

HTML tiene 6 elementos que permiten mostrar **títulos** y **subtítulos**. Estos elementos son `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`.

Cada uno de estos elementos cambian el color de la letra y su tamaño.

El elemento **h1** se usa para los títulos principales de la página, y es el que pone el tamaño de la letra más grande.

El elemento **h2** se suele usar para los subtítulos y pone la letra un poco más pequeña que el **h1**.

En caso de haber más secciones en la página entonces se van añadiendo el resto de los elementos, hasta llegar al **h6** que es el que pone la letra más pequeña de todos.

```
<h1>Título 1</h1>
<h2>Título 2</h2>
<h3>Título 3</h3>
<h4>Título 4</h4>
<h5>Título 5</h5>
<h6>Título 6</h6>
```

El tamaño del texto de estos elementos, el color y el tipo de letra se puede cambiar usando CSS.

8.2. Etiqueta `<p>`

La etiqueta **p** nos permite añadir párrafos al documento. Cada párrafo que queramos añadir habrá que ponerlo entre las etiquetas *p*.

El navegador mostrará cada párrafo en una nueva línea con algo de espacio entre un párrafo y cualquier elemento que vaya después.

```
<p>Minions ipsum bappleees belloo! Uuuhhh. Poopayee uuuhhh hahaha bananaaaa potatoooo  
aaaaaah potatoooo jeje. Me want bananaaaa! jiji me want bananaaaa! Me want bananaaaa!  
Hahaha tulaliloo bananaaaa la bodaaa belloo! Para tú daa potatoooo baboiii tank yuuu!  
Ti aamoo! Aaaaaah underweaaar tatata bala tu bananaaaa tatata bala tu bananaaaa  
baboiii bananaaaa. Tatata bala tu potatoooo chasy daa tank yuuu!</p>
```

```
<p>Underweaaar po kass po kass para tú la bodaaa. Underweaaar jeje baboiii la bodaaa  
para tú la bodaaa ti aamoo! Pepete. Belloo! baboiii la bodaaa baboiii poulet tikka  
masala para tú hana dul sae tatata bala tu baboiii wiiiii. Poulet tikka masala bee do  
bee do bee do para tú wiiiii. Aaaaaah underweaaar poulet tikka masala potatoooo  
belloo!</p>
```

8.3. Etiqueta ****

El texto que se encuentre dentro de la etiqueta **b**, aparecerá en negrita.

Esta etiqueta nos va a permitir mostrar las palabras más importantes de un texto.

```
<p>Minions ipsum bappleees belloo! Uuuhhh. Poopayee uuuhhh hahaha bananaaaa potatoooo  
aaaaaah potatoooo jeje. Me want bananaaaa! jiji me want bananaaaa! Me want bananaaaa!  
Hahaha tulaliloo bananaaaa la bodaaa belloo! Para tú daa potatoooo baboiii tank yuuu!  
<b>Ti aamoo!</b> Aaaaaah underweaaar tatata bala tu bananaaaa tatata bala tu bananaaaa  
baboiii bananaaaa. Tatata bala tu potatoooo chasy daa tank yuuu!</p>
```

8.4. Etiqueta **<i>**

El texto que se encuentre dentro de la etiqueta **i**, aparecerá en cursiva.

Esta etiqueta nos va a permitir mostrar en un texto contenido como términos técnicos, nombres, palabras extrañas, o cualquier texto que normalmente se muestra en cursiva.

```
<p><i>Minions ipsum</i> bappleees belloo! Uuuhhh. Poopayee uuuhhh hahaha bananaaaa  
potatoooo aaaaaah potatoooo jeje. Me want bananaaaa! jiji me want bananaaaa! Me want  
bananaaaa! Hahaha tulaliloo bananaaaa la bodaaa belloo! Para tú daa potatoooo baboiii  
tank yuuu! <b>Ti aamoo!</b> Aaaaaah underweaaar tatata bala tu bananaaaa tatata bala  
tu bananaaaa baboiii bananaaaa. Tatata bala tu potatoooo chasy daa tank yuuu!</p>
```

8.5. Etiqueta **<sup>**

La etiqueta **sup** se usa para mostrar el contenido que debería de aparecer como sobreíndice.

Puede usarse para mostrar formulas matemáticas.

```
<p>(a+b)<sup>2</sup>=a<sup>2</sup>+2ab+b<sup>2</sup></p>
```

8.6. Etiqueta `<sub>`

La etiqueta **sub** se usa para mostrar el contenido que debería de aparecer como subíndice.

Puede usarse para mostrar formulas químicas.

```
<p>Dioxido de carbono: C0<sub>2</sub></p>
```

8.7. Etiqueta `
`

La etiqueta **br** nos permite añadir saltos de línea en el texto.

Por ejemplo, los párrafos añaden un salto de línea al cerrarse la etiqueta, pero podemos querer añadir un salto de línea dentro de un párrafo, y con esta etiqueta es con lo que habría que hacerlo.

```
<p>Underweaaaar po kass po kass para tú la bodaaa. Underweaaaar jeje baboiii la bodaaa  
para tú la bodaaa ti aamoo! Pepete. Belloo! baboiii la bodaaa baboiii poulet tikka  
masala para tú hana dul sae tatata bala tu baboiii wiiiiii. <br> Poulet tikka masala  
bee do bee do bee do para tú wiiiiii. Aaaaaah underweaaaar poulet tikka masala potatoooo  
belloo!</p>
```

Este es un elemento vacío por lo que la etiqueta que no se cierra.

Los **elementos vacíos** son aquellos que no necesitan contenido, por lo que no es necesario poner la etiqueta de apertura y la de cierre. Estos elementos constan de una sola etiqueta y nos la podemos encontrar de dos formas:

- `
`
- `
`

8.8. Etiqueta `<hr>`

La etiqueta **hr** muestra una línea horizontal.

Se suele usar para separar temas, por ejemplo, cuando en un libro se pasa de un capítulo a otro.

```
<p>(a+b)<sup>2</sup>=a<sup>2</sup>+2ab+b<sup>2</sup></p>  
<hr>  
<p>Dioxido de carbono: C0<sub>2</sub></p>
```

Este es un elemento vacío por lo que la etiqueta que no se cierra.

8.9. Espacios en blanco

Algunas veces, cuando estamos escribiendo el código HTML, queremos añadir espacios en blanco

para separar el contenido un poco más o para que la línea del primer párrafo empiece un poco más a la derecha que el resto de líneas.

Pero cuando el navegador se encuentra que en el texto hay más de un espacio en blanco, lo que hace es que solo va a mostrar un espacio. Esto se conoce como **colapsado de espacios blancos**.

Más adelante veremos que hay que hacer para añadir más espacios.

9. Etiqueta de enlace: `<a>`

Los enlaces son una funcionalidad importante de cualquier página web porque son los que nos van a permitir navegar de una página a otra.

Nos podemos encontrar con distintos tipos de enlaces:

- Enlaces de una página a otra página (de distinta web)
- Enlaces de una página a otra página (de la misma web)
- Enlaces de una parte de la web a otra dentro de la misma página
- Enlaces que abren la página en una nueva ventana del navegador
- Enlaces que abren el correo para enviar un email

Los enlaces se crean usando las etiquetas **a**. Los usuarios usuario pueden pulsar en el texto que se añade entre la etiqueta de apertura y de cierre.

Para decir a que página queremos navegar, tenemos que indicar la *URL* como valor para el atributo **href**.

El texto entre las etiquetas **a** se conoce como el texto del enlace. En este texto es donde deberíamos explicar a que página se va a navegar cuando un usuario pulse sobre el.

Escribir textos en los enlaces que sean claros ayudan a los usuarios a encontrar más fácilmente lo que quieren. Da una mejor impresión sobre nuestra página y ayuda a la gente que usa los *screen readers*.

Para escribir buenos textos y ponerlos en los enlaces deberíamos de usar palabras que usarían los usuarios en nuestra página para encontrar lo que quieren, es decir, si tenemos un enlace que nos envía a una sección con hoteles, en lugar de poner algo como *donde alojarme* habría que poner un texto como *hoteles*.

El texto de los enlaces se muestran por defecto en azul y con el texto subrayado.

9.1. Enlaces a páginas externas

Cuando queremos navegar a una página externa, el valor del atributo **href** tiene que ser la dirección completa de la web, es decir, la **ruta absoluta**.

```
<a href="https://www.netflix.com">Netflix</a>
```

9.2. Enlaces a distintas páginas dentro del mismo sitio

Cuando vamos a ir a distintas páginas que se encuentran dentro del mismo sitio, no hace falta especificar la *ruta absoluta*. En este caso podemos usar la **ruta relativa**, aquella a la que no se le necesita especificar el nombre del dominio de la página.

Si todas las páginas HTML de nuestro sitio se encuentran en la misma carpeta, solo hay que añadir al atributo **href** el nombre de la página a la que queremos ir.

```
<a href="pagina1.html">Página 1</a>
```

Pero si las tenemos repartidas en distintas carpetas entonces tenemos que añadir donde se encuentra la página en relación a la página actual.

```
<a href="carpeta1/pagina2.html">Página 2</a>
```

9.3. Enlaces a distintas secciones de la misma página

Nuestra página puede contener una lista de secciones bastante grande, y para ir hasta la última tendremos que ir haciendo scroll hasta llegar a ella. Esto se puede evitar creando una serie de enlaces, uno por cada sección y apuntando a esa sección para que al pulsar sobre el enlace no navegue hasta otra página sino que nos lleve a la sección correspondiente.

Antes de poner los enlaces hay que identificar los puntos a los que queremos navegar. Esto lo vamos a hacer poniendo el atributo **id** a cada sección. El valor de este atributo debe de empezar por una letra o por **_**, pero no puede empezar por un número, y también hay que tener en cuenta que este atributo debería de ser único en toda la página.

Una vez tenemos los identificadores puestos, creamos los enlaces, donde vamos a poner el atributo **href**. A este atributo le vamos a dar el valor del *id* precedido por **#** de la sección a la que queremos navegar.

```
<a href="pagina1.html">Página 1</a>
<a href="#seccion1">Seccion 1</a>
<a href="#seccion2">Seccion 2</a>
<a href="#seccion3">Seccion 3</a>
<section>
  <h2 id="seccion1">Seccion 1</h2>
  <p>...</p>
</section>
<section>
  <h2 id="seccion2">Seccion 2</h2>
  <p>...</p>
</section>
<section>
  <h2 id="seccion3">Seccion 3</h2>
  <p>...</p>
</section>
```

En caso de querer ir a un punto específico de una página distinta a la actual, también podemos hacerlo. Para ello solo tenemos que poner como valor del atributo `href` la dirección de la página a la que queremos ir, seguida del símbolo `#` y el identificador de la sección a la que queremos navegar.

```
<a href="carpeta1/pagina2.html#seccion3">Página 2</a>
```

9.4. Enlaces que abren la página en otra ventana

Si queremos que la página se abra en una nueva pestaña entonces tenemos que añadir el atributo `target` con el valor `_blank`.

Estos enlaces se suelen usar para abrir páginas externas a la nuestra. De esta forma es más fácil de que el usuario vuelva a nuestra página, pues solo tiene que cerrar la otra pestaña y ya.

```
<a href="https://www.amazon.com" target="_blank">Amazon</a>
```

9.5. Enlaces a correos

Para crear un enlace que se encargue de abrir el correo con la dirección del destinatario dada tenemos que añadir el atributo `href` y cuyo valor tiene que empezar por `mailto:` seguido de la dirección de correo a la que queremos enviar el email.

```
<a href="mailto:example@gmail.com">Correo a example@gmail.com</a>
```

9.6. Ejercicio

- Crea las páginas de:
 - Inicio
 - Sobre mi
 - Contacto
 - Añade un enlace en la página de Contacto que abra el correo para que sea fácil de contactarnos
- Añade enlaces para poder navegar entre ellas
- Añade una página con información de distintas series y al inicio una serie de enlaces que nos manden directamente al lugar de la página donde se encuentra su información

9.7. Ejercicio

- Buscar en la página de 'www.meetup.com' 10 eventos que ocurran próximamente
 - Crear una página inicial en la que se muestren todos los eventos junto a un enlace que va a llevarnos a la página detallada del evento
 - En la página de cada evento mostrar toda la información disponible

10. Elementos de listas

Cuando estamos creando una página web, la mayor parte de las veces vamos a mostrar elementos en una lista, por ejemplo, para mostrar una lista de tareas, una lista de la compra, una serie de requisitos...

HTML nos permite crear tres tipos de listas: ordenadas, desordenadas y de definiciones.

10.1. Etiqueta ``

El elemento `ul` es un contenedor donde nos vamos a encontrar una lista de elementos en la que el orden en que se muestran no tiene importancia.

10.2. Etiqueta ``

El elemento `ol` es un contenedor donde nos vamos a encontrar los distintos elementos de la lista ordenados. En este caso si que se le da importancia al orden.

10.3. Etiqueta ``

El elemento `li` representa un elemento de los que se muestran en las listas.

Cada elemento `li` que pongamos dentro de una lista ordenada se van a mostrar ordenados por número, letra, números romanos...). Según el nivel de indentación añade un tipo de orden.


```
<h3>Lista compra</h3>
<ul>
  <li>Leche</li>
  <li>Patatas</li>
  <li>Zumo de naranja</li>
  <li>Lechuga</li>
  <li>Tomate</li>
</ul>
```

```
<h3>Clasificación Liga Santander</h3>
<ol>
  <li>Valencia</li>
  <li>Barcelona</li>
  <li>Real Madrid</li>
  <li>Villarreal</li>
</ol>
```

El tipo de carácter (círculo, cuadrado, números, letras...) que aparece delante de cada elemento que se muestra en la lista se puede cambiar usando los estilos de CSS.

10.4. Etiqueta `<dl>`

El elemento **dl** es una lista de definiciones donde podemos poner distintas palabras con sus respectivos significados.

Dentro de esta etiqueta vamos a añadir pares de elementos **dt** y **dd**.

10.5. Etiqueta `<dt>`

La etiqueta **dt** es el contenedor donde se añade el término que se va a definir.

En caso de que se vayan a definir dos términos al mismo tiempo, se añaden dos elementos de este tipo seguidos.

10.6. Etiqueta `<dd>`

La etiqueta **dd** es el contenedor donde se va a añadir la definición del término.

```
<h3>Términos químicos</h3>
<dl>
  <dt>Átomo</dt>
  <dd>Partícula indivisible</dd>
  <dt>Base</dt>
  <dd>Elemento o compuesto capaz de aceptar protones</dd>
</dl>
```

10.7. Listas anidadas

HTML nos permite anidar listas, es decir, poner una lista dentro de otra añadiendo el segundo elemento lista (`ul`, `ol`) dentro de un elemento `li`.

El navegador detecta que la lista está dentro de otra lista, por lo que cambia el símbolo que hay delante de cada ítem de la lista.

```
<ul>
  <li>Cafes
    <ul>
      <li>Con leche</li>
      <li>Solo con hielo</li>
      <li>Cortado</li>
    </ul>
  </li>
</ul>
```

10.8. Ejercicio

- Crear una página HTML que muestre la receta de algún postre.
 - Mostrar los ingredientes
 - Mostrar los pasos a seguir

11. Elementos de Tablas

En las páginas web a veces hay información que la mejor forma de mostrarla es haciéndolo dentro de una tabla. Por ejemplo, la clasificación de la liga, las notas del colegio...

Cuando vamos a representar información en una tabla, hay que pensar en que esta va a tener filas y columnas, y que las filas y las columnas pueden combinarse, es decir, podemos tener una celda que ocupe dos columnas.

Las tablas nos van a permitir representar información sobre 2 ejes. Y donde se junta una fila con una columna, lo vamos a llamar **celda**.

A continuación vamos a ver los elementos de HTML que nos van a permitir crear las tablas.

11.1. Etiqueta `<table>`

La etiqueta **table** es la que se usa para crear una tabla.

El contenido de la tabla se va a ir creando fila a fila.

11.2. Etiqueta `<tr>`

La etiqueta **tr** representa el inicio de una fila.

Esta etiqueta contiene las celdas que se verán a continuación.

11.3. Etiqueta `<td>`

La etiqueta **td** representa una celda de la tabla.

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

11.4. Etiqueta `<th>`

El elemento **th** se usa como una celda, pero su propósito es representar la cabecera de una columna o de una fila.

Aunque la celda no tenga contenido deberíamos representarla como una celda vacía (*th* o *td*) para que la tabla se muestre correctamente.

```
<table>
  <tr>
    <th>Num1</th><th>Num2</th><th>Num3</th>
  </tr>
  <tr>
    <td>1</td><td>2</td><td>3</td>
  </tr>
  <tr>
    <td>4</td><td>5</td><td>6</td>
  </tr>
  <tr>
    <td>7</td><td>8</td><td>9</td>
  </tr>
</table>
```

El contenido de estas etiquetas se muestra en negrita y centradas en la celda.

Estos elementos sirven de ayuda a los motores de búsqueda a la hora de indexar la página, y también a los usuarios que usan los screen readers.

11.5. Etiqueta <thead>

La etiqueta **thead** nos indica que el contenido que se encuentre aquí, se corresponde con la cabecera de la tabla.

11.6. Etiqueta <tbody>

La etiqueta **tbody** nos indica que el contenido que se encuentre aquí, se corresponde con el contenido importante, los datos que queremos ver.

11.7. Etiqueta <tfoot>

La etiqueta **tfoot** nos indica que el contenido que se encuentre aquí, se corresponde con el pie de la tabla. Aquí es donde se suelen mostrar los resultados.

```
<table>
  <thead>
    <tr>
      <th></th><th>Ingresos</th><th>Gastos</th><th>Total</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Enero</th><td>200</td><td>65</td><td>135</td>
    </tr>
    <tr>
      <th>Febrero</th><td>380</td><td>100</td><td>280</td>
    </tr>
    <tr>
      <th>Marzo</th><td>40</td><td>100</td><td>-60</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Total</th><td></td><td></td><td>355</td>
    </tr>
  </tfoot>
</table>
```

11.8. Combinando filas y columnas

A veces vamos a necesitar combinar filas porque el contenido de una celda abarca dos filas o más, y lo mismo ocurre con las columnas.

Las celdas, tanto las **th** como las **td** tienen unos atributos que permiten hacerlo. Estos atributos son:

- **colspan**: combina columnas

- rowspan: combina filas

El valor que reciben ambos atributos son el número de filas o columnas que se van a combinar.

```
<table>
  <thead>
    <tr>
      <th></th><th>Ingresos</th><th>Gastos</th><th>Total</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Enero</th><td>200</td><td>65</td><td>135</td>
    </tr>
    <tr>
      <th>Febrero</th><td>380</td><td>100</td><td>280</td>
    </tr>
    <tr>
      <th>Marzo</th><td>40</td><td>100</td><td>-60</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th colspan="3">Total</th><td>355</td>
    </tr>
  </tfoot>
</table>
```

11.9. Ejercicio

- Mostrar en una tabla el horario de la televisión de 4 cadenas
 - Las columnas serán los programas que hay en cada televisión
 - Las filas serán los programas que hay en una franja de horas
 - Cada fila representará media hora

12. Elementos de Formularios

Los formularios nos van a permitir obtener información de los usuarios de nuestra web. Los elementos que se usan junto a los formularios nos van a facilitar este trabajo, puesto que hay elementos para recoger cualquier tipo de dato, algunos tienen validaciones...

Los formularios son uno de los elementos más importantes de las páginas web puesto que se usan para loguear un usuario, registrarse, buscar en la web, subscribirnos a una lista de correo, dar los datos para pagar...

Tenemos distintos elementos que podemos usar para obtener todos esos datos:

- Elementos para rellenar con texto
- Elementos para marcar diferentes opciones
- Elementos para enviar el formulario

Todos estos elementos van a tener una serie de atributos que nos van a ayudar a enviar la información que se ha rellenado.

12.1. Etiqueta `<form>`

Todos los campos del formulario se encuentran entre las etiquetas de apertura y cierre de **form**. Este elemento es el que representa el formulario.

Esta etiqueta suele llevar consigo dos atributos:

- `action`
- `method`

Cada formulario tiene un atributo **action** que es la URL de la página del servidor que va a recibir la información rellanada en el formulario cuando este se haya enviado.

El otro atributo es el **method** que indica mediante que método se va a enviar la información (*get* o *post*).

Con el método **get**, los valores del formulario se añaden al final de la URL especificada en el atributo **action** en forma de parámetros de búsqueda. Este método se suele usar para formularios pequeños (marcar checkboxes para realizar búsquedas), o para pedir datos del servidor.

```
<form action="http://www.ejemplo.com/buscar.rb" method="get">

</form>
```

Mientras que con el método **post**, los valores se envían en las cabeceras HTTP. Este método se suele usar para formularios largos que van a guardar información o la van a borrar, formularios que contienen datos sensibles (como contraseñas), subir archivos al servidor...

```
<form action="http://www.ejemplo.com/tarea.rb" method="post">

</form>
```

En caso de no especificar el método que se va a usar, por defecto se envían como **get**.

También puede añadirse el atributo **id** para poder acceder al formulario de una forma sencilla y comprobar que todos los campos cumplen las validaciones (usando JavaScript).

12.2. Etiqueta `<input>`

El elemento **input** se usa para crear diferentes campos para rellenar información. El atributo **type** es el que define el tipo de campo que se va a crear.

Esta etiqueta es una etiqueta vacía por lo que no necesita la etiqueta de cierre.

12.2.1. `type="text"`

Cuando el atributo **type** obtiene el valor de **text**, entonces crea un campo de texto simple.

A este campo se le puede añadir el atributo **name**, cuyo valor será con el que se va a enviar el valor del input (*la clave*).

```
<input type="text" name="usuario">
```

A este campo se le pueden añadir los atributos **minlength** y **maxlength** para controlar la longitud del texto. En caso de que no se encuentre entre esos dos valores, la validación dará un error.

```
<input type="text" name="usuario" placeholder="Robb" minlength="3" maxlength="10">
```

12.2.2. `type="password"`

Cuando el atributo **type** obtiene el valor de **password**, entonces crea un campo de texto simple, con la diferencia de que lo que vayamos a escribir aparecerá oculto (con *).

Que el texto que escribamos en este campo aparezca oculto por los * no significa que los datos se vayan a enviar de forma segura al servidor.

A este campo se le puede añadir el atributo **name**, cuyo valor será con el que se va a enviar el valor del input (*la clave*).

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

A este campo se le pueden añadir los atributos **minlength** y **maxlength** para controlar la longitud del texto. En caso de que no se encuentre entre esos dos valores, la validación dará un error.

```
<input type="password" name="password" placeholder="Stark" minlength="8" maxlength="12">
```

12.2.3. `type="radio"`

Cuando el atributo **type** obtiene el valor de **radio**, entonces crea un campo sobre el que pulsar. Estos elementos nos permiten seleccionar una de todas las opciones que haya. Si tenemos pulsada una, y seleccionamos otra distinta, la anterior quedará sin seleccionar.

A este campo se le puede añadir el atributo `name`, cuyo valor será con el que se va a enviar el valor del input (*la clave*). Además este es el atributo que controla que solo sea seleccionable una de todas las opciones, y para ello todas tienen que tener el mismo valor en este atributo.

El atributo `value` es el que indica el valor que se va a enviar al servidor, cuando se envíe el formulario.

Otro atributo que se le puede añadir es `checked`, el cual indica que el elemento está marcado.

```
<p>Que te gusta más?:  
  <input type="radio" name="hobby" value="series" checked> Series  
  <input type="radio" name="hobby" value="cine"> Cine  
  <input type="radio" name="hobby" value="libros"> Libros  
</p>
```

12.2.4. `type="checkbox"`

Cuando el atributo `type` obtiene el valor de `checkbox`, entonces crea un campo sobre el que pulsar. Estos elementos nos permiten seleccionar varias de las opciones que haya.

A este campo se le puede añadir el atributo `name`, cuyo valor será con el que se va a enviar el valor del input (*la clave*). Además este atributo tiene que tener el mismo valor en todos los checkboxes que estén relacionados.

El atributo `value` es el que indica el valor que se va a enviar al servidor, cuando se envíe el formulario.

Otro atributo que se le puede añadir es `checked`, el cual indica que el elemento está marcado.

```
<p>Que tipo de música te gusta?  
  <input type="checkbox" name="musica" value="rock">Rock  
  <input type="checkbox" name="musica" value="pop">Pop  
  <input type="checkbox" name="musica" value="heavy">Heavy  
  <input type="checkbox" name="musica" value="rap">Rap  
</p>
```

12.2.5. `type="submit"`

Cuando el atributo `type` obtiene el valor de `submit`, se crea un botón que se usa para enviar el formulario.

Este tipo de input no es necesario que contenga el atributo `name`. Mientras que el atributo `value` indica el texto que se muestra en el botón, el cual si no se le indica, suele coger por defecto *Submit*.

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


12.2.6. type="image"

Cuando el atributo **type** obtiene el valor de **image**, se crea un botón con una imagen y actúa como el *input* de tipo **submit**.

Esta etiqueta acepta los atributos **src** (ruta donde se encuentra la imagen que se va a mostrar en el botón), **width** (ancho del botón), **height** (alto de botón) y **alt** (texto alternativo en caso de que la imagen no se pueda obtener).

```
<input type="image" src="boton.png" alt="Una imagen como botón" width="150" height="50">
```

12.2.7. type="number"

Cuando el atributo **type** obtiene el valor de **number**, se crea un campo donde solo vamos a poder escribir números.

Esta etiqueta tiene una validación por la que no nos va a permitir introducir caracteres no numéricos.

```
<label>Número:  
  <input type="number">  
</label>
```

Si le damos a las flechas que aparecen dentro del campo, se va sumando o restando los números de uno en uno. Esto se puede cambiar con el atributo **step**, el cual nos permite decir cuanto se tiene que incrementar o decrementar el número al pulsar en esas flechas. Si intentamos poner un número que no se encuentra en esos pasos, fallará la validación.

```
<label>Número de 2 en 2:  
  <input type="number" step="2">  
</label>  
<label>Número de centesima a centesima:  
  <input type="number" step="0.01">  
</label>
```

También se le puede indicar que el número tiene que estar en un cierto rango, añadiendo los atributos **min** y **max**. Si intentamos poner un número que no se encuentre en ese rango, nos fallará la validación.

```
<label>Número entre 0 y 10:  
  <input type="number" min="0" max="10">  
</label>
```

12.2.8. `type="range"`

Cuando el atributo `type` obtiene el valor de **range**, se crea *slider* donde podremos mover el elemento que indica el valor.

Estos elementos se suelen utilizar para controlar el volumen y la barra de tiempo en la que se encuentra un vídeo o una canción.

A esta etiqueta se le puede poner los atributos `min` y `max` para poner un límite a los valores.

```
<label>Rango de salario:  
  <input type="range" min="0" max="10">  
</label>
```

12.2.9. `type="date"`

Cuando el atributo `type` obtiene el valor de **date**, se crea un campo con la fecha, y al pulsar en ese campo sale un *datepicker* donde podremos seleccionar la fecha que queramos.

En caso de querer darle una fecha como valor por defecto tendremos que añadirla al atributo `value` (tiene que cumplir el formato `yyyy-mm-dd`). En caso contrario se mostrará por defecto la fecha actual.

```
<label>Fecha de nacimiento:  
  <input type="date" value="2000-11-01">  
</label>
```

12.2.10. `type="email"`

Cuando el atributo `type` obtiene el valor de **email**, se crea un campo de texto en el que se va a validar que el texto introducido coincide con la forma que tiene cualquier email. En caso de que no sea correcta, al enviar el formulario nos avisará de que el email no es correcto, y no se enviará la información hasta que se cumpla esa validación.

```
<label>Email:  
  <input type="email">  
</label>
```

12.2.11. `type="url"`

Cuando el atributo `type` obtiene el valor de **url**, se crea un campo de texto en el que se va a validar que el texto introducido coincide con la forma que tiene cualquier *URL*. En caso de que no sea correcta, al enviar el formulario nos avisará de que no es correcta, y no se enviará la información hasta que se cumpla esa validación.

```
<label>Url:
  <input type="url">
</label>
```

12.2.12. type="file"

Cuando el atributo **type** obtiene el valor de **file**, entonces el usuario puede subir archivos (mp3, pdf, imagenes, vídeos...) que se encuentran en su dispositivo.

Este elemento crea un campo con un botón, y al pulsar sobre el botón, se abre una ventana donde podremos seleccionar el archivo que se va a subir.

Solo se pueden enviar archivo usando el método **post**.

```
<input type="file" name="tutorial">
```

12.3. Etiqueta <label>

La etiqueta **label** sirve para mostrar cual es el proposito del contenido que se va a rellenar junto a esta etiqueta.

Hay dos formas de usarla:

- Meter el input dentro de esta etiqueta
- Poner el input justamente después de esta etiqueta, y en este caso tendremos que usar el atributo **for**

El atributo **for** indica a que campo pertenece esta etiqueta, y como solo debería de pertenecer a un campo único, se le da el valor del atributo **id** de dicho campo al que pertenece.

```
<label>Nombre
  <input type="text" name="nombre" id="nombre">
</label>
<label for="apellidos">Apellidos</label>
<input type="text" name="apellidos" id="apellidos">
```

Esta etiqueta se puede poner en distintos sitios dependiendo de con que etiqueta se va a relacionar.

- A la izquierda o como contenedor de los inputs, textareas, selects y subida de ficheros.
- A la derecha de los checkboxes y de los radios buttons.

12.4. Etiqueta <fieldset>

Con la etiqueta **fieldset** podemos agrupar la información que está relacionada. Suele ser muy útil en formularios largos, para separarlos en distintas secciones.

```
<fieldset>
  <label for="nombre">Nombre
    <input type="text" name="nombre" id="nombre">
  </label>
  <label for="apellidos">Apellidos
    <input type="text" name="apellidos" id="apellidos">
  </label>
</fieldset>
```

12.5. Etiqueta <legend>

La etiqueta **legend** se usa junto a la etiqueta **fieldset** y sirve como leyenda para identificar el proposito de la información que se encuentra en ese grupo de campos del formulario.

Esta etiqueta se pone justamente después de la etiqueta de apertura del *fieldset*.

```
<fieldset>
  <legend>Información del usuario</legend>
  <label for="nombre">Nombre
    <input type="text" name="nombre" id="nombre">
  </label>
  <label for="apellidos">Apellidos
    <input type="text" name="apellidos" id="apellidos">
  </label>
</fieldset>
```

12.6. Etiqueta <button>

La etiqueta **button** actua como un botón, y de esta forma vamos a poder controlar cuando enviar el formulario. Mediante un script de JavaScript, podemos detectar cuando se pulsa este botón, y de esta forma realizar una serie de acciones antes de realizar el envio de la información al servidor.

Esta etiqueta nos da un mayor control sobre las acciones a realizar antes de enviar el formulario.

```
<button type="button" id="btn">Enviar form</button>
```

12.7. Etiqueta <select>

La etiqueta **select** permite seleccionar a los usuarios una de las opciones que aparecen en un menú desplegable.

A este campo se le puede añadir el atributo **name**, cuyo valor será con el que se va a enviar el valor de la opción que se haya seleccionado.

También le podemos añadir el atributo **multiple** el cual permite la selección de múltiples opciones.

En este caso para seleccionar varias opciones, tenemos que tener pulsada la tecla de *ctrl* al mismo tiempo que pulsamos sobre las opciones.

12.8. Etiqueta `<option>`

La etiqueta **option** representa cada opción que se puede seleccionar dentro de un **select**. El texto que se va a mostrar en el desplegable es el texto que se escriba entre la etiqueta de apertura y de cierre.

El atributo **value** es el que indica el valor que se va a enviar al servidor, cuando se envíe el formulario.

También tiene un atributo **selected** que indica que la opción se encuentra marcada y será la que se muestre en el campo del desplegable. Si ninguna de las opciones tiene este atributo, se mostrará la primera opción que se haya definido.

```
<p>En que país vives?</p>
<select name="pais" id="pais">
  <option value="sp">España</option>
  <option value="it">Italia</option>
  <option value="fr">Francia</option>
  <option value="po">Portugal</option>
</select>
```

12.9. Etiqueta `<textarea>`

El elemento **textarea** crea un campo de texto con múltiples líneas. Este elemento tiene que tener una etiqueta de apertura y de cierre.

El contenido que se añada entre las etiquetas, aparecerá al crear el elemento, como un texto por defecto. Si el usuario no lo borra, se enviará junto al resto de información cuando se envíe el formulario. Algunos sitios tienen implementado un script de JavaScript que se encarga de eliminar este texto cuando se pulsa en el campo.

El tamaño del campo se debe de controlar con CSS. En las antiguas versiones de HTML se le podían añadir los atributos de **cols** y **rows** para establecer el tamaño del campo.

```
<textarea name="descripcion" id="descripcion" cols="30" rows="10">Una descripción...</textarea>
```

12.10. Atributo **placeholder**

En cualquier campo de texto se puede usar el atributo **placeholder** cuyo valor se mostrará en el campo de texto hasta que el usuario pulse sobre este.

Este atributo sirve para dar una idea al usuario de la información que se espera que se añada en el

campo.

```
<input type="text" name="usuario" placeholder="Robb">
<input type="password" name="password" placeholder="Stark">
```

12.11. Atributo **autofocus**

El atributo **autofocus** funciona con los campos en los que se pueden introducir valores. Y sirve para añadir el foco en un elemento automáticamente. De esta forma no tenemos que pulsar en el campo para escribir.

```
<label>Email:
  <input type="email" autofocus>
</label>
```

12.12. Atributo **pattern**

El atributo **pattern** nos permite añadir una expresión regular para validar el campo. Si el texto introducido no la cumple, entonces nos dará un error la validación.

```
<label>Color hexadecimal:
  <input type="text" pattern="^#([a-fA-F0-9]{3})|([a-fA-F0-9]{6}))$">
</label>
```

12.13. Atributo **required**

El atributo **required** indica que un campo es obligatorio, y no se puede dejar vacío. Si el se deja sin introducir ningún valor, entonces nos dará un error la validación.

```
<input type="text" required>
```

12.14. Ejercicio

- Crear un formulario para hacer un login
 - Tiene que tener la opción de 'permanecer registrado'
 - Añadir validaciones a los campos que lo requieran

12.15. Ejercicio

- Crea un formulario que simule la compra de un producto
 - Tiene que tener campos para todos los datos del usuario (nombre, apellidos, email, fecha de

nacimiento...)

- La dirección de envío
- Opciones de envío (envío rápido, envío lento, envío de 1 día...)
- Datos de pago con distintas opciones (paypal, tarjeta de crédito...)
- Un campo para indicar si se quiere donar una cantidad
- Añadir validaciones a los campos que lo requieran
- Obtener el foco en el primer campo de texto cuando se carga la página

12.16. Ejercicio

- Crear la estructura de controles que puede tener un vídeo o un archivo de música en la web
 - Añadir barra de tiempo
 - Añadir barra de volumen
 - Añadir botones de play, pausa, silenciar, pantalla completa...
 - Mostrar el tiempo actual y el tiempo que dura el vídeo o la canción

13. Imagenes

En una página web la mayor parte de las veces se van a mostrar imagenes. Podemos querer incluir un logo, una foto, un gráfico... Pero preparar estas imagenes para que se vean bien en una web y no tarden en cargar puede tomar su tiempo.

Las imagenes pueden decir más de lo que trata el contenido que tenemos que una descripción de este. En caso de no tener imagenes, siempre existen páginas donde se pueden comprar.

Hay que tener cuidado si cogemos imagenes de internet, porque pueden tener copyright y nos podemos meter en un lío.

Si vamos a tener imagenes en nuestra página web, siempre es buena idea tenerlas todas en una carpeta (normalmente se le pone de nombre **images**).

Cuando nuestra página va creciendo es buena idea separarlas en subcarpetas para tener estas imagenes más organizadas, por ejemplo, una carpeta para los productos, otra para las imagenes de la interfaz, otra para las novedades...

13.1. Etiqueta ``

La etiqueta **img** nos permite añadir una imagen en el documento. Esta etiqueta es un elemento vacío por lo que no necesita una etiqueta de cierre.

Este elemento debe de llevar el atributo **src** que es el que le indica a la etiqueta, donde se encuentra la imagen que se tiene que mostrar. Normalmente usaremos una *URL relativa*.

```

```

A las imágenes se les puede añadir con el atributo **alt** un texto alternativo que se mostrará cuando la imagen no sea posible mostrarla. Este texto debería de ser una descripción de la imagen, para que cualquier persona que usar un *screen reader* pueda entender que se está mostrando.

```

```

También se le puede añadir el atributo **title** para añadir una descripción de lo que se muestra en la imagen. Este texto se mostrará en un *tooltip* al pasar el ratón por encima de la imagen.

```

```

Esta etiqueta también se le pueden añadir los atributos **width** y **height** para especificar el ancho y el alto de la imagen en píxeles.

```

```

13.2. Etiqueta **<figure>**

Las imágenes muchas veces se muestran con una descripción, un encabezado, o una leyenda. La etiqueta **figure** permite asociar una imagen con su leyenda.

Se pueden tener más de una imagen dentro de un elemento **figure**, si todas ellas comparten la misma leyenda.

```
<figure>
  
  <p><i>Un minion contento porque va a estrenar la nueva película</i></p>
</figure>
```

13.3. Etiqueta **<figcaption>**

La etiqueta **figcaption** permite a los autores de las imágenes añadir una leyenda o encabezado a la imagen.

En las antiguas versiones de HTML no había ninguna forma de añadir una leyenda a las imágenes.


```
<figure>
  
  <p>
    <figcaption><i>Un minion contento porque va a estrenar la nueva película</i>
  </figcaption>
  </p>
</figure>
```

13.4. Transparencia en las imágenes

Si queremos añadir una imagen que sea parcialmente transparente para que se vea los elementos que hay por detrás de ella, entonces esta imagen tiene que estar en formato `.png`.

13.5. Ejercicio

- Crear una página con una receta
 - Se tiene que mostrar imágenes con su leyenda según se van explicando los pasos a seguir
 - Al final de la página se tienen que mostrar unas cuantas imágenes en línea de otras recetas

14. Elementos semánticos

Los **elementos semánticos** son aquellos elementos que proporcionan información extra, como por ejemplo donde hay que poner énfasis en una frase, cual es el significado de un acrónimo...

Los navegadores muestran el contenido de estas etiquetas de una forma distinta, pero no deberíamos usarlos para eso, sino que su propósito es describir el contenido de las páginas web de una forma más precisa.

El uso de estos elementos frente a otros que pueden parecer que hacen exactamente lo mismo es que estos elementos proporcionan información extra a otros programas como los **screen readers** o los motores de búsqueda.

Por ejemplo, usar una etiqueta semántica que añada énfasis en una frase, hará que el *screen reader* según vaya leyendo el contenido, entonces esta parte la lea con otro tono de voz.

14.1. Etiqueta ``

La etiqueta **strong** indica que el contenido que va dentro de ella es bastante importante.

El contenido de este texto aparece en negrita.

```
<p>La última temporada de Juego de Tronos ya tiene <strong>fecha de estreno</strong>
</p>
```

14.2. Etiqueta ``

El elemento **em** le proporciona énfasis al contenido de la etiqueta, que puede cambiar el significado de la frase sutilmente.

El contenido se muestra en cursiva.

```
<p>Hoy es un día <em>soleado</em></p>
```

14.3. Etiqueta `<blockquote>`

El elemento **blockquote** se usa para escribir citas largas que ocupan un párrafo.

El párrafo se añade dentro de las etiquetas *blockquote*, y muestra el contenido indentado.

Este elemento debería de llevar el atributo `cite` para indicar de donde ha salido esta cita o donde podemos encontrar más información sobre ella. Normalmente este atributo obtiene como valor una *url*.

```
<blockquote cite="https://youtu.be/QyU7glpHg-c">
  <p>¿Se preguntan el por qué del traje rojo? ¡Ah! Es para que los malos no sepan que
  estoy sangrando. Ese tío también lo ha pillado al ponerse pantalones marrones.</p>
</blockquote>
```

No se debería de usar este elemento para indentar el contenido de un documento. Para esto tendríamos que usar CSS.

14.4. Etiqueta `<q>`

El elemento **q** se usa para escribir una cita corta.

El contenido que está entre esas etiquetas se muestra entrecomillado.

Al igual que el elemento anterior, debería de llevar el atributo `cite`.

```
<p>La frase <q>Solo sé que no se nada</q> se le atribuye a Sócrates.</p>
```

14.5. Etiqueta `<abbr>`

El elemento **abbr** se usa para mostrar la abreviación de una palabra, o para mostrar un acrónimo.

Para mostrar el significado de las siglas o la palabra abreviada completa, se suele usar el atributo `title` junto a esta etiqueta.

```
<p><abbr title="Ruby on Rails">RoR</abbr> es un framework basado en el patrón <abbr title="Modelo Vista Controlador">MVC</abbr> y escrito en Ruby.</p>
```

En la versión anterior de HTML existía la etiqueta **acronym** que se usaba para los acrónimos, pero en esta última versión de HTML, se recomienda usar la etiqueta *abbr* para ambos casos.

14.6. Etiqueta `<cite>`

El elemento **cite** contiene la referencia a una fuente de la cual se ha sacado información.

Este elemento muestra su contenido en cursiva.

```
<p><cite>Moonlight</cite> fue la última película en ganar el Oscar.</p>
```

La gente suele usar este elemento para marcar el texto de las citas en lugar de el sitio de donde se ha obtenido la información, y esto está mal. Para esto ya existen las etiquetas vistas anteriormente, `q` y `blockquote`.

14.7. Etiqueta `<dfn>`

La etiqueta **dfn** se usa para mostrar el término que se está definiendo.

Cuando aparece por primera vez un término en un documento, este término se suele definir, pues esta etiqueta se usa para contener la palabra que se está definiendo.

```
<p>El <dfn>CSS</dfn> es un lenguaje que nos permite añadir estilos a nuestros documentos.</p>
```

14.8. Etiqueta `<address>`

La etiqueta **address** tiene un uso muy específico, que es contener la información de contacto de una persona.

Esta etiqueta muestra el contenido en cursiva, y suele contener la dirección de la persona, el email, el número de teléfono...

```
<address>
  <p>666777999</p>
  <p>sherlock_h@gmail.com</p>
  <p>221b Baker St, Marylebone, London</p>
</address>
```

14.9. Etiqueta `<ins>`

El elemento **ins** se usa para mostrar contenido que se ha insertado en el documento.

El contenido que va entre estas etiquetas aparece subrayado.

Por ejemplo, este elemento se puede usar para indicar las respuestas que se han escrito en un examen de rellenar los huecos.

```
<p>La fórmula química del agua es: <ins>H<sub>2</sub>O</ins></p>
```

14.10. Etiqueta ``

El elemento **del** se usa para mostrar contenido que se ha eliminado en el documento.

El contenido que va entre estas etiquetas aparece tachado.

Por ejemplo, este elemento se puede usar para indicar que una respuesta de un examen estaba mal.

```
<p>La fórmula química del agua es: <del>H<sub>4</sub></del><ins>H<sub>2</sub>O</ins></p>
```

14.11. Etiqueta `<s>`

La etiqueta **s** indica que el contenido que se encuentra entre ella ya no es relevante pero que debería de aparecer todavía.

El contenido que va entre esta etiqueta se mostrará tachado.

Esta etiqueta se puede usar por ejemplo para mostrar el precio anterior de un producto al que se le ha aplicado un descuento.

```
<p>iPhone X: <s>1300€</s> 899€</p>
```

14.12. Etiqueta `<meter>`

La etiqueta **meter** representa un valor fraccional, o un valor dentro de un rango conocido.

A esta etiqueta se le puede añadir el atributo **value** que representa el valor que se tiene que mostrar.

También le podemos añadir los atributos **min** y **max** para representar el valor mínimo y máximo que se pueden mostrar en este elemento.

```
<meter min="0" max="5" value="4"></meter>
```

14.13. Etiqueta `<progress>`

La etiqueta **progress** se usa para ver el progreso de una tarea, y se ve como cualquier barra de progreso.

Esta etiqueta puede tener el atributo **max** que indica el valor total de la barra de progreso.

Y también tiene el atributo **value** para indicar por donde se encuentra el porcentaje de progreso con respecto al máximo valor que puede valer.

```
<progress max="100" value="40"></progress>
```

14.14. Ejercicio

- Crea un página web que represente un examen del colegio
 - Cada pregunta tiene que estar en una página distinta
 - Cada página tiene que mostrar el porcentaje de preguntas que se lleva hecho hasta esa página
 - Tiene que tener huecos que están vacíos
 - Tiene que tener algunas preguntas en las que se muestra una respuesta que no sería válida
 - Tiene que mostrar palabras importantes para una pregunta
 - Tiene que mostrar alguna cita y referencia
 - La última página tiene que mostrar la nota del examen

15. Atributos globales

Hay una serie de atributos que son globales a todos los elementos, que son:

15.1. id

El atributo **id** nos va a permitir añadir identificadores únicos en todo el documento.

El objetivo de este atributo es poder identificar elementos de forma rápida, para añadirle estilos, o obtenerlo en un script para modificarlo, eliminarlo...

El valor de los *ids* no debería repetirse entre los elementos de una misma página.

```
<ul id="lista-compra">
  <li>Calamares</li>
  <li>Tomate</li>
  <li>Leche</li>
</ul>
```

15.2. class

El atributo **class** obtiene como valor una lista de clases separadas por espacios.

Estas clases permiten identificar distintos elementos que son muy parecidos entre ellos.

Y nos va a servir para darle el mismo estilo a todos aquellos elementos que tienen la misma clase, sin necesidad de repetir los estilos por cada uno de ellos.

Y lo mismo podremos hacer con JavaScript, obtener todos aquellos que representan lo mismo. De esta forma podremos modificarlos, eliminarlos, cambiarles los atributos...

```
<ul id="lista-compra">
  <li class="producto">Calamares</li>
  <li class="producto">Tomate</li>
  <li class="producto">Leche</li>
</ul>
```

15.3. title

El atributo **title** contiene un texto que se va a mostrar al pasar el ratón sobre el. Se va a mostrar en forma de *tooltip*.

Este atributo nos proporciona información adicional sobre un texto, una imagen...

```
<h1 title="Título importante">Un título</h1>
```

15.4. hidden

El atributo **hidden** permite ocultar un elemento de la vista. Normalmente se usa para mostrar elementos que aparecen cuando se realiza una acción determinada, por ejemplo, cuando enviamos un formulario con campos erróneos, podemos mostrar distintos mensajes de error.

```
<h2 hidden>Este subtítulo está oculto</h2>
```

15.5. data

El atributo **data** nos permite añadir atributos propios en las etiquetas. Para ello después del *data* se añade un *guión* seguido del nombre de nuestro atributo.

Esto se suele usar sobre todo para pasar información de las etiquetas HTML a los scripts de JavaScript, por ejemplo para poder guardar en los atributos la información que se está mostrando en los distintos elementos de un documento.

Usando este atributo, podemos almacenar todos los valores de un elemento en una sola etiqueta para poder recuperarlos de una forma más sencilla.

```
<li class="producto" data-producto="Leche" data-precio="2" data-cantidad="3">Leche:  
3*2€</li>
```

15.6. dir

El atributo **dir** indica la dirección en la que se quiere mostrar el texto del elemento.

Los posibles valores que se le pueden dar son:

- ltr (left to right): se utiliza para idiomas que se escriben de izquierda a derecha, como el español, el inglés...
- rtl (right to left): se utiliza para idiomas que se escriben de derecha a izquierda, como el árabe.
- auto: intenta detectar un carácter que indique la direccionalidad que tiene que tomar, y la aplica.

```
<p dir="rtl">Underweaaar po kass po kass para tú la bodaaa. Underweaaar jeje baboiii  
la bodaaa para tú la bodaaa ti aamoo! Pepete. Belloo! baboiii la bodaaa baboiii poulet  
tikka masala para tú hana dul sae tatata bala tu baboiii wiiiii. Poulet tikka masala  
bee do bee do bee do para tú wiiiii. Aaaaaah underweaaar poulet tikka masala potatoooo  
belloo!</p>
```

15.7. style

El atributo **style** nos permite añadir unos estilos en línea al elemento al que se le pone el atributo.

Este atributo se usaba en las anteriores versiones de HTML, sigue funcionando, pero deberíamos evitar usarlo.

```
<h1 style="color:red">Título principal</h1>
```

15.8. spellcheck

El atributo **spellcheck** nos permite mostrar aquellas palabras de un campo de texto que pueden no estar bien escritas.

Este atributo puede tener dos valores:

- **true**: subraya las palabras que pueden estar mal escritas
- **false**: no subraya las palabras que pueden estar mal escritas

Por defecto siempre coge el valor **true**, aunque no pongamos el atributo.

```
<textarea spellcheck>La leche se ha bertido</textarea>  
<input spellcheck="false" value="La leche se ha bertido">
```

Una vez que nos encontramos con las palabras que pueden estar mal escritas, podemos ver las posibles soluciones que nos da el navegador pulsando el botón derecho del ratón sobre la palabra.

16. Etiquetas **<meta>**

Las etiquetas **meta** se ponen dentro de la etiqueta **head** y contienen información sobre la página en la que se encuentran.

Estas etiquetas no son visibles a los usuarios sino que su propósito es dar información a los motores de búsqueda como quien ha creado la página, una descripción sobre ella, unas palabras clave...

Los atributos más comunes de estas etiquetas son **name** y **content** que se usan juntos en la misma etiqueta. El valor del atributo **name** es la propiedad a la que le vamos a dar un valor, y el valor del atributo **content** es el valor que se le da a la propiedad.

El valor del atributo **name** puede ser cualquiera que nos inventemos, aunque hay algunos que ya están definidos y que son usados por los motores de búsqueda como:

- **description**: contiene una descripción de la página web. Esta descripción la suelen usar los motores de búsqueda para entender el contenido de la página, y como mucho solo se pueden usar 155 caracteres.
- **keywords**: contiene una lista de palabras clave separadas por comas, y que son las que puede usar el usuario a la hora de buscar la página.
- **author**: indica quien es el autor de la página web.


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="description" content="Una descripción de esta página">
  <meta name="keywords" content="programación, html, css, html5, css3, informática">
  <meta name="author" content="Elliot Alderson">
  <title>Document</title>
</head>
<body>
</body>
</html>
```

También podemos indicar el juego de caracteres que vamos a usar en nuestra web con el atributo `charset` e indicandoselo.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta name="description" content="Una descripción de esta página">
  <meta name="keywords" content="programación, html, css, html5, css3, informática">
  <meta name="author" content="Elliot Alderson">
  <meta charset="UTF-8">
  <title>Document</title>
</head>
<body>
</body>
</html>
```

16.1. Ejercicio

- Añadir las etiquetas metas en los ejercicios de la receta y de las series

17. Comentarios

Si se quiere añadir un **comentario** en el código HTML para que no sea visible en el navegador, se pueden poner, escribiendo el comentario entre las etiquetas que aparecen a continuación:

```
<!-- Un comentario -->
```

Es buena idea añadir comentarios al código sin importar si lo has hecho tu, porque puede servirte de ayuda si alguna vez tienes que volver a tocar esa parte del código, o si le toca a algún compañero hacerlo. Los comentarios le ayudarán a entender mejor y más rápido porque algo se ha hecho de esa forma.

Aunque es bueno poner algunos comentarios en las páginas HTML, tampoco hay que pasarse con

ellos. Solo deberían de ponerse cuando sea necesario porque el código sea difícil de entender.

18. Escapar caracteres

Hay algunos caracteres que están reservados en HTML como pueden ser los símbolos de **mayor** y **menor**. En caso de que queramos que estos caracteres aparezcan en la página necesitamos **escaparlos** y lo vamos a hacer con los **códigos de escape**.

Los *códigos de escape* son unos códigos especiales que se pueden usar para mostrar símbolos como los que se han dicho antes, u otros como el símbolo de **copyright**, los símbolos de algunas monedas...

Algunos de los *códigos de escape* que se pueden usar son los siguientes:

Table 1. Códigos

Símbolo	Códigos
Espacio en blanco	nbsp;
Tabulación	emsp;
Menor que	lt; y #60;
Mayor que	gt; y amp;
Ampersand	amp; y #38;
Euro	euro; y #8364;
Libra	pound; y #163;
Yen	yen; y #165;
Copyright	copy; y #169;
Marca registrada	reg; y #174;
Símbolo de multiplicación	times; y #215;
Símbolo de división	divide; y #247;



Todos los códigos van precedidos de **&**.

18.1. Ejercicio

- Crea una página web que tenga una apariencia similar a la de la siguiente imagen

Por una Zaragoza accesible para todas las personas

ZaraGOZADA es un proyecto de Plena Inclusión Aragón para mejorar la accesibilidad cognitiva

EL PERIÓDICO
santiago@elporperiodico.com
ZARAGOZA



Plena Inclusión Aragón sigue realizando acciones para mejorar la accesibilidad cognitiva de la ciudad Zaragoza.

Mejorar la accesibilidad cognitiva consiste en hacer fácil de entender todo lo que nos rodea.

Hablamos de hacer más accesibles espacios donde es difícil orientarse. También hacer más comprensibles textos con palabras complicadas.

Una de las iniciativas de Plena Inclusión Aragón para mejorar la accesibilidad es un proyecto llamado ZaraGOZADA.

Este proyecto es posible gracias

a la colaboración de la Obra Social la Caixa.

Con este proyecto se van a conseguir distintos objetivos.

El primero de estos objetivos es sensibilizar a la sociedad de la importancia de la accesibilidad cognitiva para las personas con discapacidad intelectual.

En grupo

El segundo objetivo es crear un grupo de personas con discapacidad intelectual especializadas en accesibilidad cognitiva.

Este grupo redactará textos en un lenguaje más sencillo. También realizará estudios de accesibilidad cognitiva en distintos espacios de Zaragoza.

Plena Inclusión Aragón ha lanzado una convocatoria para Instituciones, Empresas y Entidades de Zaragoza que quieran hacer



Reunión del grupo denominado Alavoz de la Accesibilidad.

más accesibles sus espacios.

De entre todas las organizaciones inscritas se seleccionarán a tres para realizar un estudio de accesibilidad gratuito.

El estudio

Este estudio lo llevará a cabo el equipo de personas con discapacidad intelectual especializadas en accesibilidad cognitiva.

Gracias a este estudio, las organizaciones podrán conocer cómo hacer sus espacios o textos más accesibles.

Ser accesibles conlleva muchos beneficios para cualquier organización. Uno de los motivos es que

hay otros colectivos de personas como la tercera edad o los inmigrantes, que también necesitan esta accesibilidad.

El plazo para poder inscribirse a la misma finaliza el lunes, 27 de marzo, a las 14.00 horas.

Para inscribirse y encontrar toda la información de esta convocatoria hay que entrar a la página web www.plenainclusionaragon.com.

También se puede llamar al teléfono de Plena Inclusión Aragón, que es 976 738 581, o escribir a esta dirección de correo electrónico: santiagoaguado@plenainclusionaragon.com.