CONTENIDO:

TIPOS DE ETIQUETAS

Etiquetas con contenido.

Son etiquetas que tiene tres partes (por este orden)

1. Apertura de la etiqueta
2. Contenido de la etiqueta
3. Cierre de la etiqueta

Algo así:

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

HTML tiene un número limitado de etiquetas y no es necesario conocer todas. Con una lista más o menos pequeña podemos construir la gran mayoría de páginas web. En caso contrario, siempre podemos visitar las referencias.

Un ejemplo de etiqueta de HTML con contenido sería un párrafo:

```
<p>
    HOLA MUNDO
</p>
```

- `<p>` Sería la apertura
- **HOLA MUNDO** sería el contenido, en este caso sólo texto aunque podríamos meter muchas cosas "dentro"
- `</p>` Sería el cierre de la etiqueta

Etiquetas sin contenido

Son etiquetas que sólo tienen parte de apertura y carecen de contenido. Pueden estar cerradas o no, aunque yo recomiendo que se cierren. De esta manera:

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

Un ejemplo de etiqueta HTML sin contenido sería una imagen:
`<img ... />`

Atributos

Las etiquetas pueden tener atributos de los que nos interesa saber lo siguiente:

- Proporcionan información adicional sobre la etiqueta.
- Las etiquetas pueden tener o no tener atributos e incluso tener más de uno.
- Siempre se añaden en la etiqueta de apertura.
- Hay atributos generales (para todas) o específicos (para algunas).
- Se representan nombre_atributo="valor_atributo".

Un ejemplo:

```
<img src="foto.png"  />
```

Tipos de etiquetas

1. Etiquetas de cabecera
2. Etiquetas de texto
3. Etiquetas de listas imágenes, tablas, listas y enlaces
4. Etiquetas multimedia
5. Etiquetas semánticas
6. Etiquetas interactivas
7. Otras etiquetas

Comentarios

Además de todas estas etiquetas nuestra página web podrá llevar comentarios que son, normalmente, texto descriptivos que no se van a mostrar en nuestra web.

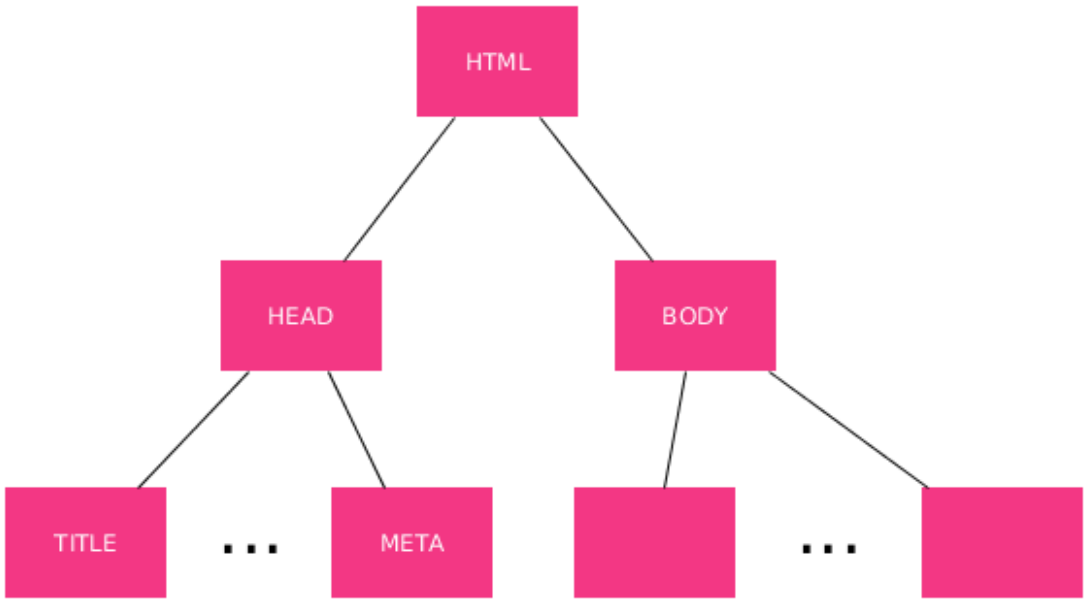
Los comentarios van encerrados en esta estructura `<!--  -->` y un ejemplo sería:

```
<!-- Esto es un comentario en HTML -->
```

Una vez ya tenemos claro qué es eso de *etiqueta* con lo que yo voy a llenar mi página web el siguiente paso es tener muy claro qué estructura tiene que tener mi web.

NOTA: Recordad que HTML define la estructura que tiene que tener una web.

De manera general podemos representar toda página Web como un árbol *genealógico* cuya estructura común, para todas las webs, podemos decir que será la siguiente:



Esto significa lo siguiente:

- Hay una etiqueta *padre* de todo, la etiqueta **html** y entre la apertura y el cierre de esta etiqueta meteremos el resto de nuestra página.
- La etiqueta **html** tiene dos hijos, la etiqueta **head** que es la cabecera, que ya veremos qué elementos contiene, y la etiqueta **body** que es la que en realidad contiene los elementos de mi web.
- A su vez esas dos etiquetas anteriores pueden tener sus propios hijos y así sucesivamente.

ETIQUETAS DE CABECERA

La cabecera de mi página web es lo que está dentro de la etiqueta y en relación a ella es importante saber lo siguiente:

- Todo lo que va dentro de esa etiqueta no representa contenido alguno, es decir, nada de lo que yo pongo se va a ver en nuestro navegador.
- Contiene otras etiquetas que nos van a ayudar a dar una descripción de mi página.
- Puede contener enlaces a hojas de estilos y scripts.

De las etiquetas que puede contener la cabecera destacaremos las siguientes:

- **title:** que nos sirve para indicar el título de la página que es lo que se muestra en la parte de arriba del navegador.
- **style:** que es una etiqueta que me permite incluir estilos y que veremos con más detenimiento al final de este mismo curso.
- **meta:** que es una etiqueta que puede aparecer varias veces y con distintos atributos y que nos va a servir para dar una descripción de la página de diversas formas y maneras.
- **link:** para enlazar con archivos externos, normalmente hojas de estilos.
- **base:** para indicar la ruta base para construir el resto de las rutas a archivos en mi página web.
- **script:** para incluir normalmente ficheros Javascript que doten de vida o animación a mi web. Para utilizar Javascript, lo que hacemos es indicar al HTML que queremos cargar un script (generalmente, un archivo de texto con código Javascript) y hacerlo funcionar sobre la página actual. Se utiliza la etiqueta <script></script>, que permite indicar una serie de atributos. El código JS se incluye en el HTML dentro de la etiqueta o bien se incluye en el fichero Javascript enlazado en el atributo **src**.

```
<script src="/carpeta/archivo.js"></script>
```

Etiquetas `<meta>`

Es una etiqueta muy importante ya que, etiquetas meta o meta tags, encabezan un documento HTML y suministran información a navegadores y motores de búsqueda sobre una página. Los metadatos son invisibles para el usuario y se encargan de añadir información para facilitar el análisis de los archivos HTML y la gestión del contenido de una página web.

- `<meta charset="utf-8">` Indica formato de codificación de caracteres utilizada en nuestra página. Es importante para no tener problemas con caracteres como vocales acentuadas, ñ, etc. o con otros alfabetos.

- `<meta name="description" content="......">` En la descripción podemos describir brevemente el contenido de la página web. Esta información se muestra como una síntesis en dos líneas del tema de una página que aparece bajo la URL en los buscadores como Google o Bing, por lo que se recomienda cuidar su redacción. Ese texto descriptivo irá en el atributo *`content`*.

- `<meta name="keywords" content="...">` Con esta etiqueta meta los administradores tienen la posibilidad de definir palabras clave para el buscador. Las keywords son aquellos criterios a los que responde un buscador para ofrecerle al usuario páginas HTML como respuesta, donde tales palabras clave son parte de los meta tags.
- Esta etiqueta era considerada como **el factor SEO** más importante, ya que los buscadores primitivos recurrían a este atributo como característica central para el posicionamiento en las listas de resultados. Pero hoy es ignorado por Google y su importancia ha sido minimizada por Bing y otros buscadores debido a la facilidad con que puede ser manipulado. Aunque esto no significa que un mal uso de este meta tag no pueda llevar a la devaluación en el posicionamiento.

  Por ejemplo: *`content="programacion,html,meta"`*

- `<meta name="author" content="Autor de la página">`
- `<meta name="copyright" content="Propietario del copyright" />`

  Estos dos meta tags, de uso opcional desde el punto legal, permiten hacer referencia al diseñador de una página web y al propietario de los derechos del código fuente de una página HTML

- `<meta name="viewport" content="width=device-width, initial-scale=1.0">` que es nuevo en HTML5 y que me asegura que la página se escalará para adaptarse a la pantalla del dispositivo. Esto, no obstante, no asegura que se vea bien, para ello debemos utilizar CSS.
- `<base href= "http://miDominio.com/" target="_blank">` El elemento HTML <base> especifica la dirección URL base que se utilizará para todas las direcciones URL relativas contenidas dentro de un documento. El atributo _target indica el destino por defecto de los hipervínculos. Sólo puede haber un elemento <base> en un documento HTML.

Etiqueta `<meta equiv-http>`

Los meta tags con el atributo `"http-equiv"` contienen **información equivalente a las de los encabezado HTTP** que envía el servidor al navegador del cliente junto con el archivo HTML.

Los encabezados HTTP con el mismo nombre tienen prioridad frente a los meta tags de nuestro documento HTML, pero será utilizada, por ejemplo, si en el servidor no ha configurado ese encabezado.

Las más importantes:

Control de caché

Para asegurar la fluidez del funcionamiento de Internet, las webs se suelen guardar en servidores proxy intermedios o en el caché del navegador para ser descargadas fácilmente en el futuro. Para impedirlo usamos el meta tag "cache-control" con el valor no-cache:

```
<meta http-equiv="cache-control" content="no-cache"/>
```

Expires

Es posible que no se quiera prohibir el almacenamiento temporal en caché de forma absoluta. En este caso se puede usar el atributo "http-equiv" para definir una fecha de expiración para los datos HTML solicitados.

```
<meta http-equiv="expires" content="tiempo de expiración en segundos">
```

Así es posible asignar a las páginas HTML la fecha de caducidad que se deseen. Una vez se supera esta fecha, el navegador tiene que cargar el documento HTML desde la página de origen.

Refresh

Se define una fecha de vencimiento, a partir de la cual los navegadores o los motores de búsqueda serán dirigidos a una URL establecida previamente.

```
<meta http-equiv="refresh" content="10; url=http://www.nueva.com"/>
```

Si no se indica la **url** funcionará igual que expires, pasado el tiempo establecido se cargará la misma página desde el origen.

La etiqueta <link >

La etiqueta HTML <link> cuenta con un mecanismo de añadir al documento actual información externa, es decir, el elemento <link> permite a los autores de páginas web vincular su documento a otros recursos. Es un elemento vacío, no tiene etiqueta de cierre

Este elemento <link> se usa con mayor frecuencia para enlazar hojas de estilos externo y esta información se añade en la cabecera del documento, dentro del elemento HEAD.

Atributos

href

Este atributo especifica la URL del recurso enlazado. Y la URL del documento vinculado puede ser absoluta o relativa.

rel

El atributo rel es necesario, porque especifica la relación entre el documento actual y el documento vinculado.

Este atributo puede contener, las siguientes palabras claves permitidas:

author: Da un enlace al autor del documento o artículo actual.

help: Proporciona un enlace a la ayuda contextual.

icon: Importa un icono para representar el documento actual.

license: Indica que el contenido principal del documento actual está cubierto por la licencia de copyright descrita por el documento de referencia.

next: Indica que el documento actual es parte de una serie y que el siguiente documento de la serie es el documento de referencia.

prev: Indica que el documento actual es parte de una serie y que el documento anterior de la serie es el documento de referencia.

stylesheet: Importa una hoja de estilo para el documento actual.

media

se utiliza para indicar los tipos de medio para los que el recurso fue diseñado, por ejemplo:

screen (valor por defecto), para presentación en pantallas de computadoras no-paginadas;

print, para salida a una impresora;

projection, para presentaciones en proyectores;

braille, para presentación en dispositivos braille;

tv, para televisores;

type

indica el tipo de contenido que contiene el recurso enlazado. Aquí se debe utilizar un tipo MIME

Etiqueta link para enlazar CSS

HTML define la estructura de una página web y CSS define la presentación de dicha página.
Las siglas CSS (Cascading Style Sheets) significan «Hojas de estilo en cascada» y parten de un concepto simple pero muy potente: aplicar estilos (colores, formas, márgenes, tipo de letra, tamaña, etc.) a uno o varios documentos HTML de forma automática y masiva.
La idea de CSS es la establecer una separación entre la presentación y el contenido. Ya que a medida que nuestra aplicación o sitio web crezca, esto hará que el código sea más fácil de modificar y mantener.

La idea es la siguiente:

- Los documentos HTML (contenido) incluirán sólo información y datos, todo lo relativo a la información a transmitir.
- Los documentos CSS (presentación) incluirán sólo los aspectos relacionados con el estilo (diseño, colores, formas, etc.).

Para que los estilos que se implementan con CSS se reflejen en el HTML, tenemos que encontrar una manera de **enlazar el CSS con el HTML**.
Puedes hacer el enlace escribiendo CSS:

- **CSS en línea**: atributo style contiene información de CSS que se aplica al elemento que contiene el atributo.
- **CSS interno**: etiqueta o elemento <style> contiene información de CSS que se aplica a todo el documento. Este elemento HTML <style></style> solo puede ser insertado en el encabezado del documento <head>.
- **CSS externo**: etiqueta <link>

Enlazar un archivo externo de CSS

```
<link rel="stylesheet" type="text/css" href="carpeta/styles.css">
```

Para enlazar CSS contenido en archivo externo con un documento HTML, debemos usar la etiqueta <link ….> que debe ir en el <head> del documento HTML.

El atributo rel: es la relación entre el archivo externo y el archivo actual. Para el CSS, se utiliza stylesheet. Por ejemplo, rel="stylesheet".

El atributo type: es el tipo de documento que está enlazando al HTML. Para CSS, es text/css. Por ejemplo, type="text/css".

El atributo href: se utiliza para especificar la ubicación del archivo CSS y el nombre del archivo. Es un enlace en el que se puede hacer clic, por lo que también puede mantener pulsada la tecla CTRL y hacer clic en él para ver el archivo CSS.

El atributo media: se utiliza para especificar el tipo de dispositivo para el que fue escrito el archivo.

```
<link rel="stylesheet" type="text/css"   media="screen" href="pantalla.css">
<link rel="stylesheet" type="text/css"   media="print" href="impresora.css">
```

Etiqueta link para insertar icono en la pestaña

Podemos usar la etiqueta <link> y el atributo rel="icon" para agregar un favicon a la pestaña del navegador en HTML.

El favicon es el icono o símbolo que acompañará a tu URL en los navegadores. Esta pequeña imagen tiene como función principal ayudarte a identificar más fácilmente una determinada URL.

Debemos establecer la ubicación de la imagen que se utilizará como favicon en el atributo **href**.

El tamaño más común para crear un favicon es 16x16 píxeles. Sin embargo, también pueden aparecer en dimensiones un poco más grandes 32x32 px.

Los formatos admitidos de imagen son: .ico, .png, .jpg, .gif y .svg. Aunque se puede realizar el favicon con cualquiera de ellos, el formato óptimo es el .**png**, ya que no tiene pérdida de calidad y permite transparencias. Otros formatos como .svg (vectorial), también están bien, pero por el momento no lo admiten todos los navegadores.

El .ico, por su parte, era uno de los formatos más utilizados por ser el que requerían los navegadores en un inicio, pero en la actualidad empieza a decaer su uso.

```
<link rel="icon" type="image/png" href="ruta/icono_16.png" sizes="16x16">

<link rel="icon" type="image/png" href="ruta/icono_32.png" sizes="32x32">
```

https://www.convertico.com/

ETIQUETAS DE TEXTO

Párrafo

La etiqueta está pensada para definir párrafos de texto e imágenes.

```
<p>Página a la que llega el formulario</p>
<p>Quien tiene un amigo tiene un tesoro.</p>
<p>Amigo en la adversidad, es amigo de verdad.</p>
<p>Cuando marzo mayea, mayo marcea.</p>
<p>La primavera la sangre altera.</p>
```

Quien tiene un amigo tiene un tesoro.

Amigo en la adversidad, es amigo de verdad.

Cuando marzo mayea, mayo marcea.

La primavera la sangre altera.

Separador temático: <hr>

La etiqueta vacía <hr> está pensada para separar párrafos e indica un cambio de tema. En versiones previas de HTML representaba una línea horizontal. Aún puede ser representada como una línea horizontal en los navegadores visuales, pero ahora es definida en términos semánticos y no tanto en términos representativos, por tanto, para dibujar una línea horizontal se debería usar el CSS apropiado.

```
<p>Quien tiene un amigo tiene un tesoro.</p>
<p>Amigo en la adversidad, es amigo de verdad.</p>
<hr/>
<p>Cuando marzo mayea, mayo marcea.</p>
<p>La primavera la sangre altera.</p>
```

Quien tiene un amigo tiene un tesoro.

Amigo en la adversidad, es amigo de verdad.

Cuando marzo mayea, mayo marcea.

La primavera la sangre altera.

Encabezados <h1>…<h6>

Son etiquetas que nos van a permitir añadir "títulos" o encabezados a distintas secciones de nuestra página. Estas etiquetas tienen el siguiente formato:

```
<hx>
 ...
 Contenido o texto
 ...
</hx>
```

x deberá ser sustituido por un número del 1 al 6, desde 1 que es el mayor tamaño hasta 6 que es el más pequeño. El texto se mostrará en negrita.

Un ejemplo simple:

```
<h1>Hola</h1>
<h2>Hola</h2>
<h3>Hola</h3>
<h4>Hola</h4>
<h5>Hola</h5>
<h6>Hola</h6>
```

En un documento HTML solo debe haber una etiqueta <h1>

En HTML5, el elemento <h1> se utiliza para representar el encabezado principal o el título de una página web. Según las especificaciones y las prácticas recomendadas de HTML, se aconseja tener un solo <h1> por documento HTML. Razones:

- **Jerarquía y estructura semántica**: Los encabezados (<h1>, <h2>, <h3>, etc.) se utilizan para estructurar y organizar el contenido de la página en una jerarquía. El <h1> se considera el **título principal** y define la jerarquía más alta en la estructura de la página. Tener múltiples <h1> puede confundir a los navegadores y a los motores de búsqueda sobre cuál es el título principal de la página, lo que afecta la comprensión y la indexación adecuada del contenido.

- **Accesibilidad**: Para las personas que utilizan lectores de pantalla u otras tecnologías de asistencia, tener múltiples títulos principales puede dificultar la navegación y comprensión del contenido de la página. Un único <h1> claro y descriptivo proporciona una estructura más clara y coherente para estos usuarios.

- **SEO y motores de búsqueda**: Los motores de búsqueda utilizan la estructura de encabezados para comprender la jerarquía y el contexto del contenido de la página. Usar varios <h1> puede confundir a los motores de búsqueda sobre el tema principal de la página, lo que puede afectar la relevancia y el posicionamiento en los resultados de búsqueda.

Si se necesita dividir el contenido en secciones o subsecciones, se deben utilizar los encabezados en orden secuencial (por ejemplo, <h2>, <h3>, <h4>, etc.) para mantener una estructura lógica y coherente, comenzando con un solo <h1> que representa el título principal de la página.

Etiquetas de formato

Hay multitud, todas le dan cierto estilo al texto que contienen y destacaremos las siguientes:

- **...** para poner un texto en negrita.
- **...** para poner un texto en cursiva.
- **~~...~~** para mostrar un texto tachado.
- **...** para poner un texto con énfasis (similar a cursiva).
- **...** para dar más énfasis que en el caso de
- **^{...}** para poner un texto como superíndice de otro texto.
- **_{...}** para poner un texto como subíndice de otro texto.
- **<mark>...</mark>** para poner un texto resaltado o marcado (nuevo en html5).
- **<q>...</q>** para mostrar una pequeña cita.
- **<cite>...</cite>** para mostrar el título de una referencia bibliográfica.
- **<time datatime="YYYY-MM-DD HH:SS" >...</time>** para mostrar horas.
- **<address>...</address>** para mostrar direcciones.
- **<blockquote>...</blockquote>** para poner citas largas.
- **<small> </small>** hace el tamaño del texto una talla más pequeña (por ejemplo, de largo a mediano, o de pequeño a extrapequeño) que el tamaño mínimo de fuente del navegador. En HTML5, este elemento es reutilizado para representar comentarios laterales y letra pequeña, incluyendo derechos de autor y texto legal, independientemente de su estilo de presentación.
- **<u>...</u>** para subrayado de palabras o texto.
- **<big></big>** está obsoleto en HTML5

```
<p>Ahora puedes hacer tus compras gratis
  <small>
    (Promoción válida hasta el <time  datetime="2022-12-31"> (31/12/2022) </time>
solo para clients post-pago)
  </small>
</p>
<footer>
   <p> <small> ©2022 Todos los derechos reservados </small> </p>
</footer>
```

Ahora puedes hacer tus compras gratis (Promoción válida hasta el (31/12/2022) solo para clients post-pago)

©2022 Todos los derechos reservados

`<time datatime="YYYY-MM-DD">`

Representa un periodo específico en el tiempo.

El **datetime** atributo de este elemento se usa para traducir la hora a un formato legible por máquina para que los navegadores puedan agregar recordatorios de fechas a través del calendario del usuario y los motores de búsqueda puedan producir resultados de búsqueda más inteligentes.

Tengo un examen de lenguaje de marcas <time datetime="2022-10-14 20:00"> el jueves 14 de octubre a las 8 de la tarde</time>

El formato de datetime debe ser YYYY-MM-DD para la fecha y una hora en formato de 24 horas, HH:MM

Preformateado: `<pre></pre>` y `<code></code>`

La etiqueta <pre> se utiliza cuando se quiere conservar los espacios en blanco, los saltos de línea y los tabuladores del texto original. En el resto de las etiquetas, los navegadores no muestran ni las líneas en blanco ni varios espacios en blanco seguidos.

Un bloque preformateado <pre> se muestra con la familia de fuente **`monospace`** (normalmente, el tipo de letra es Courier).

La etiqueta <code> hace lo mismo que la etiqueta <pre> pero **añade un matiz semántico**. Indica a los navegadores que el fragmento de texto preformateado es código de un programa.

<pre> también se puede utilizar para mantener el formato de un poema, de una tabla tabulada… En ese caso <code> no sería adecuado.

Las tipografías monoespaciadas son aquellas cuyos caracteres **ocupan siempre el mismo espacio horizontal**, en contraposición a las tipografías de anchura variable o proporcional, donde los caracteres poseen tamaños diferentes entre sí.

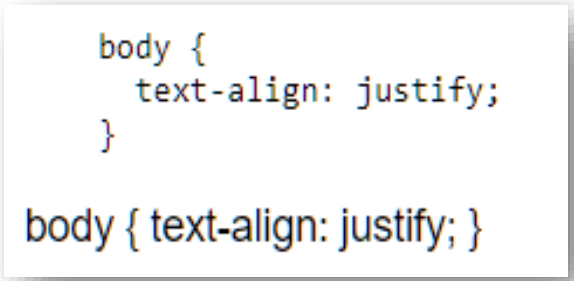
Las palabras que tengan el mismo número de letras ocuparán el mismo espacio horizontal.

El ejemplo siguiente muestra cómo se puede imitar una tabla usando la etiqueta <pre>. Como en esa etiqueta se respetan los espacios en blanco, se puede conseguir que los datos se muestren en columnas alineadas. Sin embargo, si se utilizan etiquetas (una o varios), el navegador no respeta los espacios en blanco o los saltos de línea y se pierde la alineación pretendida.

```
<pre>
        País      Continente       Población (2013)
        España    Europa             47 millones
        India     Asia               1250 millones
</pre>

    <p>País Continente Población (2013)
        España Europa 47 millones
        India Asia 1250 millones</p>

    <p>País Continente Población (2013)</p>
    <p>España Europa 47 millones</p>
    <p>India Asia 1250 millones</p>
```

El ejemplo siguiente es similar al anterior. La etiqueta **<pre>** respeta el sangrado, mientras que la etiqueta **<p>** no lo hace.

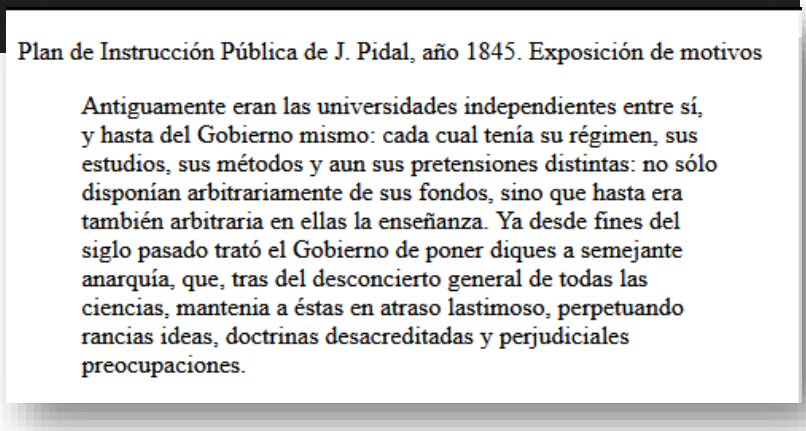
```
<pre>
    body {
      text-align: justify;
    }
</pre>

<p>
    body {
      text-align: justify;
    }
</p>
```



Bloque de cita: <blockquote>

La etiqueta <blockquote> está pensada para identificar una cita larga, que puede contener varios párrafos u otras etiquetas. Los navegadores suelen mostrar la etiqueta <blockquote> **con márgenes a izquierda y derecha**.

```
<p>Plan de Instrucción Pública de J. Pidal, año 1845. Exposición de motivos</p>
<blockquote>
   <p>Antiguamente eran las universidades independientes entre sí, y hasta del
Gobierno mismo: cada cual tenía su régimen, sus estudios, sus métodos y aun sus
pretensiones distintas: no sólo disponían arbitrariamente de sus enseñanza. Ya
desde fines del siglo pasado trató el Gobierno de poner diques a semejante
anarquía, que, tras del desconcierto general de todas las ciencias, mantenía a
éstas en atraso lastimoso, perpetuando rancias ideas, doctrinas desacreditadas y
perjudiciales preocupaciones.
   </p>
</blockquote>
```



Plan de Instrucción Pública de J. Pidal, año 1845. Exposición de motivos

Antiguamente eran las universidades independientes entre sí, y hasta del Gobierno mismo: cada cual tenía su régimen, sus estudios, sus métodos y aun sus pretensiones distintas: no sólo disponían arbitrariamente de sus fondos, sino que hasta era también arbitraria en ellas la enseñanza. Ya desde fines del siglo pasado trató el Gobierno de poner diques a semejante anarquía, que, tras del desconcierto general de todas las ciencias, mantenia a éstas en atraso lastimoso, perpetuando rancias ideas, doctrinas desacreditadas y perjudiciales preocupaciones.

Etiqueta `<output></output>`

Representa el resultado de un cálculo. Etiqueta semántica, sin ningún cambio en el diseño.

```
<p>Resultado de una operación:</p>
<output>50</output>
```

Etiqueta `<br>` y `<wbr>`

El elemento HTML line break <br> produce un salto de línea en el texto. Es útil para escribir un poema o una dirección, donde la división de las líneas es significante. Es una etiqueta vacía, no tiene cierre.

```
    <p>
       IES Venancio Blanco<br>
       Calle Filipinas, 33<br>
       CP 37003 Salamanca<br>
    </p>
```

El elemento HTML word break opportunity **<wbr>** representa una posición dentro del texto donde el **navegador puede opcionalmente saltar una línea,** aunque sus reglas de salto de línea no crearían un salto en esa posición.

Cuando una palabra es demasiado larga y se teme que el navegador va a romper sus líneas en el lugar equivocado, puede utilizar el <wbr> elemento para agregar oportunidades de ruptura de palabras.

Ejemplo, si tienes que mostrar una **url** muy larga, se recomienda romper la línea antes de un signo de puntuación para evitar errores ya que el lector podría confundirla con el final de la información

http://this.is.a.really.long.example.com/With/deeper/level/pages/deeper/level/pages/deeper/level/pages/deeper/level/pages/deeper/level/pages

```
 <p>http://this<wbr>.is<wbr>.a<wbr>.really<wbr>.long<wbr>.e
xample<wbr>.com/With<wbr>/deeper<wbr>/level<wbr>/pages<wbr>
/deeper<wbr>/level<wbr>/pages<wbr>/deeper<wbr>/level<wbr>/p
ages<wbr>/deeper<wbr>/level<wbr>/pages<wbr>/deeper<wbr>/lev
el<wbr>/pages
    </p>
```

Etiqueta `<address></address>`

El elemento HTML <address> indica que las líneas siguientes proporcionan información de contacto de una persona u organización.

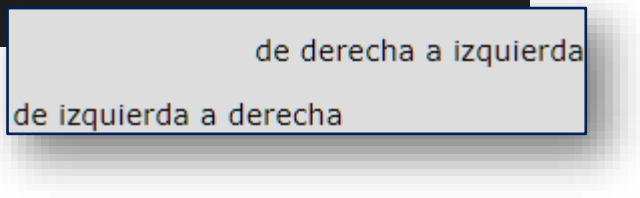
```
<p>Contacta con el autor de esta página:</p>
<address>
     <a href="mailto:jim@rock.com">jim@rock.com</a><br>
     <a href="tel:+34123456789">(+34) 123-456789</a>
</address>
```

Atributo `dir`

Podemos especificar la dirección del texto en el HTML utilizando el atributo puede tomar uno de estos valores:

- **ltr** acrónimo de "left to right", de izquierda a derecha
- **rtl** acrónimo de "right to left", de derecha a izquierda

```
<div dir="rtl">
      <p>de derecha a izquierda</p>
 </div>
 <div dir="ltr">
       <p>de izquierda a derecha</p>
 </div>
```



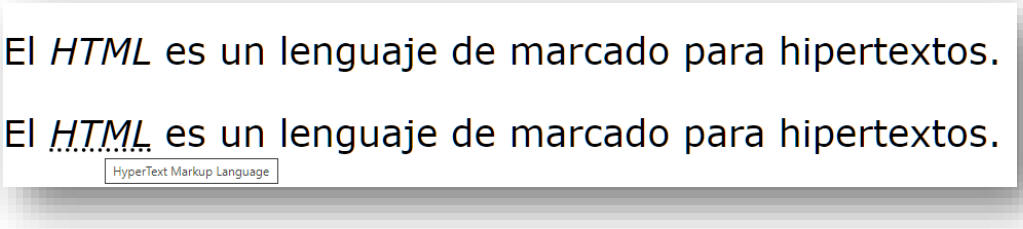
Etiqueta <abbr> </abbr> <dfn> </dfn>

El elemento HTML <abbr> representa una abreviación o acrónimo; Los navegadores habitualmente muestran el contenido del atributo "title" en un recuadro al colocar el ratón encima del contenido del elemento. NO debes olvidar que el contenido del atributo "title" provee la expansión/explicación de la abreviación.

El elemento HTML <dfn> sirve para marcar un término que se quiere va a definir.

Estas dos etiquetas pueden utilizarse juntas, **dfn** para indicar que se va a definir un término y si este es un acrónimo con **abbr** explicamos la abreviatura.

```
<p> El <dfn>HTML</dfn> es un lenguaje de marcado para hipertextos.</p>
<p> El <dfn><abbr title="HyperText Markup Language">HTML</abbr></dfn>
es un lenguaje de marcado para hipertextos.</p>
```



Etiqueta <ruby> junto con <rt> <rp>

Las nuevas etiquetas ruby, rt y rp de HTML5, su funcionalidad es la enmarcar el contenido con formato ruby.

Este contenido es utilizado en los idiomas de Asia del Este, como por ejemplo el japonés o el chino que usan 'caracteres/dibujos' para realizar las frases.

Se utiliza para proporcionar una anotación de texto en un idioma que incluye caracteres pequeños (generalmente caracteres fonéticos o caracteres ideográficos) **sobre o al lado de un texto base.**

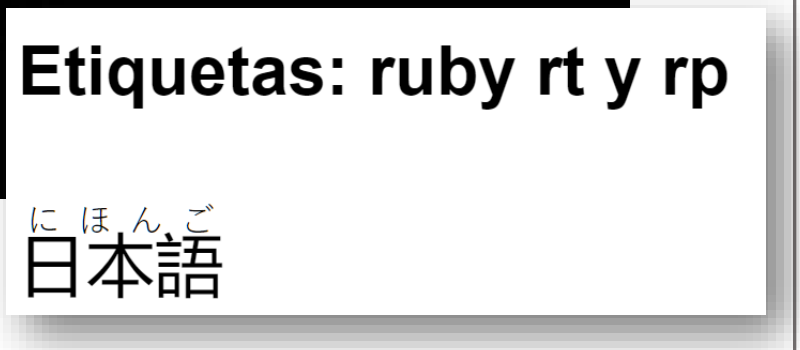
Esto es útil en situaciones en las que se necesita proporcionar una pronunciación o explicación para el texto base en un idioma diferente o para mostrar caracteres pequeños en un idioma que no se pueda representar fácilmente en el sistema de escritura principal.

La estructura básica de la etiqueta **<ruby>** incluye tres elementos principales:

- **<rb>** (Ruby Base) para el texto base
- **<rt>** (Ruby Text) para el texto pequeño que se muestra **sobre o al lado** del texto base.
- **<rp>** (Ruby Parentheses) o (Ruby Punctuation) se utiliza para incluir paréntesis u otros caracteres de puntuación alrededor del texto Ruby, se usa para indicar la pronunciación o proporcionar información adicional sobre el texto base.

- Cuando un navegador no es compatible con Ruby Text, mostrará el contenido de <rp> para proporcionar un contexto adicional al usuario.

```html
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Ruby Text</title>
</head>
<body>
    <h4>Etiquetas: ruby rt y rp</h4>
    <ruby>
        <rb>日本語</rb>
        <rt>にほんご</rt>
    </ruby>
</body>
</html>
```



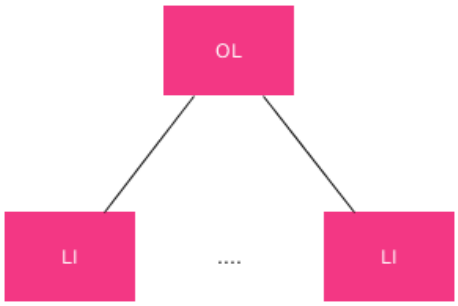
LISTAS

Las listas son una de las construcciones más usadas a la hora de elaborar textos, no sólo en HTML. Son elementos de bloque. Pueden ser de 3 tipos:

- **Listas numeradas:** que son aquellas que expresan un orden entre los diferentes elementos de la lista. Este orden podrá ser numérico, alfabético etc.
- **Listas no numeradas:** que simplemente muestra los elementos de la lista uno tras otro.
- **Listas de definición:** que muestran diversos términos junto con su definición.

A continuación, vamos a ver el árbol DOM para cada una de estas variedades.

Lista numeradas u ordenadas



Un ejemplo:

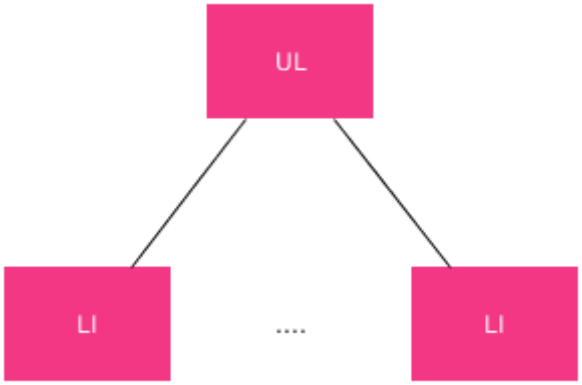
```
    <!-- Listas numeradas -->
       <h3>Lista Numerada - Ordered List (ol)</h3>
       <ol>
           <li>Primer elemento</li>
           <li>Segundo elemento</li>
           <li>Tercer Elemento</li>
           <li>Cuarto elemento</li>
       </ol>
```

ol sería la etiqueta padre y cada uno de los elementos de la lista iría en una etiqueta **li**.

Etiqueta	** **
Descripción	Lista ordenada o numerada
Atributos	reversed: Dar la vuelta al orden. start: Inicio. type: Tipo de numerador (a, A, i, I y 1).
Tipo	Bloque

Los atributos de ol están obsoletos, deben modificarse con CSS

Listas no numeradas o desordenadas



Un ejemplo:

```
<!-- Listas no numeradas-->
<h3>Lista NO Numerada - Unordered List (ul)</h3>
    <ul>
        <li>Primer elemento</li>
        <li>Segundo elemento</li>
        <li>Tercer elemento</li>
        <li>Cuarto elemento</li>
     </ul>
```

ul sería la etiqueta padre y cada uno de los elementos de la lista iría en una etiqueta **li**.

Etiqueta	
Descripción	Lista desordenada o no numerada
Atributos	type: circle, square, disc
Tipo	Bloque

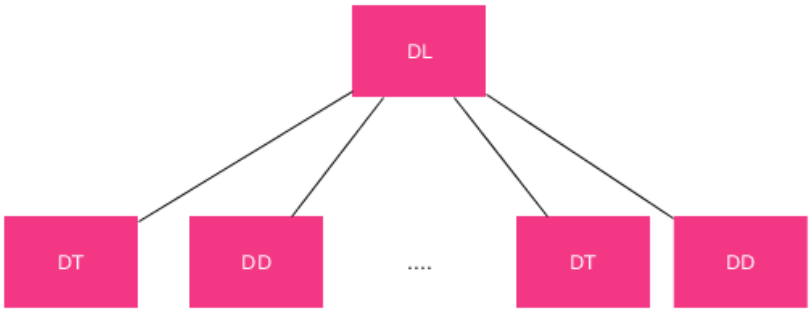
En versiones anteriores de HTML se usaba el atributo "type", en HTML5 no recomendado, está obsoleto (deprecated).

La sintaxis recomendada para indicar la apariencia se basa en usar CSS como indicamos a continuación:

```
<ul style="list-style-type:disc">
<ul style="list-style-type:circle">
<ul style="list-style-type:square">
```

La apariencia final que realmente consigamos **dependerá del navegador**, es decir, no todos los navegadores se comportan de la misma forma a la hora de mostrar un símbolo gráfico como una viñeta o marca.

Listas de definición



Un ejemplo:

```
<!-- Listas de definición-->
     <h3> Lista de descripción - Description List (dl)</h3>
        <dl>
            <dt>Primer término</dt>
            <dd>Definición del primer término</dd>
            <dt>Segundo término</dt>
            <dd>Definición del segundo término</dd>
            <dt>Tercer término</dt>
            <dd>Definición del tercer término</dd>
            <dt>Cuarto término</dt>
            <dd>Definición del cuarto término</dd>
        </dl>
```

dl es la etiqueta padre y cada término se define mostrando consecutivamente las etiquetas **dt**, que se corresponde con el término que vamos a definir, y **dd** que es la definición del término anterior.

IMÁGENES

La inclusión de imágenes simples se viene haciendo desde las primeras versiones de HTML con la etiqueta sin contenido **`<img ... />`**

Es importante destacar que la imagen es **un elemento en línea** que se coloca, si cabe, inmediatamente después de los elementos previamente añadidos.

Los atributos más comunes que le podemos poner a esta etiqueta son:

- **src:** que indica la ruta en la que se encuentra el archivo de la imagen.
- **alt:** un texto alternativo para describir la imagen en caso de que no se cargue o para dispositivos especiales para usuarios con discapacidad visual (por ejemplo)
- **width:** para especificar la anchura de la imagen (px, %). Si no se escogerá la anchura propia de la misma.

- **height:** para especificar la altura de la imagen (px, %). Si no se escogerá la altura propia de la misma.

Modificar las dimensiones de la imagen

HTML incorpora dos atributos la etiqueta img: width (ancho) y height (alto); con ellos podemos indicar explícitamente la anchura y altura que tendrá una imagen.

Normalmente estos valores se emplean para establecer los valores reales de la imagen, lo que facilita el proceso de carga de la página, pero también se pueden emplear para que una imagen se muestre más pequeña, más grande o simplemente distorsionada en alguno de sus ejes.

Atributo	Significado
`alt`	**Obligatorio.** Indica un texto alternativo. Ese texto aparece cuando la imagen no se ha podido cargar (o durante la carga). También suele aparecer cuando arrimamos el cursor a la imagen a fin de informarnos sobre ella. Es un texto tenido en cuenta por los buscadores a fin de identificar lo que muestra la imagen. Deberíamos tomarnos este atributo como obligatorio
`width`	Anchura de la imagen. No es aconsejable su uso para modificar el tamaño de la imagen, ya que si la ampliamos no se verá en buena calidad y si la reducimos estaremos cargando una imagen grande para luego mostrarla en pequeño; sería más inteligente reducirla primero con un editor de imágenes. En cualquier caso, es importante utilizar este atributo (junto con height) para que el navegador sepa de antemano el tamaño de la imagen y así que prepare la página correctamente. De este modo si la imagen no se carga, al menos veremos el rectángulo que la misma ocuparía y la página no se desbarata.
`height`	Altura de la imagen. Tiene las mismas connotaciones que el atributo anterior.

Si modificas **solo uno** de los valores de una imagen (ya sea el ancho o la altura) y dejas el otro valor sin especificar, el navegador web intentará mantener las proporciones originales de la imagen. Los navegadores respetan la relación de aspecto de la imagen para evitar distorsiones visuales. Ambos atributos se acompañan de un valor entre comillas y puede venir expresado en dos medidas diferentes:

- **Píxeles**: si el valor indicado no lleva ningún tipo o las letras px, estamos indicando que la medida es en píxeles.

Por ejemplo o

- **Porcentaje**: el tamaño puede estar expresado en relación al espacio de visualización o a su contenedor, como un porcentaje. Lo indicamos con el signo %.

Por ejemplo, en

En HTML5, cuando especificas la altura (height) y el ancho (width) de una imagen utilizando porcentajes, estos porcentajes **se calculan en relación al tamaño de su contenedor o espacio de visualización**, no en relación al tamaño de la imagen original. Esto significa que el porcentaje se basa en el espacio en el que se encuentra la imagen dentro de la página web, **no en el tamaño original de la imagen**.

Atributo `draggable` para las img y otros elementos HTML (como listas, hipervínculos, párrafos). Este atributo se puede utilizar para indicar **si un elemento puede ser arrastrado y soltado.** Sin embargo, el comportamiento exacto puede variar dependiendo del navegador y del tipo de elemento. En el caso de las imágenes, la mayoría de los navegadores permiten arrastrarlas por defecto, incluso sin el atributo draggable.

Sin JavaScript su funcionalidad será limitada, se puede hacer que un elemento sea arrastrable, pero no es posible controlar dónde se puede soltar.

```
<img src="mafalda.png" draggable="true" width="300" alt="Una imagen arrastrable">
```

Rutas

El concepto de ruta es un concepto muy importante ya que se utiliza es muchos temas relacionados con la informática y en concreto, en la creación de páginas WEB, se utiliza para referenciar archivos, recursos y/o partes de alguna web. De manera general podemos distinguir:

- **Relativas:** Toman como base el directorio en el que se encuentra nuestro fichero. Son las recomendadas.

```
<img alt="Logo de Empresa" src="img/logo.jpg">
```

- **Absolutas:** Toman como base el directorio raíz de mi equipo (/ en Linux y c:\ en Windows). Cuidado, sólo funcionarán en tu mismo equipo.

```
<img alt="Logo de empresa" src=" C:\Users\USER\img\logo.jpg">
```

- **Url:** La dirección de Internet de un recurso (fichero, imagen etc..). Puede desaparecer y entonces dejará de mostrarse en nuestra web.

```
<img alt="Logo de Empresa" src="https://www.linuxadictos.com/wp-content/uploads/logo.jpg">
```

Algunos ejemplos de utilización de estos atributos podrían ser:

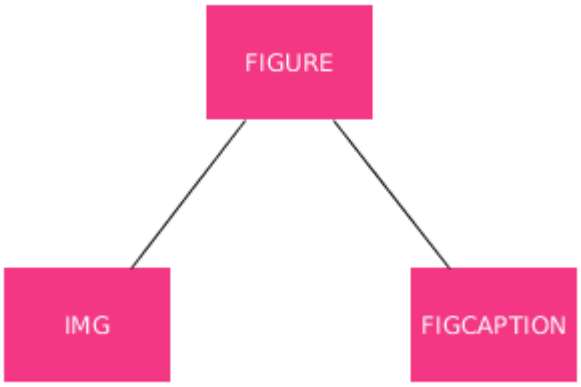
```
<!-- Si no indico nada se respetan las dimensiones originales de la imagen-->
    <!-- Ruta relativa -->
        <img alt="Logo de empresa" src="img/logo.jpg
    <!-- Ruta absoluta -->
        <img alt="Logo de empresa" src="/home/public_html/examples/img/logo.jpg">
    <!-- Ruta URL -->
       <img alt="Logo de empresa" src="https://www.linuxadictos.com/uploads/logo.jpg">

<!-- Si modifico una dimensión se respetan las proporciones-->
        <img height="50px" alt="Logo de empresa" src="img/logo.jpg">
        <img width="50px" alt="Logo de empresa" src="img/logo.jpg">

<!-- Si pongo ambas dimensiones puedo modificar las proporciones-->
        <img height="300px" width="100px" alt="Logo de Empresa" src="img/logo.jpg">
```

Etiqueta <figure></figure>

Una novedad en HTML5 es la construcción de etiquetas alrededor de la imagen que nos va a permitir mostrar una imagen con un texto asociado.

Son elementos de bloque.

En este caso el árbol DOM será el siguiente:



Donde:

- **`<figure>`** es la etiqueta padre.
- **`<img .../>`** es una etiqueta de imagen que hemos visto anteriormente.
- **`<figcaption>`** es una etiqueta que contendrá el texto asociado a la imagen.

Un ejemplo:

```
<figure>
    <img alt="Logo de Empresa" src="img/logo.jpg">
    <figcaption>Logotipo de La empresa con la etiqueta figure </figcaption>
</figure>
```

Píxel:

Los pixeles son la unidad más pequeña de información que compone una imagen, siendo en general de forma cuadrada.

El término se acuñó para definir al más pequeño de los elementos que conforman una imagen: El **Picture Element**, de lo cual partió el vocablo, formado por el "**Pix**" de Picture y el "**el**" de Element.

Si la cantidad de pixeles que conforman una imagen son pocos, y el tamaño en centímetros de esta es grande, la definición de la imagen expuesta será pobre, mientras que, a mayor número de ellos, la imagen se visualizará mejor, permitiéndole contar con un grado de detalle superior, tanto en los contornos como en el cuerpo de la imagen.

La cantidad de pixeles de una imagen determina el tamaño en bytes que la misma tendrá, es decir, que mientras más píxeles posea una imagen, mayor espacio ocupará en el disco duro de nuestra computadora y, por lo tanto, más difícil de manipular será.

La profundidad de color o bits por píxel es la cantidad de colores que puede mostrar una imagen.

El número de bits que se utilizan para representar cada pixel determina precisamente cuántos colores o gamas de gris pueden ser visualizados.

Si utilizamos un solo bit para representar cada píxel de color, tenemos 2 valores por píxel, encendido y apagado, es decir, blanco y negro:

$$1 \text{ bit por píxel: } 2^1 = 2 \text{ colores (blanco y negro)}$$

Cuando utilizamos 8 bits por píxel podemos representar hasta 256 colores por cada píxel. Los formatos que usan esta profundidad de bits son GIF y PNG8. Se utiliza una paleta de 256 colores creada a partir de los colores originales de la imagen.

$$8 \text{ bits por píxel: } 2^8 = 256 \text{ colores}$$

Con una profundidad de color de 24 bits podemos representar más de 16 millones de colores. Los formatos que usan esta profundidad de bits son JPG y PNG24.

Se habla de color verdadero porque es, aproximadamente, el número de colores que el ojo humano puede distinguir:

24 bits por píxel: 2^{24}= 16.777.216 color (true color)

Cuando se utilizan 32 bits para representar un color se agrega, para cada tono primario, un cuarto canal denominado alfa, que representa la transparencia. Este valor se utiliza cuando, por ejemplo, superponemos imágenes:

32 bits: agrega un **cuarto canal (alfa)** que representa la transparencia de cada tono.

Imágenes de mapa de bits:

A la imagen de mapa de bits se le conoce también como imagen rasterizada o bitmap, y está compuesta por una cuadrícula de píxeles, organizados en una rejilla.

Cada uno de los píxeles que conforma el mapa de bits tiene un color definido que presenta un valor.

Definición de imagen vectorial:

Las imágenes vectoriales se basan en una serie de coordenadas matemáticas que definen su posición, forma, color y otros atributos.

Estas imágenes se componen de vectores, que son unas figuras geométricas que pueden ser puntos, líneas, polígonos o segmentos. Por ejemplo, un rectángulo está definido por dos puntos, el círculo por un centro y un radio, mientras que una curva por varios puntos y una ecuación.

Eso sí, una imagen vectorial permite representar únicamente formas simples, lo que significa que no todas las imágenes se pueden describir todas las imágenes con vectores.

A las imágenes vectoriales al estar compuestas por entidades matemáticas, se le pueden aplicar transformaciones geométricas a la misma, como ampliar, expandir o reducir**, sin que pierdan nada de calidad**, ya que continuaremos viendo las diferentes líneas y manchas de colores perfectamente definidas.

Además, las vectoriales permiten definir una imagen con muy poca información, lo que hace **que los archivos tengan un tamaño bastante reducido**.

DIFERENCIA ENTRE MAPA DE BITS Y VECTORES

Las imágenes de mapa de bits cuentan con una retícula perfectamente definida y, por tanto, su calidad se mantiene fija. Si se amplía o disminuye la imagen, vemos como los píxeles que la forman se multiplican o dividen, lo que provoca una pérdida de calidad.

Por otro lado, las imágenes vectoriales al estar basadas en fórmulas matemáticas tienen una resolución infinita, y por tanto se pueden ampliar o reducir sin riesgo de que su calidad baje.

Aunque hay casos en los que resulta realmente complicado distinguir entre imágenes vectoriales y mapa de bits, si hacemos zoom sobre ellas nos resultará mucho más sencillo, ya que si nos encontramos con una retícula de píxeles se tratará de mapa de bits, mientras que si la definición no varía será vectorial.

PRINCIPALES USOS DEL MAPA DE BITS E IMAGEN VECTORIAL

Los formatos de la imagen vectorial se utilizan especialmente para logotipos, iconos, ilustraciones, infografías (diagramas visuales), tipografías (diseño de letras) u otros elementos para webs, que necesitan mantener su calidad al máximo en todo momento.

En cambio, el bitmap se utiliza más para fotografías o imágenes que tienen una buena resolución, pero no resulta tan importante mantener la calidad al hacer zoom.

VECTOR Y MAPA DE BITS: VENTAJAS Y DESVENTAJAS

VENTAJAS DE LA IMAGEN DE MAPA DE BITS

- Tienen una gran capacidad para representar la realidad a la perfección.
- Se muestran como la mejor opción para imágenes como fotografías, ya que destacan por el alto nivel de detalle que pueden alcanzar.

DESVENTAJAS DE LA IMAGEN DE MAPA DE BITS

- Al hacer zoom sobre ellas o reducirlas es habitual que pierdan mucha calidad y acaben *pixelándose*.
- Es posible que no tengan la calidad suficiente como para poder imprimirlas.
- Un mapa de bits BMP o en otro formato de mucha calidad, puede ocupar un espacio excesivo.

VENTAJAS DE LA IMAGEN VECTORIAL

- Se pueden escalar sin riesgo a que pierdan nada de calidad.
- Son la mejor opción para imprimir, ya que no pierden nada de calidad.
- Al estar basadas en fórmulas matemáticas en vez de en píxeles, son capaces de almacenar la información más compleja sin ocupar demasiado espacio.

DESVENTAJAS DE LA IMAGEN VECTORIAL

- La principal desventaja de los diferentes formatos de imagen vectorial es que tienen muchas limitaciones para la creación de imágenes reales.
- Resulta muy complicado reproducir una fotografía a partir de vectores, aunque no es imposible.
- Existen imágenes vectorizadas que representan fotografías con una excelente calidad, aunque por lo general son archivos muy complejos y pesados.

Formato de imágenes

Las imágenes son fundamentales para que una página web sea más atractiva. Son el primer contenido no textual que se planteó poder incorporar al estándar HTML. Prácticamente no hay páginas sin imágenes.

Los navegadores tienen capacidad de mostrar imágenes, pero sólo las que pertenezcan a tipos concretos. Los tipos de imágenes reconocidos por la mayoría de los navegadores son:

- Formato bmp (Bitmap): El formato BMP es uno de los formatos más simples y antiguos para mapas de bits. Almacena imágenes sin comprimir píxel por píxel, lo que garantiza una calidad de imagen excepcional, pero da como resultado archivos más grandes en comparación con otros formatos.

- Formato jpg. Son imágenes que ocupan muy poco gracias a su alta compresión. No admiten animaciones ni zonas marcadas con transparencia. Son el formato habitual de la imagen digital en una página web.
- Formato gif. Imágenes con hasta 256 colores, por lo que son buenas para dibujos y logotipos, pero no para fotografías. Pueden incluso contener animaciones (uno de sus usos habituales) y colores marcados como transparentes, a través de los cuales se mostraría lo que esté por debajo de la imagen.
- Formato png. Imágenes fotográficas comprimidas al estilo de jpg pero con más calidad (ocupan más, normalmente) Permite la posibilidad además de utilizar canales alfa, es decir marcar opacidades con hasta 255 niveles. De esta forma habrá partes de la imagen que se mezclarán con el fondo. También hay posibilidad de hacer animaciones estilo gif.
- Formato webp. Formato de imagen auspiciado por Google para conseguir mejorar a PNG y JPG haciendo que ocupen menos sin pérdida de calidad respecto a estos formatos.
- Formato SVG. Es un formato vectorial de imágenes. Las imágenes vectoriales tienen la ventaja de que nunca pierden calidad independientemente de cuanto las ampliemos o las reduzcamos. La desventaja es que no se pueden utilizar para imágenes fotográficas.

Descartando a las imágenes SVG, el tamaño en disco de las imágenes puede ser mayor o menor dependiendo de su tamaño y su compresión. De modo que un tamaño grande implica más tardanza al cargar la página, pero una mayor nitidez en la imagen.

Las imágenes son fundamentales en las páginas web, su elección resulta vital para la estética de la misma, nunca se considera un mero acompañamiento ya que el impacto visual y las sensaciones sobre la profesionalidad de la página siempre le consiguen las imágenes con ayuda de la tipografía (pero ahí ya entra el lenguaje CSS) y la disposición o maquetación (para lo que también necesitamos CSS).

Para minimizar el efecto de la incompatibilidad de algunos formatos de imagen se dispone del elemento picture que se explica más adelante.

Picture <picture></picture>

Esta nueva etiqueta de HTML5 supone una mejora respecto a la etiqueta <img>, sobre todo si queremos diseñar páginas web que se adapten a todo tipo de pantallas.

La idea fundamental de <picture> es mostrar diferentes imágenes según ciertas condiciones. Para cada imagen que queramos mostrar utilizaremos una etiqueta **source**, es obligatorio poner al menos una. También es obligatorio poner, al final, una etiqueta **img**.

Para cada etiqueta source se pueden especificar unas condiciones, si se cumplen se mostrará la imagen indicada en la propiedad **srcset**, y ya no se evaluarán el resto de las etiquetas source; si no se cumplen las condiciones se pasará a comprobar la siguiente etiqueta source.

Si no se cumplen las condiciones para ninguna etiqueta srcset, se ejecutará la etiqueta img.

Si hay un error en la carga de la imagen, los navegadores a menudo no mostrarán el texto alternativo especificado en las etiquetas <srcset>. En su lugar, mostrarán el texto alternativo de la etiqueta <img> que actúa como un "fallback" (respaldo o alternativa) en caso de que ninguna de las fuentes de imagen se cargue correctamente.

El atributo width en la etiqueta source no admite porcentajes. En lugar de eso, se utiliza con un número entero que representa el ancho de la imagen en píxeles.

Se puede establecer medidas en % pero desde CSS

```
 <source srcset="img/bicicletaVectorial.svg" media="(min-width: 768px)
and (max-width: 1023px)" style="width: 100%;">
```

Las condiciones de las etiquetas source pueden ser de diversos tipos, todos los que permiten las **Media Query (MQ)**, vamos a explicar algunos de los que consideramos más útiles como la anchura de la ventana del navegador en la que se muestran las imágenes.

```
<picture>
    <source media="(min-width: 800px)" srcset="imagenes/en_rio_grande_800.jpg">
    <source media="(min-width: 600px)" srcset="imagenes/en_rio_mediana_600.jpg">
    <img src="imagenes/enOtroCaso.jpg" alt="Si no se cumple ninguna condición">
</picture>
```

Tipos de características

Tipo de característica	Valores	¿Cuándo se aplica?
`width`	tamaño	Si el dispositivo tiene el tamaño indicado exactamente.
`min-width`	tamaño	Si el dispositivo tiene un tamaño de ancho mayor al indicado.
`max-width`	tamaño	Si el dispositivo tiene un tamaño de ancho menor al indicado.
`aspect-ratio`	*aspect-ratio*	Si el dispositivo encaja con la proporción de aspecto indicada.
`orientation`	landscape | portrait	Si el dispositivo está en colocado en modo vertical o apaisado.

Aspect-ratio: Se calcula dividiendo el largo entre la altura de la imagen visible en pantalla, y se expresa normalmente como «**X:Y**». Ejemplo 4:3, 16:9

La etiqueta `<link rel="icon"  >`

Los tamaños y formatos más comunes para íconos (favicons) varían según los dispositivos y las resoluciones de pantalla. Aquí tienes una lista de algunos de los tamaños y formatos más comunes:

Tamaños comunes:

- **16x16 píxeles**: Para iconos en pestañas de navegadores.
- **32x32 píxeles**: También para iconos en pestañas y en algunos navegadores más antiguos.
- **48x48 píxeles**: Para dispositivos con pantallas de alta densidad y algunos navegadores.
- **64x64 píxeles**: A menudo utilizado en dispositivos móviles y tabletas.
- **128x128 píxeles**: Para pantallas de alta resolución y pantallas de inicio en dispositivos móviles.

Formatos comunes:

1. ****.ico**: Este es el formato más común para íconos en Windows y se usa ampliamente en navegadores.

2. ****.png**: A menudo se utiliza para íconos en navegadores web y es compatible con la mayoría de los navegadores.

3. **.svg: Scalable Vector Graphics (SVG) es un formato que es escalable y se adapta a diferentes resoluciones. Se utiliza en navegadores modernos y es especialmente útil para íconos simples.

Es importante proporcionar iconos en varios tamaños y formatos para garantizar una representación adecuada en una variedad de dispositivos y navegadores. Puedes incluir múltiples etiquetas `<link>` con diferentes tamaños y formatos en la sección `<head>` de tu documento HTML para cubrir una amplia gama de situaciones. Por ejemplo:

```
<link rel="icon" href="favicon.ico" sizes="16x16" type="image/x-icon">

<link rel="icon" href="favicon-32x32.png" sizes="32x32" type="image/png">

<link rel="icon" href="favicon-128x128.png" sizes="128x128" type="image/png">
```

Recuerda que la compatibilidad puede variar entre navegadores, por lo que es una buena práctica proporcionar una variedad de tamaños y formatos para asegurarse de que el favicon se vea bien en todas las situaciones.

ENLACES

Los enlaces, que se representan mediante el uso de la etiqueta **<a>** es una de las construcciones más importantes en HTML. Esta etiqueta puede tener varios atributos, de lo cuáles los más importante son:

- **href:** que es la dirección de Internet de destino (ya sea otra página web, una imagen, un fichero o lo que sea).
- **target:** que indica dónde voy a abrir ese enlace. Si no pongo nada se abrirá en la misma pestaña y si le doy el valor **target="_blank"** se abrirá en una nueva pestaña del navegador.

```
<p><a href="http://www.empresa.net">
     Enlace con url y que se abrirá en la misma pestaña </a></p>

<p><a href="/mistrabajos/segunda.html" target="_blank">Enlace que se
abrirá en una pestaña nueva y con ruta relativa</a></p>

<!-- Haciendo que una imagen sea enlace. Anidando etiquetas -->
<a href="http://www.empresa.net">
    <img width="100px" alt="Logo de empresa" src="img/logo.jpg">
</a>
```

Enlaces dentro de la misma página. Anclas

Puedo enlazar enlaces dentro de la misma página con construcciones como la siguiente:

```
    ...
    <a href="#contacto">Contacto</a>
    ...
    <section id="contacto">
        ...
    </section>
    ...
```

Enlaces para descargar un archivo

```
<a href="/tresImagenes/fox_terrier.png" download>
       Haz clic para descargar
</a>
<a href="/tresImagenes/fox_terrier.png" download="perro.png">
       Haz clic para descargar
</a>
```

Enlaces a un correo y teléfono

Al pinchar en el enlace **se abre el programa de correo que tengas instalado**, por ejemplo, el Outlook y se envía un correo a la dirección indicada a continuación de **mailto.**

La única diferencia con un enlace normal es al utilizar el texto **tel**: seguido del número de teléfono fijo o móvil al que queremos que el usuario pueda llamar, los navegadores y dispositivos móviles detectan que el enlace no es una URL si no que es un número de teléfono. Y sugieren al usuario la aplicación a utilizar para realizar la llamada.

```
<a href="img/fox_terrier.png" download="perro.png">Haz clic para
descargar</a>

<a href="#arriba">Moverse dentro de una página(ancla)</a>

<a href="otraPagina.html#posicion">Ir a otra página y en una posición
dentro de esa página(ancla)</a>
<a href="https://www.iesvenancioblanco.es/" target="_blank">Enlace a
URL y en otra pestaña</a>

<p><a href="mailto:pepito@gmail.com">Enviar correo</a></p>
<p><a href="tel:+34123456789">(+34) 123 456789</a></p>
```

```
<h2>Enlaces a archivos</h2>
<p><a href="img/fox_terrier.png" download="perro.png">Descargar imagen</a></p>
<!-- Este enlace permite visualizar un archivo PDF -->
<p><a href="documento.pdf">Abrir documento PDF</a></p>
<!-- Este enlace permite reproducir a un archivo de audio. -->
<p><a href="audio.mp3">Reproducir audio</a></p>
<!-- Este enlace permite descargar a un archivo de vídeo -->
<p><a href="video.mp4" download>Descargar video</a></p>
<h2>Anclas. Enlaces a secciones de una página web</h2>
<!--Este enlace ancla permite moverse dentro de la misma página. -->
<p><a href="#arriba">Moverse en la misma página elemento con id=arriba</a></p>
<!--Este enlace para colocarse en un posición exacta de otra página-->
<p><a href="otraPagina.html#posicion">Colocarse otra página y en una posición
dentro de esa página(ancla)</a></p>
<h2>Enlaces a URL externas en distintas pestanas</h2>
<p><a href="https://www.iesvenancioblanco.es/" target="_blank">
   Enlace URL externa y en otra pestaña</a></p>
<h2>Enlaces para enviar correo o llamar por teléfono</h2>
<!-- este enlace mostrará el teclado numérico en los dispositivos móviles -->
<p><a href="tel:+34123456789">Llamar al teléfono (+34) 123 456789</a></p>
<!-- Este enlace abrirá el cliente de correo del S.O del usuario -->
<p><a href="mailto:pepito@gmail.com">Enviar correo</a></p>
<h2>Enlaces para descarga software</h2>
<!-- Enlace para descarga de software -->
<p><a href="https://www.mozilla.org/es-ES/firefox/download/thanks/"
 download="Firefox" target="_blank">Descargar Firefox</a></p>
<!-- Enlace para descargar visual studio code -->
<p><a href="https://code.visualstudio.com/docs/?dv=win64user" download
 target="_blank">Descargar Visual Studio Code</a></p>
<h2>Enlaces para ver vídeo en stream</h2>
<!-- Enlace para ver vídeo en stream -->
<p><a href="https://www.youtube.com/watch?v=QH2-TGUlwu4" target="_blank">Ver video</a></p>
<h2>Enlaces a redes sociales</h2>
<!-- Enlace a Instagram -->
<p><a href="https://www.instagram.com/" target="_blank">Instagram</a></p>
<!-- Enlace a TikTok -->
<p><a href="https://www.tiktok.com/" target="_blank">TikTok</a></p>
<h2>Enlaces con imágenes</h2>
<p><a href="https://www.iesvenancioblanco.es/" target="_blank">
   <img src="img/logo.jpg" alt="VenancioBlanco"></a></p>
```

TABLAS

Otras de las construcciones básicas en HTML son las tablas.

Además de para la presentación de elementos se usaron históricamente para dar estructura a las páginas. No obstante, ya no se maqueta con tablas, CSS Grid es un sistema de maquetación web que divide la página en una cuadrícula o rejilla (grid) a partir de la cual se pueden posicionar los diferentes elementos de manera más sencilla, versátil y coherente

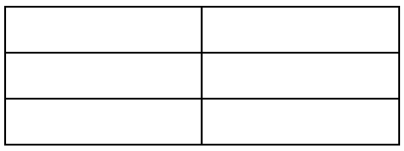
A pesar de eso siguen siendo un elemento importante y a continuación vamos a presentar varias formas de hacer tablas.

Tablas Simples

```
<table> Para crear la tabla </table>
<tr> Para crear cada fila </tr>
<td> Para cada celda de datos </td>
```

- Dentro de cada `<tr>` (fila) tantas etiquetas `<td>` como celdas queramos que tengan nuestras filas.
- Podemos utilizar `<td>` o `<th>` para crear cada celda, la única diferencia es que `<th>` aparece centrado y en negrita, se suele utilizar para indicar las celdas de la cabecera de la tabla.

Todas las etiquetas tienen apertura y cierre



EJEMPLO TABLA BÁSICA

Tabla de 3 filas y 2 columnas

```
<table>
  <tr>
    <td>.....</td>
    <td>.....</td>
  </tr>
  <tr>
    <td>.....</td>
    <td>.....</td>
  </tr>
  <tr>
    <td>.....</td>
    <td>.....</td>
  </tr>

</table>
```

Al definir una tabla, se debe pensar en primer lugar en las filas que la forman y a continuación en las columnas.

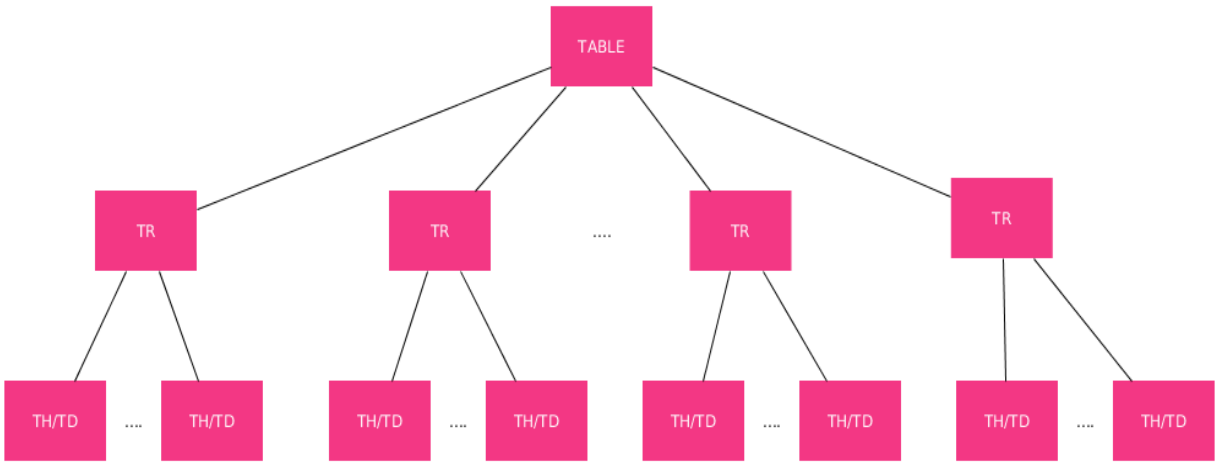
El motivo es que HTML procesa primero las filas y por eso las etiquetas **<tr>** aparecen antes que las etiquetas **<td>**

La estructura del árbol DOM más simple para una tabla sería similar a la siguiente:

- Una etiqueta **`<table>`** que contiene toda la tabla.
- Como hijos directos, tantas etiquetas como filas queramos que tenga nuestra tabla.
- Dentro de cada fila, tantas celdas como queramos que tengan nuestras filas. La única diferencia entre estas dos es que el contenido en la segunda se presenta centrado y el texto en negrita.

NOTAS:

- Si no coinciden el número de celdas en todas las filas veremos que suceden cosas "extrañas".
- La anchura de las celdas de una misma columna será la anchura de la más ancha de la columna.
- La altura de las celdas de una misma fija será la altura de la más alta de la fila.



```html
<table>
        <tr>
            <th>Nombre</th>
            <th>Apellidos</th>
            <th>Dirección</th>
        </tr>
        <tr>
            <td>Ana</td>
            <td>Pérez</td>
            <td>Calle Sol</td>
        </tr>
        <tr>
            <td>Manuel</td>
            <td>Sánchez</td>
            <td>Calle Ibiza</td>
        </tr>
        <tr>
            <td>María</td>
            <td>Fernández</td>
            <td>Calle Luna</td>
        </tr>
        <tr>
            <td>María</td>
            <td>Martín</td>
            <td>Calle Marte</td>
        </tr>

    </table>
```

Nombre	Apellidos	Dirección
Ana	Pérez	Calle Sol
Manuel	Sánchez	Calle Ibiza
María	Fernández	Calle Luna
María	Martín	Calle Marte

Tablas Completas

Existe además una forma más precisa y completa de construir tablas. Una forma que puede contener (aunque no es obligatorio) otras etiquetas dentro de la etiqueta raíz **<table>**. Estas nuevas etiquetas pueden ser:

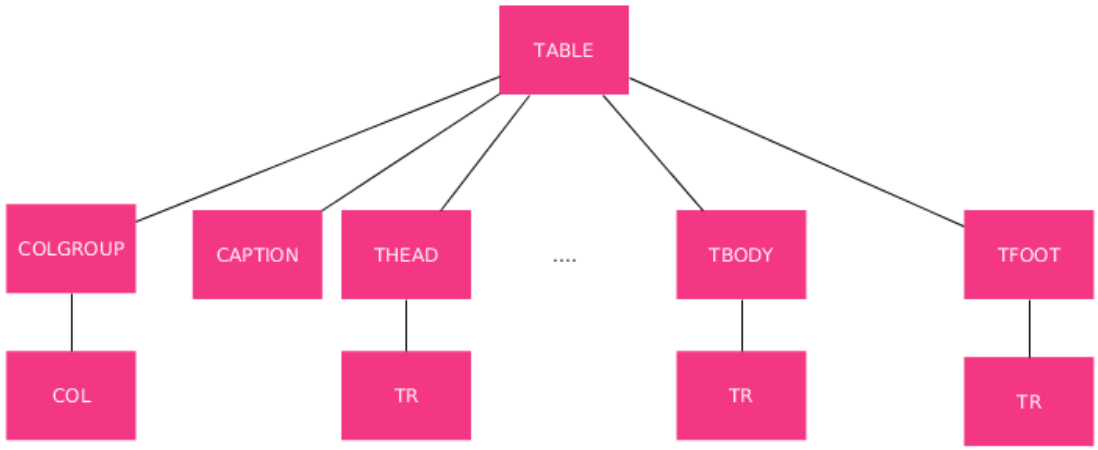
- `<colgroup>` y `<col>` nos permite agrupar las columnas para darles estilos. Cada uno de esos grupos lo definiremos usando una etiqueta con un **atributo span** para definir el número de columnas de cada grupo.

  - ✓ El elemento <colgroup> debe usarse como contenedor para las especificaciones de la columna. A diferencia de otras versiones, en HTML5 el elemento <col> no puede ser declarado hijo de table. Debe ser contenido por un elemento <colgroup> el cual representa a un grupo de columnas.
  - ✓ Cada grupo se especifica con un elemento <col>. (Sin etiqueta de cierre)
  - ✓ El atributo span especifica cuántas columnas consecutivas serán afectadas por el elemento <col>.
  - ✓ El conteo de columnas va de izquierda a derecha, de modo que para tener un elemento <col> que represente a la tercera columna de una tabla, debes primero representar con otro elemento <col> a las dos primeras.
  - ✓ El atributo style especifica el estilo para dar a las columnas.

- `<caption>` para añadir un título o leyenda a la tabla en la parte superior.

Las siguientes etiquetas sirven para agrupar las filas de una tabla en cabecera, pie y cuerpo. NO cambian la estructura de la tabla, se utilizarán para darle estilos a los diferentes grupos de filas.

- `<thead>` que contendrá la fila o filas que sean la cabecera de una tabla.
- `<tbody>` que es donde pondremos las filas que son el contenido propiamente dicho de la tabla, el cuerpo.
- `<tfoot>` que es donde pondremos las filas que son el pie de nuestra tabla.



Un ejemplo sería:

```html
<!-- Tabla con estructura completa -->
<table>
    <colgroup>
        <col span="3" style="background-color: grey">
        <col style="background-color: yellow">
        <col style="background-color: green">
    </colgroup>
    <caption>Alumnos matriculados</caption>
    <thead>
        <tr>
            <th>Nombre</th>
            <th>Apellidos</th>
            <th>Dirección</th>
            <th>Teléfono</th>
            <th>Email</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Pepe</td>
            <td>Pérez</td>
            <td>Aquí</td>
            <td>11111111</td>
            <td>yosoy@pepe.es</td>
        </tr>
        <tr>
            <td>Manuel</td>
            <td>López</td>
            <td>Allí</td>
            <td>222222222</td>
            <td>yosoy@manuel.es</td>
        </tr>
        <tr>
            <td>María</td>
            <td>Fernández</td>
            <td>Mas allá</td>
            <td>33333333</td>
            <td>yosoy@maria.es</td>
        </tr>
        <tr>
            <td>Sara</td>
            <td>Gallardo</td>
            <td>Mas aquí</td>
            <td>44444444</td>
            <td>yosoy@sara.es</td>
        </tr>
    </tbody>
    <tfoot>
        <tr>
            <td>Pie de la tabla donde pongo más texto para que se vea como crecen</td>
        </tr>
    </tfoot>
</table>
```

Alumnos matriculados

Nombre	Apellidos	Dirección	Teléfono	Email
Pepe	Pérez	Aquí	11111111	yosoy@pepe.es
Manuel	López	Allí	222222222	yosoy@manuel.es
María	Fernández	Mas allá	33333333	yosoy@maria.es
Sara	Gallardo	Mas aquí	44444444	yosoy@sara.es
Pie de la tabla donde pongo más texto para que se vea como crecen				

Tablas Complejas

En la vida real, nos encontraremos con estructuras tabulares más complejas. Éstas también se pueden construir en HTML utilizando los siguientes atributos en las etiquetas de las celdas <td> <th>.

- `rowspan`**:** Indica el número de filas que ocupa la celda.
- `colspan:` Indica el número de columnas que ocupa la celda.

atributo	significado
colspan	Combina la celda actual con el número de celdas a la derecha que se indique. Por ejemplo *colspan="3"* une esta celda con las dos que tiene a su derecha, formando una combinación de tres celdas en horizontal.
rowspan	Combina la celda actual con el número de celdas hacia abajo que se indique. Por ejemplo *rowspan="3"* une esta celda con las dos que tiene hacia abajo, formando una combinación de tres celdas en vertical.

NOTA: Antes de escribir el código HTML de una tabla compleja es recomendable estudiar su estructura previamente.

Un ejemplo sería:

```
<table>
    <caption>HORARIO DE CLASE CURSO 2018-2019</caption>
    <thead>
        <tr>
            <th>HORAS</th>
            <th>Lunes</th>
            <th>Martes</th>
            <th>Miércoles</th>
            <th>Jueves</th>
            <th>Viernes</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>8:00</td>
            <td rowspan="2">Matemáticas</td>
            <td>Lengua</td>
            <td>Inglés</td>
            <td colspan="2">Ciencias</td>
        </tr>
        <tr>
            <td>9:00</td>
            <td>Historia</td>
            <td>Ciencias</td>
            <td>Matemáticas</td>
            <td>Lengua</td>
        </tr>
        <tr>
            <td>10:00</td>
            <th colspan="5">RECREO</th>
        </tr>
        <tr>
            <td>10:30</td>
            <td>Inglés</td>
            <td rowspan="2">Ciencias</td>
            <td>Matemáticas</td>
            <td>Lengua</td>
            <td>Historia</td>
        </tr>
        <tr>
            <td>11:30</td>
            <td>Historia</td>
            <td>Ciencias</td>
            <td>Matemáticas</td>
            <td>Inglés</td>
        </tr>
    </tbody>
</table>
```

Se han presentado las tablas, pero no se ha tratado el tema de los bordes que normalmente pueden presentar estas estructuras.

Los navegadores no les ponen borde a las tablas, si deseamos una tabla con bordes, se lo debemos indicar utilizando CSS.

Elemento `<style></style>`

Permite colocar código CSS en la página web. Se usa un atributo type para indicar el tipo de contenido, por ahora, el único admitido es **text/css**, si se omite, se tomará por defecto.

El elemento HTML style puede ser insertado únicamente en el encabezado del documento (**elemento HTML head**)

```
<head>
  <!--Etiqueta para definir estilos-->
   <style>
     /*Estilos para los bordes*/
        ...
   </style>
</head>
```

Tres tipos de bordes, como ejemplo, hay muchos más

- **Bordes Simple**
- **Bordes sin colapsar**
- **Bordes inferior o superior**

Bordes Simples

Pondremos dentro la etiqueta `<style>` lo siguiente:

```
<style>
    table {
       border-collapse: collapse;
    }
    td,
    th {
       border: 1px solid black;
    }
</style>
```

HORARIO DE CLASE CURSO 2018-2019

HORAS	Lunes	Martes	Miércoles	Jueves	Viernes
8:00	Matemáticas	Lengua	Inglés	Ciencias	
9:00		Historia	Ciencias	Matemáticas	Lengua
10:00	**RECREO**				
10:30	Inglés	Ciencias	Matemáticas	Lengua	Historia
11:30	Historia		Ciencias	Matemáticas	Inglés

Bordes sin Colapsar

Pondremos dentro la etiqueta `<style>` lo siguiente:

```
<style>
    table,
    td,
    th {
        border: 1px solid black;
    }
</style>
```

HORARIO DE CLASE CURSO 2018-2019

HORAS	Lunes	Martes	Miércoles	Jueves	Viernes
8:00	Matemáticas	Lengua	Inglés	Ciencias	
9:00		Historia	Ciencias	Matemáticas	Lengua
10:00	RECREO				
10:30	Inglés	Ciencias	Matemáticas	Lengua	Historia
11:30	Historia		Ciencias	Matemáticas	Inglés

Bordes Inferior/Superior

Pondremos dentro la etiqueta `<style>` lo siguiente:

```
<style>
    table {
        border-collapse: collapse;
    }
    td,
    th {
        border-bottom: 1px solid black;
    }
</style>
```

Si quisiéramos borde superior debemos cambiar *`border-bottom`* por *`border-top`*.

HORARIO DE CLASE CURSO 2018-2019

HORAS	Lunes	Martes	Miércoles	Jueves	Viernes
8:00	Matemáticas	Lengua	Inglés	Ciencias	
9:00		Historia	Ciencias	Matemáticas	Lengua
10:00	RECREO				
10:30	Inglés	Ciencias	Matemáticas	Lengua	Historia
11:30	Historia		Ciencias	Matemáticas	Inglés

FORMULARIOS

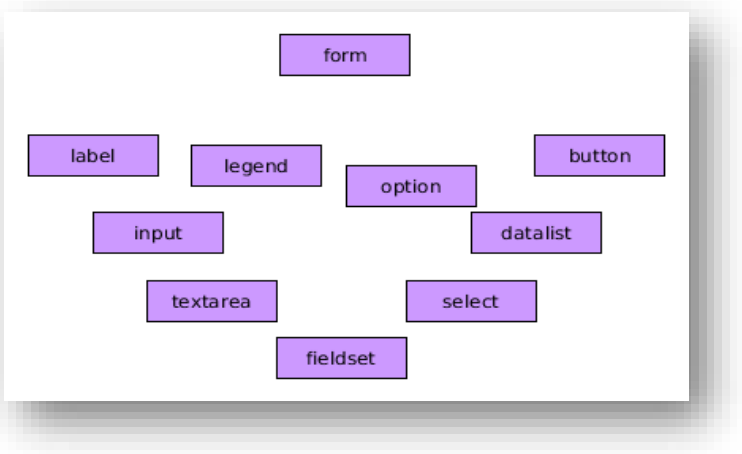
Los formularios son otros de los elementos que se han asociado desde siempre a las páginas webs. Los hemos usado tanto que es difícil calcular cuántos formularios habremos rellenado a lo largo de nuestra larga vida como usuario de la web.

Formularios de registro, usuario y contraseña, solicitudes etc. son contextos en los que los podemos encontrar. De un modo más formal podemos decir lo siguiente:

- Los formularios son elementos, que, como los enlaces, permiten una interacción del usuario con la página web.
- Su tarea principal es recoger información. El usuario, por tanto, debe introducir esa información en los campos del formulario.
- Una vez se ha recogido esa información el formulario la enviará al servidor, para ser tratada, mostrada y/o almacenada.

Estructura del formulario

En un formulario nos podemos encontrar con muchos elementos.



El primer el elemento y padre de la estructura DOM, es la etiqueta `<form>` que va a contener en su interior todos los elementos que conformen un formulario. Tiene su correspondiente etiqueta de cierre `</form>`.

Esta etiqueta puede tener varios atributos de entre los que destacamos:

- **method** que indica cómo se va a pasar la información al destino. Puede ser por **GET** (se ve la información en la barra del navegador, es la opción por defecto) y por **POST** (la información no es visible y va en el cuerpo del mensaje).
- **action** que indica el destino de nuestros datos. Normalmente será una URL o dirección de Internet. Existen más opciones. A continuación, vamos a ver algunos de los elementos más frecuentes y a describir su estructura y funcionamiento.

```
<form action="paginade_envío.php" method="post">
<form action="pagina_envío.php" method="get">
```

Elemento <input > y Elemento <label..>

Para la recogida de información los formularios usado etiquetas <input>.

La etiqueta <label> se utiliza **para asociar un texto descriptivo a un elemento de formulario**, como un campo de entrada (input), una casilla de verificación (checkbox), un botón de radio (radio button) o una selección (select).

Para saber qué campos estamos rellenando una buena práctica es poner una etiqueta **<label> delante de cada input** para dar nombre y asociar ambas utilizando for-id.

Se puede hacer de otras formas menos elegantes y correctas.

Ejemplo:

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

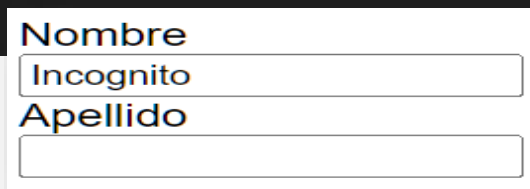
Estamos asociando la etiqueta al campo usando los atributos ***for*** e **id**

Los input tener muchos tipos, pero los más importantes son: **text, radio, submit, reset**

<input type=text >

El type= text simplemente coloca una caja de una línea donde poder introducir texto. El atributo, value permite colocar un valor en la caja por defecto

```
 <form action="paginaenvio.php" method="get">
     <label for="uno">Nombre</label><br>
     <input type="text" name="nombre" id="uno" value="Incógnito"><br>
     <label for="dos">Apellido</label><br>
     <input type="text" id="dos" name="apellido">
</form>
```

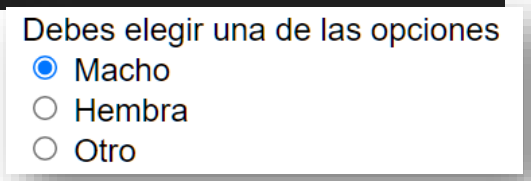


<input type=radio >

El type= radio coloca una serie de botones de opción única (deben tener el mismo name si queremos que sean excluyentes). El atributo value indica el valor que es enviado. El atributo checked indica la opción que es marcada por defecto

<input type="radio" name="genero" value="other" checked>

```
<form action="paginaenvio.php" method="get">
    <label for="nombre">Nombre</label><br>
    <input type="text" name="nombre" id="nombre"><br>
    <label for="sexo">Debes elegir una de las opciones</label><br>
    <input type="radio" name="sexo" value="hombre" checked> Macho<br>
    <input type="radio" name="sexo" value="mujer"> Hembra<br>
    <input type="radio" name="sexo" value="otro"> Otro<br>
</form>
```



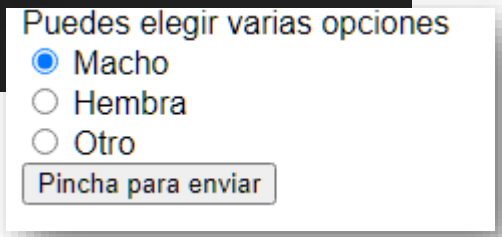
<input type= submit >

El type= submit coloca un botón que al ser pulsado envía los datos que hayamos introducido en el formulario. El atributo value indica el texto que aparece dentro del botón (si no se pone aparece por defecto Enviar)

<input type="submit" value="Pincha para enviar">

Se pueden elegir varias opciones puesto que los input radio tienen distinto valor en atributo name. Pero si es necesario elegir varias opciones, lo correcto es utilizar casillas de verificación, checkbox.

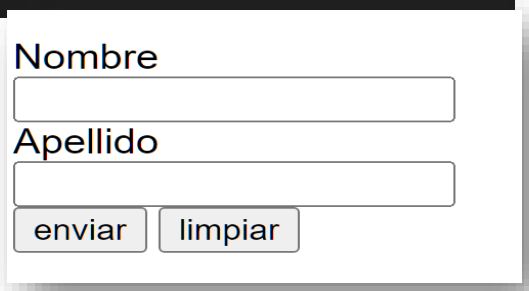
```
<form action="paginaenvio.php" method="get">
    <label for="nombre">Nombre</label><br>
    <input type="text" name="nombre" id="nombre"><br>
    <fieldset>
        <legend>Puedes elegir varias opciones</legend>
            <input type="radio" name="sexo1" value="hombre" checked> Macho<br>
            <input type="radio" name="sexo2" value="mujer"> Hembra<br>
            <input type="radio" name="sexo3" value="otro"> Otro<br>
    </fieldset>
    <input type="submit" value="Pincha para enviar">
</form>
```



<input type=reset >

El type= reset coloca un botón que al ser pulsado limpia todos los datos que hayamos introducido en el formulario. El atributo value indica el texto que aparece dentro del botón (si no se pone aparece por defecto Restablecer)

```
<form action="paginaenvio.php" method="get">
    <label for="uno">Nombre</label><br>
  <input type="text" name="nombre" id="uno" spellcheck="true"><br>

    <label for="dos">Apellido</label><br>
    <input type="text" id="dos" name="apellido1"><br>
    <input type="submit" value="enviar">
    <input type="reset" value="limpiar">
</form>
```



Listas desplegables <select>

<select> Lista desplegable (drop-down list) </select>

Nos muestra una caja donde podemos ver los distintos valores a seleccionar.

Hay que acompañarla de la etiqueta `<option value="uno">Ciclo SMR</option>` que contendrá los posibles valores a desplegar

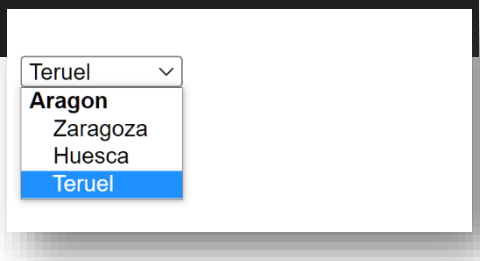
Son elementos que al hacer click sobre ellos nos muestran varias opciones de las que podremos escoger una o varias.

Para crear listas desplegables usaremos las etiquetas:

- `<select>` como etiqueta padre
- una etiqueta `<option>` para cada una de las opciones que tengamos en la lista desplegable.
- Si queremos agrupar esas opciones las meteremos a su vez dentro de una etiqueta `<optgroup>`.

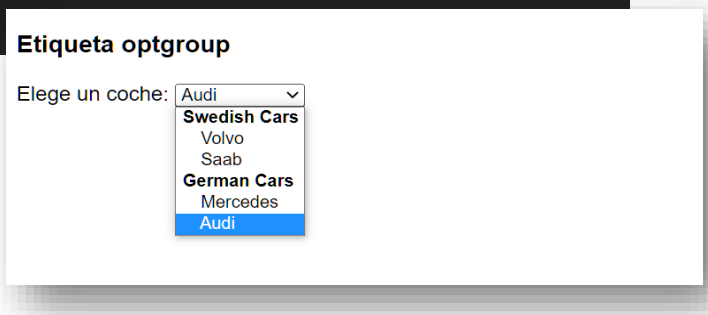
Ejemplo:

```
<form action="paginaenvio.php" method="get">
    <label for="n15">Selecciona una provincia:</label>
    <select name="provincia" id="n15">
        <optgroup label="Aragon">
            <option value="Z">Zaragoza</option>
            <option>Huesca</option>
            <option selected>Teruel</option>
        </optgroup>
    </select>
</form>
```



Otro ejemplo:

```
<form action="paginaenvio.php" method="get">
        <label for="cars">Elege un coche:</label>
        <select id="cars" name="cars" multiple>
            <optgroup label="Swedish Cars">
                <option value="volvo">Volvo</option>
                <option value="saab">Saab</option>
            </optgroup>
            <optgroup label="German Cars">
                <option value="mercedes">Mercedes</option>
                <option value="audi" selected>Audi</option>
            </optgroup>
        </select>
</form>
```



Áreas de Texto <textarea>

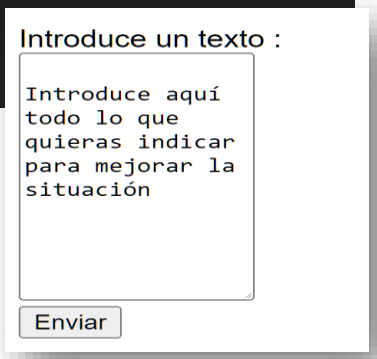
Nos muestra una caja donde podremos introducir texto (como en el elemento input type=text) pero que admite más de una línea

Admite los atributos:

- **rows:** para indicar el número de filas que tiene el campo.
- **cols:** para indicar el número de columnas que tiene el campo.

Un ejemplo sería:

```
<form action="paginaenvio.php" method="get">
     <label for="textoInformativo">Introduce un texto :</label><br>
     <textarea name="observacion" id="textoInformativo" rows="10" cols="15">
        Introduce aquí todo lo que quieras indicar para mejorar la situación
     </textarea><br>
     <input type="submit">
</form>
```



Botones <button>

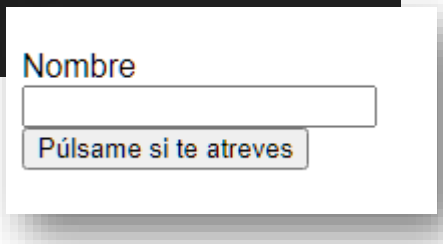
Nos muestra un botón sobre el que hacer clic.

Puede tomar como valores del atributo type: button, reset y submit.

Dentro del botón aparece el texto que pongamos antes de la etiqueta de cierre, también podemos poner una imagen en lugar del texto </button>

Muy utilizado con javascript, con el atributo: type=button, se mostrará igual en todos los navegadores

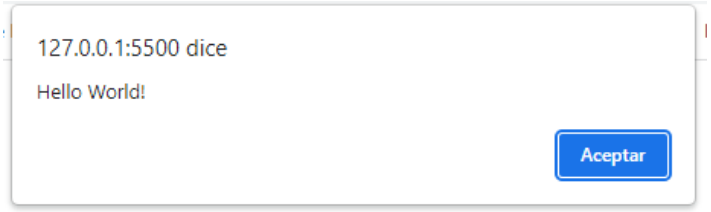
```
<form action="paginaenvio.php" method="get">
  <label for="uno">Nombre</label><br>
  <input type="text" name="nombre" id="uno" autocomplete="given-name">

  <button type="button">
      Púlsame si te atreves
  </button>
</form>
```



button: se comporta como un botón que puede estar presionado o no.

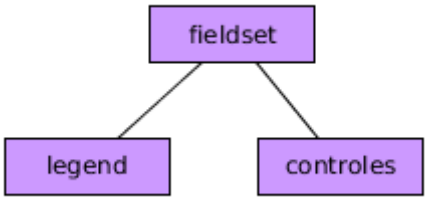
- Su función NO es enviar los datos del formulario.
- Solo enviará si le indicamos type="submit" y limpiará el formulario si es de type="reset"

```
<button type="button" onclick="alert('Hello World!')">
    Púlsame si te atreves
</button>
```



Agrupando elementos de un formularios

Hay situaciones en las que los campos de los formularios por significado, por importancia o por cualquier razón, queremos que se muestren agrupados. Para estos casos tenemos la etiqueta `<fieldset>` que la usaremos siguiente el siguiente esquema:



Siendo:

- `<legend>` una etiqueta que contiene el nombre del grupo y que se mostrará en la parte de arriba.
- **controles** todos los campos que queramos meter.

NOTA: la etiqueta tiene un borde por defecto

Un ejemplo:

```
<form action="paginaenvio.php" method="get">
        <fieldset>
            <legend>Datos Personales</legend>
            <p>
               <label for="n36">Nombre:</label>
               <input type="text" id="n36" name="nombre" placeholder="Inserta Nombre" required>
            </p><p>
                <label for="n37">Apellidos:</label>
                <input type="text" id="n37" name="apellidos" value="Pérez" readonly>
            </p><p>
                <label for="n38">Provincia</label>
                <select id="n38" name="provincia">
                    <optgroup label="Aragon">
                        <option value="Z">Zaragoza</option>
                        <option value="H">Huesca</option>
                        <option value="T" selected>Teruel</option>
                    </optgroup>
                </select>
            </p><p>
                <input type="submit" value="Enviar" />
            </p>
        </fieldset>
</form>
```

input tipo list

```
<input list="  ">
<datalist id="  ">
      <option value=" ">
      <option value=" ">
</datalist>
```

El elemento **<datalist>** es un elemento que contiene un conjunto de opciones predefinidas para un elemento <input>.

El elemento **<datalist>** se utiliza para proporcionar una función de "autocompletado" para los elementos <input>. Los usuarios verán una lista desplegable de opciones predefinidas a medida que introducen datos.

Con el atributo **datalist** indicamos la lista de opciones que ya ofrece el control y en la etiqueta **input** podemos agregar otras opciones, si las que nos muestra el control no es la que deseamos.

Para usar el elemento <datalist>:

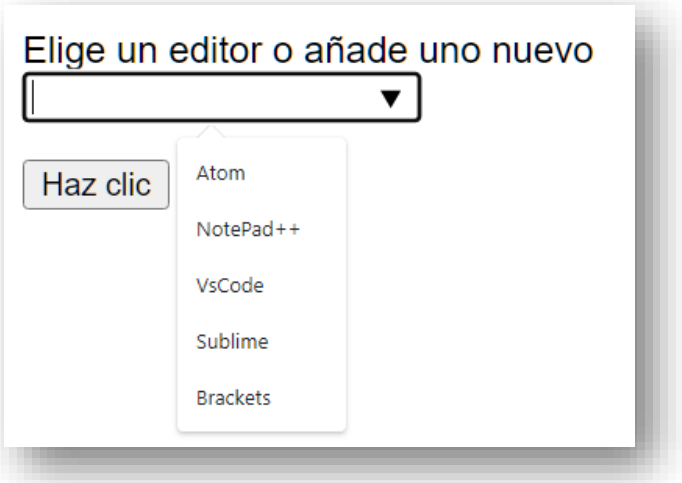
1. Se debe asignar un atributo id al elemento <datalist>
2. Y se debe asignar un atributo list al elemento <input> que haga referencia al id del <datalist>.

De esta forma, se vinculan entre sí.

El atributo list del input <input list="milista"> debe tener el mismo valor que id del elemento datalist, <datalist id="milista">

Un ejemplo sería:

```
<form action="paginaenvio.php" method="get">
   <label for="n21">Elige un editor o añade uno nuevo</label><br>
   <input list="editor" id="n21" >
   <datalist id="editor" name="unDataList">
         <option value="Atom">
         <option value="NotePad++">
         <option value="VsCode">
         <option value="Sublime">
         <option value="Brackets">
   </datalist>
   <p><button type="submit">Haz click</button></p>
</form>
```



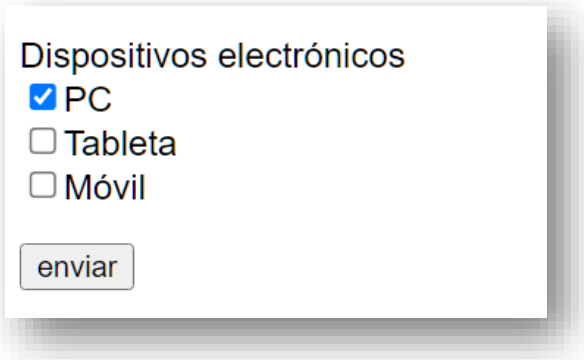
Casilla de verificación `<input type="checkbox">`

Es una agrupación de opciones, muestra una lista de las cuáles podemos **elegir una o varias** de las opciones. Para agrupar las opciones utilizar el ***type="checkbox"***, todas deben de tener el mismo valor para **name**.

Con el input checkbox, siempre se pueden elegir varias opciones. Si necesitar que solo se seleccione una opción debes utilizar la etiqueta input radio.

Un ejemplo sería:

```
<form action="paginaenvio.php" method="get">
    <label for="dispositivos">Dispositivos electrónicos</label><br>
    <input type="checkbox" name="dispositivos" value="pc" checked >PC<br>
    <input type="checkbox" name="dispositivos" value="table">Tableta<br>
    <input type="checkbox" name="dispositivos" value="movil">Móvil
    <p><input type="submit" value="enviar"></p>
</form>
```

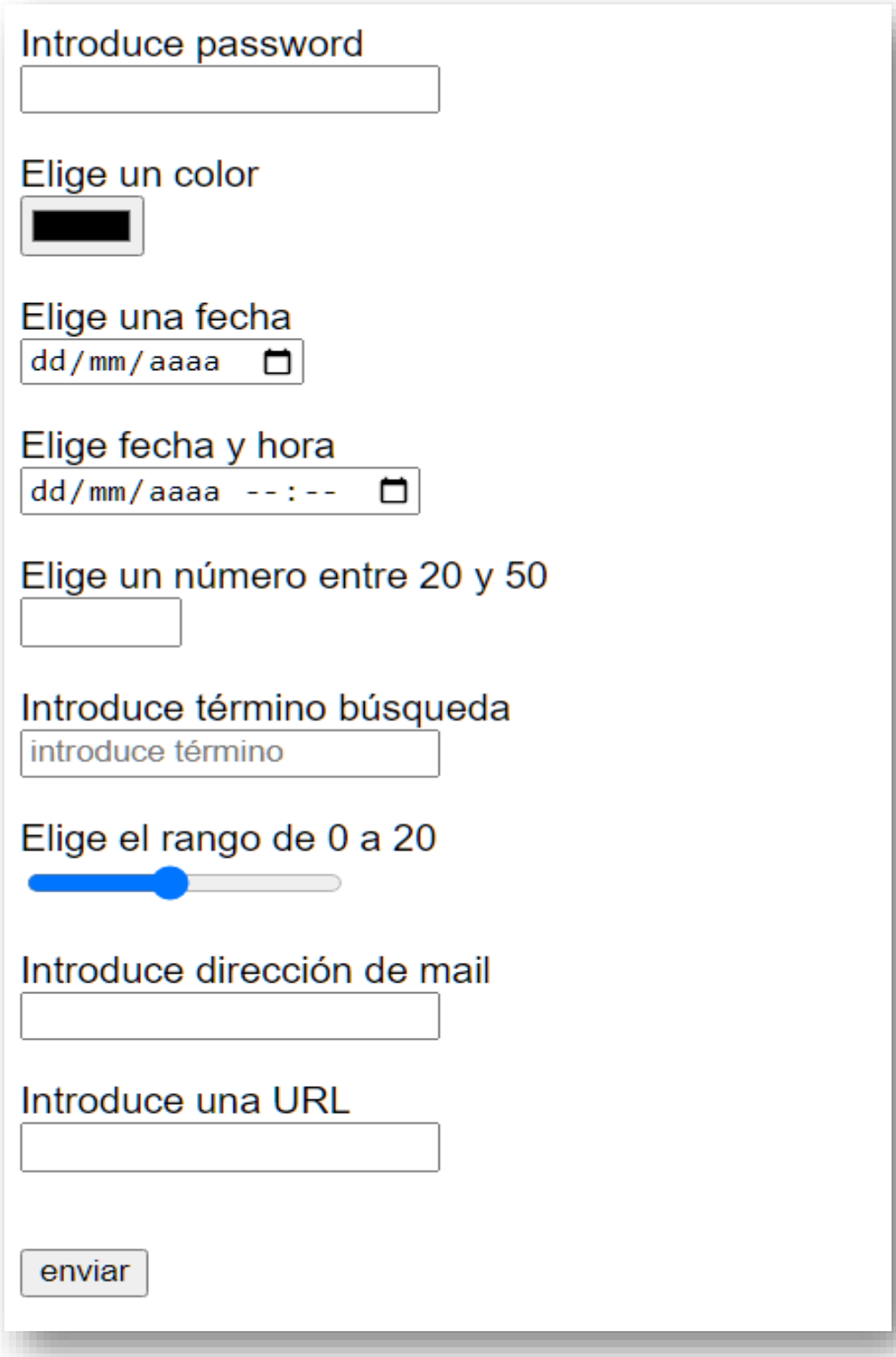


Otros tipos de input

```
<input type="password” >
<input type="color" >
<input type="date" >
<input type="datetime-local">
<input type="month">
<input type="week">
<input type="tel">
<input type="time">
<input type="number" min="0" max="10" >
<input type="search" >
<input type="range" min="0" max="10" step="5">
<input type="email">
<input type="url">
<input type="hidden" value="valorOculto">
<input type="file" >
```

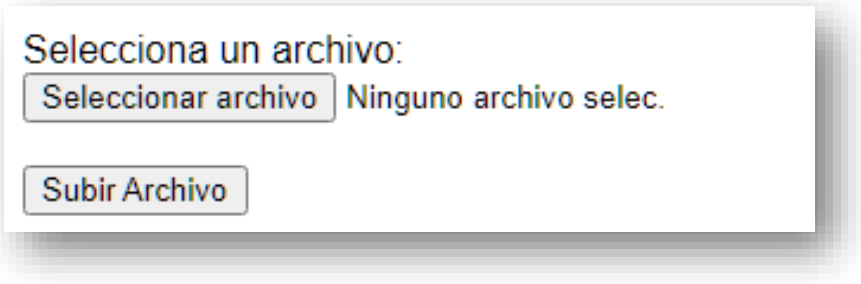
```
<form method="post" enctype="multipart/form-data"
action="archivo.php" >
<input type="file" name="archivo[]" multiple
accept="image/*,.pdf">
```

```html
<form action="paginaenvio.php" method="get">
      <label for="mi_clave" required>Introduce password</label><br>
      <input id="mi_clave"  name="una_clave" type="password"><br><br>

      <label for="mi_color">Elige un color</label><br>
      <input id="mi_color" name="un_color" type="color"><br><br>

      <label for="fecha">Elige una fecha</label><br>
      <input id="fecha" name="una_fecha"  type="date"><br><br>

      <label for="fechaHora">Elige fecha y hora</label><br>
      <input id="fechaHora"  name="una_hora" type="datetime-local"><br><br>

      <label for="numero">Elige un número entre 20 y 50</label><br>
      <input id="numero" name="un_numero" type="number" min="20" max="50"><br><br>

      <label for="busqueda">Introduce término búsqueda</label><br>
      <input id="busqueda" type="search" placeholder="introduce término"><br><br>

      <label for="rango">Elige el rango de 0 a 20</label><br>
      <input id="rango" name="un_rango"  type="range" min="0" max="20" step="3"><br>

      <label for="mail">Introduce dirección de mail</label><br>
      <input id="mail" name="un_mail" type="email"><br><br>

      <label for="internet">Introduce una URL</label><br>
      <input id="internet" name="una_url"  type="url"><br><br>

      <p><input type="submit" value="enviar"></p>
</form>
```


Introduce password
Elige un color
Elige una fecha
dd/mm/aaaa
Elige fecha y hora
dd/mm/aaaa --:--
Elige un número entre 20 y 50
Introduce término búsqueda
introduce término
Elige el rango de 0 a 20
Introduce dirección de mail
Introduce una URL
enviar

<input type="file">

```
<form action="fichero.php" method="post" enctype="multipart/form-data">
   <label for="archivo">Selecciona un archivo:</label>
   <input type="file" id="archivo" name="archivo" accept=".jpg,.png">
   <input type="submit" value="Subir Archivo">
</form>
```



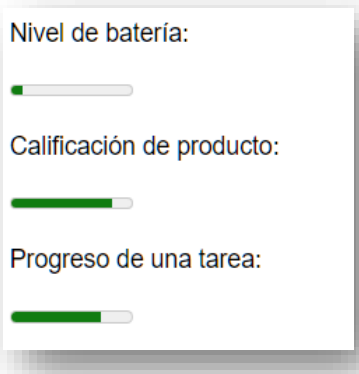
La etiqueta <meter></meter>

Se utiliza para representar una medida escalar dentro de un rango conocido, o una fracción en bruto. Por ejemplo, el uso del disco, el nivel de la batería o una puntuación en una encuesta.

La etiqueta <meter> muestra esta medida como una barra gráfica con un indicador que refleja el valor dentro del rango establecido por los atributos min y max.

El atributo **value** de la etiqueta <meter> es el valor actual de la medida.

```
<body>
    <p>Nivel de batería:</p>
    <meter value="0.10">10%</meter>

    <p>Calificación de producto:</p>
    <meter min="0" max="5" value="4.2">4.2 de 5</meter>

    <p>Progreso de una tarea:</p>
    <meter min="0" max="100" value="75">75% completado</meter>
</body>
```



El contenido que se coloca entre las etiquetas <meter></meter> se considera el "contenido alternativo" y se muestra en los navegadores que no admiten el elemento <meter>.

ATRIBUTOS ELEMENTOS HTML5

Los formularios HTML5 son una forma de recoger información del usuario a través de una página web. Los formularios HTML5 se componen de diferentes elementos, como campos de texto, botones, casillas de verificación, listas desplegables, etc. **Cada elemento tiene una serie de atributos que se pueden utilizar para personalizar su apariencia y comportamiento.**

Listado de los atributos más comunes que se pueden utilizar en los formularios HTML5:

- **action**: Especifica la URL a la que se enviará el formulario cuando el usuario lo envíe.
- **method**: Indica el método HTTP que se utilizará para enviar el formulario. Los valores más habituales son "get" y "post".
- **enctype**: Define el tipo de codificación que se utilizará para enviar los datos del formulario. Es necesario cuando se incluyen elementos de tipo “file” para subir archivos.
- **name**: Asigna un nombre al elemento del formulario. Este nombre se utiliza para identificar el elemento y enviar su valor al servidor.
- **id**: Asigna un **identificador único** al elemento del formulario. Este identificador se utiliza para referirse al elemento desde el código JavaScript o CSS.
- **type**: Define el tipo de elemento del formulario. Hay muchos tipos disponibles, como “text”, “email”, “password”, “number”, “date”, “file”, “checkbox”, “radio”, “submit”, etc. Cada tipo tiene sus propias características y validaciones.
- **value**: Establece el valor inicial del elemento del formulario. Este valor se puede cambiar por el usuario o por el código JavaScript.
- **placeholder**: Muestra un texto de ayuda en el elemento del formulario para indicar al usuario qué debe introducir. El texto desaparece cuando el usuario empieza a escribir.
- **required**: Indica que el elemento del formulario es obligatorio y no se puede dejar en blanco. Si el usuario intenta enviar el formulario sin rellenar el elemento, se mostrará un mensaje de error.
- **multiple**: Este atributo se utiliza en elementos <select> <input type="radio"> para permitir que el usuario seleccione varias opciones. Cuando se establece en “multiple”, el usuario puede seleccionar varias opciones manteniendo presionada la tecla “Ctrl” mientras hace clic en las opciones deseadas.
- **checked**: Este atributo se utiliza en elementos <input> de tipo "radio" o “checkbox” para indicar que la opción debe estar marcada por defecto. Cuando se establece en “checked”, la opción se selecciona automáticamente cuando se carga la página.
- **selected**: Este atributo se utiliza en elementos <option> para indicar que la opción debe estar seleccionada por defecto. Cuando se establece en "selected", la opción se selecciona automáticamente cuando se carga la página.
- **disabled**: Deshabilita el elemento del formulario, lo que significa que no se puede interactuar con él ni enviar su valor.
- **readonly**: Hace que el elemento del formulario sea de solo lectura, lo que significa que el usuario puede ver su valor pero no modificarlo

- **max**: Establece el valor máximo que se puede introducir en un elemento del formulario de tipo numérico o de fecha.
- **step**: Especifica el tamaño de los incrementos o decrementos para un elemento del formulario de tipo numérico o de fecha.
- **hidden**: Oculta un elemento de forma predeterminada, similar a aplicar reglas de CSS display: none; Ejemplo: <input type="hidden" value="Contenido oculto">
- **maxlength**: Limita el número máximo de caracteres que se pueden introducir en un elemento del formulario de tipo texto o textarea.
- **pattern**: Permite definir una expresión regular que debe coincidir con el valor introducido en un elemento del formulario de tipo texto o email.
- **autofocus**: Hace que un elemento del formulario obtenga el foco automáticamente cuando se carga la página. Solo puede haber un elemento con este atributo en cada página.
- **form**: Indica a qué formulario pertenece un elemento. Esto es útil cuando hay varios formularios en una página o cuando el elemento está fuera del elemento <form>.
- **novalidate**: Evita la validación HTML5 del formulario al enviarlo. Esto puede ser útil si se quiere usar una validación personalizada con JavaScript. Se debe colocar en elemento **form.**
- **spellcheck:** Este atributo se utiliza para habilitar o deshabilitar la verificación ortográfica en un campo de texto o textarea. Valores: true o false.
- **tabindex** para asegurarnos que podemos navegar por los elementos usando el tabulador podemos establecer un orden para cada uno. Ej. `tabindex="3"`
- **autocomplete**: Controla si el navegador debe ofrecer sugerencias de autocompletado para un elemento del formulario. Puede ser útil para ayudar al usuario a rellenar el formulario más rápidamente. Posibles valores para el atributo:
  - **on/ off**
  - **given-name**: Sugiere el nombre propio del usuario,
  - **family-name**: Sugiere el apellido del usuario,
  - **email**: Sugiere direcciones de correo electrónico,
  - **tel**: Sugiere números de teléfono.

- Y muchos más…

La lista de tipos (type) de inputs que podemos tener es muy larga. Muchos de ellos son nuevos en HTML5 por lo que es importante comprobar cómo se ven en los distintos navegadores.

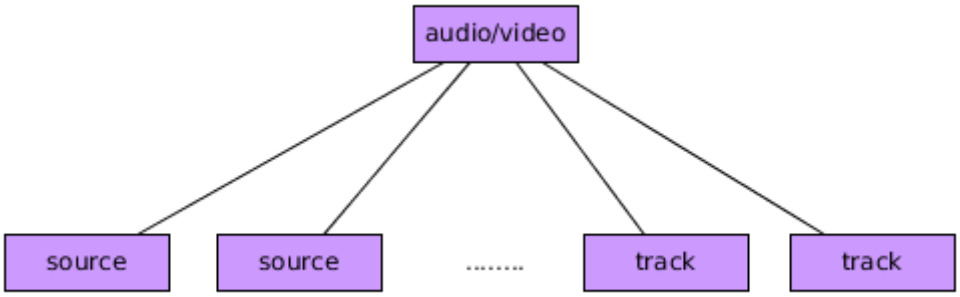
.

MULTIMEDIA

Una de las novedades más esperadas en HTML5 es la aparición de etiquetas específicas para multimedia.

En HTML5 se pretende que el contenido multimedia, (audios y vídeos) formen parte de la página y dejen de ser unos elementos incrustados. Se pretende por tanto que los nuevos navegadores se adapten a ello e incorporen recursos para poder reproducir este contenido sin tener que usar *plugins* externos.

Las más importantes son las etiquetas `<audio>` y `<video>` y para representarlas vamos a usar, de manera general estructuras de árbol similares a la siguientes (hay otros árboles válidos, pero este es más fácil y general):



- La etiqueta raíz será `<audio>` o `<video>` según lo que queramos mostrar en nuestro navegador.
- Dentro de esta etiqueta raíz tendremos tantas etiquetas `<source>` como formatos distintos de ese mismo elemento ofrezcamos. Es importante asegurarnos de que el formato del fichero multimedia es soportado por los navegadores. Si ofertamos varias fuentes el navegador decidirá cuál reproduce.
- Opcionalmente podremos añadir descripción de audio, subtítulos o similares usando una o varias etiquetas `<track>`.

Las etiquetas `<audio>` y `<video>` pueden tener los siguientes atributos.

- **controls** para mostrar los controles gráficos de reproducción (play, stop, pause, volumen).
- **autoplay** para que se empiece a reproducir al cargar la página de manera automática.
- **loop** si queremos que se reproduzca en bucle de manera infinita.
- **muted** si queremos que se reproduzca en silencio.

Adicionalmente la etiqueta `<video>` puede tener más atributos:

- `width` y `height` para indicar la anchura y altura solo en el control de vídeo.
- **poster** para mostrar una imagen mientras el vídeo se carga.

Etiqueta `<source>`

Tiene dos atributos fundamentales:

- **src** para indicar dónde está el fichero multimedia (una ruta).
- **type** para indicar el tipo de fichero, por ejemplo: `type="audio/mp3"` o `type="video/webm"`

Etiqueta `<track>`

Tiene 5 atributos fundamentales:

- **src** para indicar dónde está el fichero de descripción (una ruta).
- **kind** para indicar si es subtítulo, descripción etc. Tiene diversos valores que se pueden consultar en la documentación.
- **scrlang** que indica el código del idioma (es, en, fr,…) de la descripción. El navegador elegirá uno u otro dependiendo de la configuración el usuario.
- **label** la etiqueta que describe el idioma (español, inglés, etc..).
- **type** el tipo de fichero. Hay diversos, aunque lo más normal es *type=”text/vtt”*.

Un ejemplo con diferentes formatos y controles:

```
<h1>Música de la Pantera Rosa</h1>
<audio loop controls>
    <source src="ficheros/PanteraRosa.mp3" type="audio/mp3" />
</audio>
<h1>Queen Bohemian Rhapsody y subtítulos</h1>
<video width="800" height="600" controls>
    <source src="ficheros/BohemianRhapsody1.mp4" type="audio/mp4" />
    <!-- <track src="ficheros/BohemianRhapsody1.vtt" type="text/vtt" kind="subtitles"
    srclang="en" label="Inglés"> -->
    <track src="ficheros/BohemianRhapsody2.vtt" type="text/vtt" kind="subtitles" srclang="es"
    label="Español">
</video>

<!-- Etiqueta de vídeo-->
<h1>Banda Sonora Serie El HIERRO</h1>
<video width="320" height="240" controls>
    <source src="ficheros/elHIERRO.webm" type="video/webm">
    Tu navegador no soporta esta etiqueta (video)
</video>
```

Página para descargar archivos de vídeo de Internet y otra para convertir formatos

https://bitdownloader.io/

https://www.office-converter.com/

Archivos VTT

Los archivos VTT (WebVTT, "Timed Text Markup Language", Lenguaje de marcado de texto temporizado) son un formato de archivo utilizado para crear subtítulos o transcripciones de videos en línea.

El formato de los archivos VTT es el siguiente:

- La primera línea debe ser **WEBVTT** para indicar que se trata de un archivo VTT.
- Las líneas siguientes deben estar separadas por una línea en blanco y contener las indicaciones, que son los bloques de texto que se muestran en el vídeo.
- Cada indicación debe tener un marcador de tiempo de inicio y de fin, separados por una flecha (`-->`), que indica cuándo se debe mostrar y ocultar el texto.
- El formato del marcador de tiempo es `hh:mm:ss.mmm`
    - ✓ **hh** son las horas
    - ✓ **mm** los minutos
    - ✓ **ss** los segundos
    - ✓ **mmm** los milisegundos

- Por ejemplo, 00:01:23.456 significa 1 minuto, 23 segundos y 456 milisegundos.
- Debajo del marcador de tiempo debe ir el texto de la indicación, que puede tener varias líneas y algunos estilos como negrita, cursiva o subrayado. También se puede indicar el idioma, el nombre del hablante o el tipo de indicación (subtítulos, subtítulos o descripciones).

```
WEBVTT

00:00:05.500 --> 00:00:08.900
¿Es esto la vida real?

00:00:09.100 --> 00:00:12.100
¿Es solo fantasía?

00:00:12.300 --> 00:00:15.700
Atrapado en un derrumbe

00:00:15.900 --> 00:00:19.900
No hay escape de la realidad
```

Formatos, Contenedores y Codecs

Elementos de Video y Audio en HTML5: Los elementos **<video>** y **<audio>** en HTML5 proporcionan una manera estándar de reproducir archivos de vídeo y audio en navegadores web sin necesidad de plugins adicionales.

Plugins: Los plugins son pequeños programas o extensiones que se utilizan para agregar funcionalidades adicionales o mejoras a los programas o navegadores web. En el contexto de la reproducción de medios, los plugins solían ser necesarios para visualizar contenido multimedia en el pasado, pero HTML5 ha permitido que la mayoría de los navegadores admitan la reproducción de video y audio sin necesidad de plugins.

Formatos y Contenedores: Los formatos multimedia constan de un contenedor que almacena varios tipos de datos, como pistas de video, audio, metadatos, subtítulos y más. Algunos ejemplos de contenedores multimedia comunes incluyen:

- AVI: Un formato contenedor desarrollado por Microsoft.
- MPG: Utilizado para archivos MPEG (Moving Picture Experts Group).
- MOV: El formato de contenedor de QuickTime, desarrollado por Apple.
- ASF: Utilizado para archivos de Windows Media, como WMV y WMA.
- MP4: Un formato contenedor ampliamente utilizado para video y audio.
- Ogg: Un formato de contenedor abierto y gratuito desarrollado por la Fundación Xiph.
- OGM: Otro formato de contenedor de medios desarrollado por la Fundación Xiph.
- Wave (WAV): Utilizado para archivos de audio sin compresión.
- Matroska (MKV): Un formato de contenedor de medios de código abierto y ampliamente compatible.

Codecs: Los codecs son algoritmos de compresión y descompresión utilizados para comprimir pistas de video y audio en los archivos multimedia. Ejemplos de códecs:

- H.264 (también conocido como AVC): Ampliamente utilizado para vídeo de alta calidad.
- VP9: Un códec de vídeo de código abierto desarrollado por Google.
- AAC (Advanced Audio Coding): Comúnmente utilizado para audio en formato MP4.
- MP3: Un codec de audio muy conocido y utilizado.
- Vorbis: Un codec de audio de alta calidad utilizado en archivos Ogg.
- Opus: Un codec de audio de código abierto que ofrece una calidad excelente.

Compatibilidad del Navegador: Cada navegador web tiene su propia capacidad para reproducir diferentes formatos de medios. Algunos formatos son estándares y ampliamente admitidos, mientras que otros pueden ser propietarios o específicos de ciertos navegadores. Por lo tanto, es importante considerar la compatibilidad del navegador al elegir los formatos de medios a utilizar en tu sitio web para garantizar que la mayoría de los usuarios puedan reproducirlos sin problemas.

OBJETOS INCRUSTADOS

`<iframe>` `<object>` `<embed>`

Elemento `<iframe></iframe>`

`<iframe>` sirve para crear un espacio dentro de la página donde se puede incrustar otra web, es decir, un iframe **es utilizado para incrustar otro documento HTML dentro del documento actual.**

Atributos

- `src`: Especifica la URL del documento a incrustar.
- `width y height`: Especifican las dimensiones del marco.
- `sandbox`: Habilita una serie de restricciones de seguridad.
- `srcdoc`: Especifica el contenido HTML del documento a incrustar.
- `allowfullscreen`: Permite al marco entrar en modo de pantalla completa.

Es un cuadrado cuyas dimensiones se deben especificar en la propia página, con los atributos width y height en la propia etiqueta `<iframe>`.

El iframe tiene asociada una página web, que se carga en el espacio y operará de manera totalmente independiente. Esa página web tendrá sus propios contenidos y estilos.

Además, será perfectamente funcional: si tiene enlaces se mostrarán en ese mismo espacio y si tiene scripts o aplicaciones dentro se ejecutarán también de manera autónoma en el espacio reservado al <iframe>.

La integración de páginas web mediante el empleo de un elemento <iframe> pueden provocar agujeros de seguridad en tu sitio web.

HTML5 ha incorporado un atributo para mejorar la seguridad **sandbox.**

El atributo sandbox en la etiqueta <iframe> se utiliza para agregar restricciones de seguridad al contenido del iframe. Este atributo puede tomar varios valores, cada uno de los cuales habilita o deshabilita una cierta funcionalidad:

- allow-same-origin: Permite que el documento se trate como si estuviera en el mismo origen que el documento principal.
- allow-top-navigation: Permite que el documento cambie la URL del documento principal.
- allow-forms: Permite que el documento envíe formularios.
- allow-scripts: Permite que el documento ejecute scripts.
- allow-popups: Permite que el documento abra ventanas emergentes.
- allow-pointer-lock: Permite que el documento use el bloqueo del puntero.
- allow-modals: Permite que el documento abra diálogos modales.
- allow-presentation: Permite que el documento inicie una presentación en pantalla completa.

Pero si añadimos un atributo vacío al elemento <iframe>:

```
<iframe src="https://www.ejemplo.com" sandbox></iframe>
```

El contenido aislado del <iframe> queda realojado dentro del navegador aplicándosele las siguientes restricciones, la página incrustada tiene prohibido:

- Ejecutar scripts (JavaScript).
- Enviar o recibir cookies.
- Crear ventanas emergentes.
- Cambiar la URL del documento principal.
- Acceder al almacenamiento local.
- Enviar formularios.

```
<h2>Incrustando Google Maps</h2>
<iframe
    src="https://www.google.com/maps/embed?pb=!1m14!1m12!1m3!1d
24099.832553579043!2d-
5.674649376776689!3d40.9709894667372!2m3!1f0!2f0!3f0!3m2!1i1024!
2i768!4f13.1!5e0!3m2!1ses!2ses!4v1699469965719!5m2!1ses!2ses"
    width="800"
    height="600"
    style="border:0;"
    allowfullscreen
    loading="lazy"
    sandbox="allow-scripts allow-same-origin allow-popups">
    Este navegador no soporta iframe
</iframe>
```

`loading="lazy"` este atributo se usa para retrasar la carga de imágenes y iframes que están fuera de la pantalla hasta que el usuario se desplace cerca de ellos. Esto mejora el rendimiento de la página al reducir la cantidad de contenido que necesita ser cargado cuando la página visualiza por primera vez.

Para incrustar una página de Google Maps en un iframe, necesitas obtener el enlace de incrustación de la página de Google Maps:

1. Abre Google Maps en tu navegador.
2. Busca la ubicación que deseas incrustar.
3. Haz clic en el botón "Compartir" y selecciona "Incrustar un mapa".
4. Copia el enlace proporcionado.

Elementos <object> <embed>

Los elementos object, embed e iframe en HTML se utilizan para incrustar contenido externo en una página web:

- <iframe> se utiliza para incrustar otra página HTML dentro de la actual. Es muy útil para **incrustar contenido de terceros**, como mapas de Google o videos de YouTube.
- <object> puede ser una buena opción, si estás incrustando un tipo de contenido específico como un PDF.
- <embed> casi no se utiliza debido a problemas de seguridad y compatibilidad, ya que no dispone del atributo sandbox.

Los elementos <embed> y <object> tienen una función diferente a la de <iframe>, estos elementos son herramientas de incrustación de propósito general para incrustar múltiples tipos de contenido externo, que incluyen tecnologías de plugin como Java Applets y Flash, PDF (que puede mostrarse en un navegador con un plugin de PDF), e incluso contenido como vídeos, SVG e imágenes.

Sin embargo, es poco probable que utilices estos elementos: los Applets no se utilizan desde hace años, Flash ya no es muy popular, debido a una serie de razones, los **PDFs tienden a estar mejor enlazados que incrustados**, y otros contenidos como las imágenes y el vídeo tienen elementos mucho mejores y más fáciles de manejar.

Un plugin es un software que proporciona acceso a contenidos que el navegador no puede leer de forma nativa.

Hace tiempo, los plugins eran indispensables en la web. ¿Recuerdas los días en que tenías que instalar Adobe Flash Player sólo para ver una película online? Y luego recibías constantemente molestas alertas sobre la actualización de Flash Player y tu Java Runtime Environment.

Las tecnologías web se han vuelto mucho más robustas, y esos días han terminado. Para la mayoría de las aplicaciones, ha llegado el momento de dejar de ofrecer contenidos que no dependan de plugins y empezar a aprovechar las tecnologías web.

<object></object>

```
<h2>Incrustando un pdf con object</h2>
<object
   width="800"
   height="600"
   type="application/pdf"
   data="https://www.iesvenancioblanco.es/phocadownload/solicitud%20titulos2022.pdf">
      Si tu navegador no tiene plugins de pdf, puedes acceder al archivo con un enlace
      <a href="https://www.iesvenancioblanco.es/phocadownload/solicitud%20titulos2022.pdf">
        Solicitud Título
      </a>
</object>
```

<embed>

```
<h2>Incrustando un pdf con embed(no tiene etiqueta de cierre)</h2>
<embed
  src="https://www.iesvenancioblanco.es/phocadownload/solicitud%20titulos2022.pdf"
  width="700"
  height="500"
  type="application/pdf">
```

ELEMENTOS INTERACTIVOS

DETALLES <DETAILS> </DETAILS>

La etiqueta <details> permite mantener oculta una parte de la página web, que el usuario puede hacer visible u ocultar de nuevo haciendo clic en un icono con forma de triángulo.

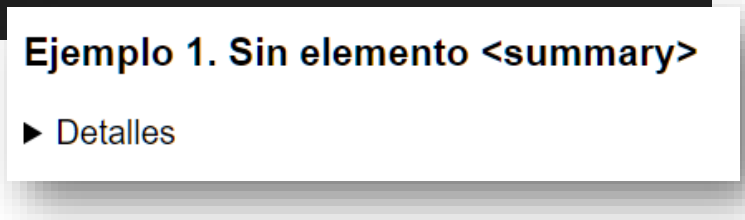
El atributo **open** **(opcional)** hace que el contenido de <details> se muestre cuando se carga la página, aunque se pueda ocultar posteriormente.

RESUMEN <SUMMARY></SUMMARY>

El elemento <summary> se puede incluir dentro de un elemento <details> y contiene la parte del contenido que está siempre visible y sobre la que también se puede hacer clic para mostrar el resto del contenido,

Si no se incluye el elemento <summary>, los navegadores muestran el texto "Detalles"

```
<h3>Ejemplo 1. Sin elemento &lt;summary&gt;</h3>
<details>
  <p>Esto es un párrafo de ejemplo, permanecerá oculto hasta que hagas clic en el triángulo.</p>
</details>
```



```
<h3>Ejemplo 2. Con elemento &lt;summary&gt;</h3>
<p>En la película de 1968 <i><b>El Planeta de Simios</b></i> tres astronautas llegan a un planeta. En la que los simios son la raza dominante y los humanos bestias que no saben hablar. El protagonista en la escena final descubre ...</p>
<details>
    <summary>Spoiler</summary>
        <p>que está en la Tierra.</p>
    </details>
```

Ejemplo 2. Con elemento <summary>

En la película de 1968 ***El Planeta de Simios*** tres astronautas llegan a un planeta. En la que los simios son la raza dominante y los humanos bestias que no saben hablar. El protagonista en la escena final descubre ...

▶ Spoiler

```
<h3>Ejemplo 3. Con atributo open</h3>
<p>Formulario de registro</p>
<details open>
   <summary>Datos personales</summary><br>
   <label for="nombre">Nombre: </label>
   <input type="text" name="nombre" id="nombre"><br>
</details>
<br>
<details open>
   <summary>Datos profesionales</summary><br>
   <label for="empresa">Empresa: </label>
   <input type="text" name="empresa" id="empresa"><br>
</details>
```

Ejemplo 3. Con atributo open

Formulario de registro

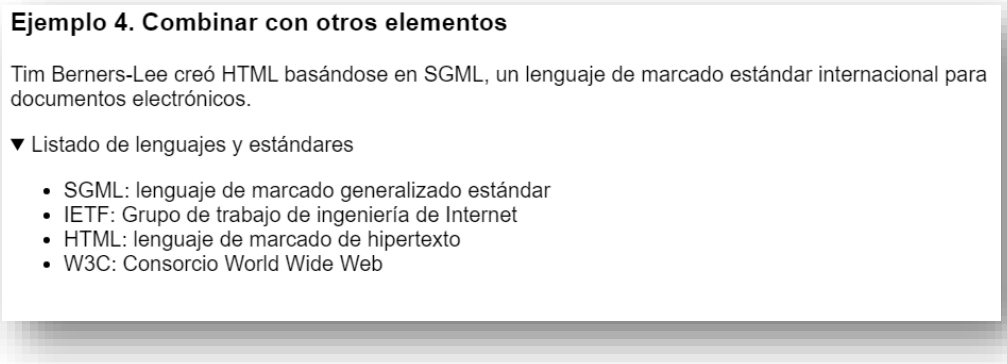
▼ Datos personales

Nombre:

▼ Datos profesionales

Empresa:

```html
<h3>Ejemplo 4. Combinar con otros elementos</h3>
<p>Tim Berners-Lee creó HTML basándose en SGML, un lenguaje de marcado estándar internacional para documentos electrónicos.</p>
<details>
  <summary>Listado de lenguajes y estándares</summary>
  <ul>
     <li>SGML: Standar Generalized Markup Language</li>
     <li>IETF: Internet Engineering Task Force</li>
     <li>HTML: HyperText Markup Language</li>
     <li>W3C: World Wide Web Consortium</li>
  </ul>
</details>
```



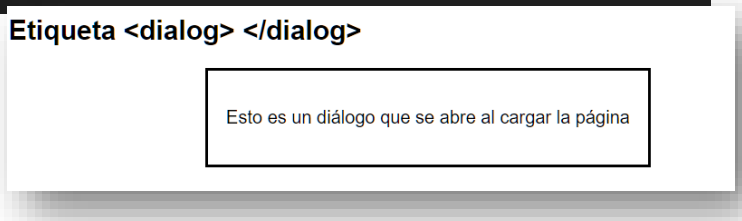
VENTANA DE DIÁLOGO <DIALOG>

El elemento <dialog> se utiliza para crear un **cuadro de diálogo modal** que se muestra encima del contenido principal de la página y que generalmente contiene información importante o solicitudes de interacción del usuario.

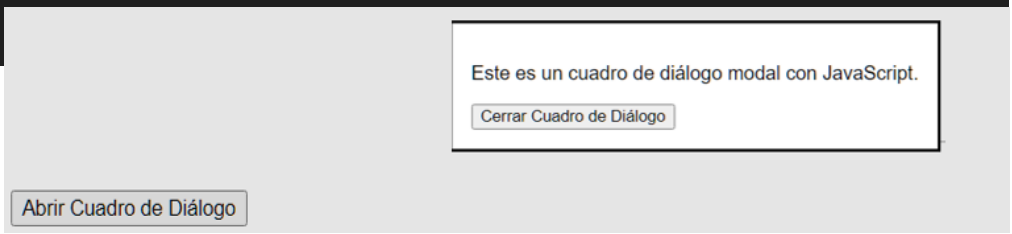
Atributos open (opcional): Si se incluye, el cuadro de diálogo se mostrará abierto de forma predeterminada cuando se cargue la página.

Se puede utilizar JavaScript para controlar la apertura y cierre del cuadro de diálogo.

```html
<h2>Etiqueta &lt;dialog&gt; &lt;/dialog&gt;</h2>
<dialog open>
  <p>Esto es un diálogo que se abre al cargar la página</p>
</dialog>
```



```html
<button id="abrirDialogo" onclick="openModal()">
      Abrir Cuadro de Diálogo
</button>
<dialog id="unDialogo">
    <p>Este es un cuadro de diálogo modal con JavaScript.</p>
    <button id="cerrarDialgo" onclick="closeModal()">
       Cerrar Cuadro de Diálogo
    </button>
</dialog>
<!--Código JavaScript -->
<!--Programando los botones de apertura y cierre  -->
<script>
   function openModal() {
      const dialogElement = document.getElementById("unDialogo");
      dialogElement.showModal();
   }
   function closeModal() {
      const dialogElement = document.getElementById("unDialogo");
      dialogElement.close();
   }
</script>
```



Mapas de imágenes

Etiquetas <map></map> y <area>

La etiqueta <map> en HTML5 se utiliza junto con las etiquetas <area> para crear mapas de imágenes interactivos.

Un mapa de imagen permite definir áreas específicas de una imagen como hipervínculos, lo que permite a los usuarios hacer clic en partes de la imagen para navegar a diferentes páginas o realizar otras acciones.

Elemento img del mapa

Al mapa tendremos que asociar un elemento <**img**> para conseguir tener las áreas enlazables.

`Nombre del Mapa`

Una de pasos más importantes para crear los mapas de imágenes es crear un elemento imagen con el atributo `usemap`, cuyo valor debe llevar `#`. Este nombre será el que enlacemos sobre la imagen para poder usar el mapa de imágenes.

```
<img src="imagenes/dibujo.jpg"
    alt="dibujo"
    usemap="#miMapa"
    width="400" height="379">
```

El elemento <map> anida un conjunto de elementos <area>. Los elementos <area> serán los que establezcan las zonas enlazables dentro de la imagen.

Para darle nombre del mapa de imágenes se utiliza el atributo name.

```
<map name="miMapa">
    <area shape=" ">
    <area shape=" ">
</map>
```

Tipos de áreas, atributo shape

```
<area shape="forma" coords="coordenadas" href="enlace" alt="texto" />
```

Dentro de los tipos de áreas que podemos crear dentro de una imagen tenemos diferentes formas:

- `shape` es la figura usada para definir el área
- `coords` el conjunto de coordenadas que define la forma. Dependiendo de la forma utilizada serán unas coordenadas u otras.
- `href` contendrá el enlace
- `alt` el texto alternativo a ese enlace

Atributo shape
Especifica la forma de la figura usada para definir un área en un mapa de imagen. Los valores son insensibles a mayúsculas/minúsculas y deben corresponderse con los listados a continuación:

- **default**: selecciona toda el área.
- **rect**: define una región rectangular.
- **circle**: define una región circular.
- **poly**: define una región poligonal.

Atributo coords
Especifica las coordenadas y la posición de la figura definida en el atributo "shape". Las coordenadas dependen de la figura utilizada:

- rect (rectángulo): si se establece shape="rect", las coordenadas en el atributo coords deben representar un rectángulo. Hay que proporcionar cuatro valores numéricos separados por comas en el atributo coords, que representan las coordenadas del punto superior izquierdo (x1, y1) y el punto inferior derecho (x2, y2) del rectángulo.

  Ejemplo, "0, 0, 10, 10" define un cuadrado con diez píxeles de lado que toca los bordes superior e izquierdo de la imagen).

```
<area shape="rect" coords="0,0,10,10" alt=" " href=" ">
```

- circle (círculo): si se establece shape="circle", las coordenadas en el atributo coords deben representar un círculo. Hay que proporcionar tres valores numéricos separados por comas en el atributo coords, que representan las coordenadas del centro del círculo (x, y) y el radio (r) del círculo.

  Ejemplo "20, 20, 30" define un círculo en el punto (20,20) con un radio de 30 píxeles.

```
<area shape="circle" coords="20,20,30" alt=" " href=" ">
```

- poly (polígono): si se establece shape="poly", se espera que las coordenadas en el atributo coords representen un polígono, es decir, una forma con múltiples lados. Hay que proporcinar una lista de valores numéricos separados por comas en el atributo coords, donde **cada par de valores (x, y)** representa un punto en el polígono.
  - ✓ El navegador conectará estos puntos para formar un polígono.
  - ✓ Se pueden diseñar polígonos irregulares, según las necesidades
  - ✓ Se recomienda que la primera coordenada y la última coincidan o sean cercanas, de lo contrario, el navegador cerrará el polígono de la mejor forma posible.

```
<area shape="poly"
   coords="22,300,22,333,200,333,200,350,34,350,22,300"
   alt=""
   href=" "/>
```

Las coordenadas son relativas a la esquina superior izquierda del objeto asociado y dependen del tamaño con el que se muestre la imagen. Si el tamaño cambia, las coordenadas NO coincidirán.

Atributo `href`

El atributo "`href`" especifica un recurso de destino al que iremos cuando se haga clic en las diferentes áreas.

Lienzo para dibujar <canvas> </canvas>

CANVAS es una nueva etiqueta del HTML 5, permite dibujar en dicha área mediante JavaScript.

Con el elemento canvas especificamos un área dentro de la página en la que podemos dibujar, poner imágenes, etc., todo ello mediante scripts (**necesitaremos saber JavaScript**), los resultados pueden ser parecidos a emplear flash, con la ventaja de que no necesitamos utilizar *plugins* externos.

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Canvas</title>
    <script>
        window.onload = function () {
            var lienzo = document.getElementById("micanvas");
            var contexto = lienzo.getContext('2d');
            contexto.fillRect(20, 10, 260, 110);
        }
    </script>
</head>

<body>
    <h1>Primer dibujo con canvas</h1>
    <canvas id="micanvas" width="300" height="150">
        Si ves este texto es que tu navegador no admite canvas;
    </canvas>
</body>

</html>
```

ETIQUETAS SEMÁNTICAS

Las etiquetas semánticas ayudan a definir la estructura del documento y permiten que las páginas web sean mejor indexadas por los buscadores.

Son etiquetas que indican **qué son los datos** que contienen, en lugar **de cómo se debe formatear** al mostrar el documento HTML en un cliente web(navegador).

Una etiqueta se califica como semántica si nos informa sobre lo que trata su contenido.

Ganan mucha importancia en el marco del HTML y de la composición de un documento web por ayudar a motores de búsqueda como Google a indexar más correctamente los contenidos de un sitio.

Dentro del etiquetado semántico también tenemos varias funciones, pero las más importantes son aquellas que nos ayudan a maquetar páginas web. Son las que sirven para definir el esquema principal del documento.

Por ejemplo, la etiqueta <section> nos dice que contiene una sección o capítulo dentro de la página.

Frente a las etiquetas semánticas tenemos otros tipos de etiquetas como las que afectan al formato <video> …

También son etiquetas semánticas que señalan elementos concretos de la página:

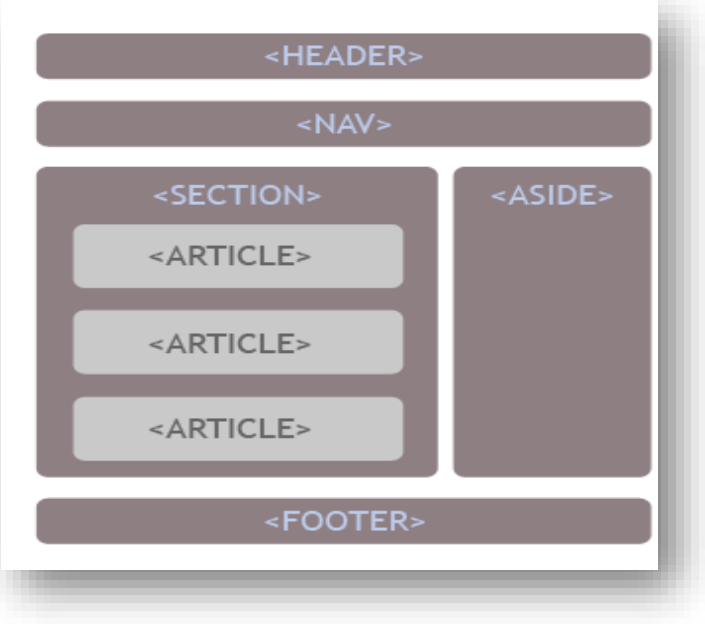
- la etiqueta <figure>, para imágenes y gráficos,
- la etiqueta <figcaption>, que es un pie de imagen,
- la etiqueta <time> para definir fechas y horas,
- las etiquetas <sumary> y <details> que representan un contenido oculto que será mostrado a petición del usuario

Etiquetas semánticas estructurales

```
<header> <main> <nav> <article>
<section> <aside> <footer>
```

Antes de la existencia de las etiquetas semánticas, el contenido se estructuraba con etiquetas que no aportaban ninguna información sobre el tema que trataban, salvo que se añadiese aprovechando el valor dado a la propiedad ID, o a la propiedad CLASS para hacer referencia a su contenido.

Las etiquetas a las que vamos a prestar más atención son las que definen la estructura general del sitio (MAIN, HEADER, ARTICLE, SECTION, ASIDE, FOOTER, NAV).



HEADER <header></header>

Se utiliza como una introducción del elemento que la contiene. Si colocamos un HEADER en el BODY, a primer nivel, será una introducción o presentación de toda la página. Si colocamos un HEADER dentro de un SECTION, contendrá una introducción a esa sección. También se pueden colocar dentro de un ARTICLE.

El HEADER suele llevar el título, la descripción corta y el logo de la página. El HEADER por si mismo no separa contenidos, no se debe utilizar aislado, sino dentro de otra etiqueta semántica, o a primer nivel en el BODY.

El HEADER no es equivalente a la parte superior de la página, aunque frecuentemente está en esa posición.

Es bastante frecuente utilizar las etiquetas H1, H2, ... para escribir los títulos dentro del HEADER.

SECTION <section></section>

Una SECTION agrupa contenido con un tema común, por ejemplo, las diferentes partes de un periódico: política, sociedad, deportes, ...

Un sitio de cocina podría estar organizado en secciones. Una SECTION podría ser recetas de carne, otra receta de pescado y otra de verduras. Dentro de cada SECTION tendríamos un ARTICLE por cada receta.

Cada SECTION puede estar encabezada por una etiqueta H1, H2,... dentro de una SECTION suelen haber etiquetas ARTICLE, y también etiquetas y <p>.

Una SECTION no debe ser usada con el único fin de dar formato, para eso se utiliza la etiqueta .

ARTICLE <article></article>

Representa una unidad de contenido, es decir, contenido que responde a un tema concreto. El ejemplo más claro es un artículo dentro de una revista, por ejemplo, en una página de cocina, un ARTICLE sería cómo elaborar un determinado plato. En un foro, un ARTICLE sería un post, o entrada en el foro.

Se pueden anidar un ARTICLE dentro de otro ARTICLE. Dentro de un ARTICLE el contenido se suele estructurar con las etiquetas DIV y P.

También está permitido poner dentro de un ARTICLE una SECTION, siempre que el artículo sea largo y tenga diferentes secciones, pero lo más natural es que el ARTICLE vaya dentro de una SECTION. Más adelante trataremos cómo crear estructuras de páginas complejas, con tres niveles.

NAV <nav></nav>

Esta etiqueta está pensada para contener la barra de navegación o los enlaces a las páginas del sitio web.

Normalmente va en la parte superior de la página o en un lateral. Suele estar formada por una lista de enlaces con las etiquetas UL y LI. Aunque hay multitud de diseños diferentes de barras de navegación y menús.

ASIDE <aside></aside>

Esta etiqueta está diseñada para contenido tangencial, es decir, menos importante que el cuerpo de la página, y que suele ser de un tema relacionado indirectamente (o no relacionado) con el tema principal.

Se suele colocar en los laterales de la página. Pero no todo lo que va en los laterales es ASIDE. Por ejemplo, si la página tiene publicidad en el lateral derecho, puede colocarse en una etiqueta ASIDE, también una pequeña reseña sobre el autor podría ir en esta etiqueta.

FOOTER <footer></footer>

Normalmente va al final de la página y contiene información del tipo: autor, copyright, contacto, mapa del sitio, etc.

Se pueden poner HEADER y FOOTER dentro de SECTION y ARTICLE, aunque no es lo más normal. No hay nada prohibido, pero debe tener cierta lógica o sentido semántico.

MAIN <main></main>

Tiene como función contener los temas principales de un documento.

Este contenido central puede interpretarse como todo lo que resta después de quitar anuncios, encabezados (header), pies (footer), secciones de navegación (nav), vínculos relacionados y otros elementos secundarios.

No debe haber más de un elemento <main> en un documento, y este no debe ser descendiente de un elemento <article>, <aside>, <footer>, <header>, o <nav>.

Los únicos elementos que puede tener como antepasados la etiqueta <main> en HTML son <html>, <body>, y <form>.

Su principal función es mejorar la accesibilidad. La etiqueta HTML <main> es muy útil para las aplicaciones especiales desarrolladas para personas con discapacidades, tales como navegadores de voz o lectores de pantalla, ya que pueden ir directamente al tema principal del documento.

Las aplicaciones desarrolladas para ayudar a personas con discapacidad reconocen la etiqueta **main** y proveen un atajo para que el usuario pueda saltar directamente al contenido de esta etiqueta sin tener que pasar por el resto.

Es sumamente recomendable añadir el rol ARIA "main" en un elemento <main>:

```
<main role="main">
  ...
</main>
```

Especificación **ARIA** (Accessible Rich Internet Applications) define una forma de hacer que el contenido y las aplicaciones web sean más accesibles para las personas con discapacidad.
Es una colección de atributos que definen como realizar contenido y aplicaciones web (especialmente las desarrolladas con Javascript) más accesibles para las personas con discapacidades, cuando los problemas de accesibilidad no se pueden gestionar con HTML nativo, ARIA puede ayudar a cerrar esas brechas.

role es uno de estos atributos que nos permite pasar información a los navegadores acerca del propósito de ese elemento, permitiendo escanear rápidamente el contenido de nuestra web, haciéndola más accesible.

<header role=”banner”></header>
<nav role=”navigation”></nav>
<main role=”main”></nav>
<article role=”article”></article>
<aside role=”complementary”></aside>

```
<body>
    <header>
        <div id="logo">El logo</div>
        <nav>…</nav>
    </header>
    <main role="main">
        <section id="productos">
            <h1>Nuestros Productos</h1>
            <p>Aquí encontrarás los productos que vendemos.</p>
            <article id="estrella">
                <h2>Producto Estrella</h2>
                <p>…</p>
            </article>
            <article id="barato">
                <h2>Producto barato</h2>
                <p>…</p>
            </article>
        </section>
    </main>
    <footer>…</footer>
</body>
```

Conclusiones

El uso de las etiquetas semánticas es bastante simple e intuitivo, según las características de cada una que acabamos de ver.

Realmente no existen normas fijas, cada diseñador puede darles el uso que considere más apropiado. El único límite es el sentido común, no debemos perder de vista que los buscadores las utilizarán para indexar la página, por lo tanto, nos interesa usar las etiquetas semánticas con la lógica para la que fueron diseñadas.

Es importante aclarar que las etiquetas semánticas no añaden ningún formato. Es decir, por usar <nav> no se crea un menú, ni el <header> tiene un tipo de letra más grande. El formato debemos definirlo nosotros con CSS. La excepción es que muchos navegadores incorporan la propiedad **display: block** a las etiquetas semánticas.

Etiqueta y Etiqueta

Las etiquetas y , no tienen ningún tipo de significado semántico especial.

Se utilizan actualmente en un documento web, **para establecer opciones de formato mediante hojas de estilo a bloques completos** **o elementos individuales** **del documento**.

** ** se define como **un elemento de bloque** y por tanto el navegador mostrará un salto de línea antes y después de la misma. La etiqueta **div** se emplea para definir un bloque de contenido dentro de **section**, **article**, **aside** o **footer**, y poder aplicar diferentes estilos sobre cada bloque específico, utilizando CSS.

**** es un elemento **contenedor en línea genérico**, span en sí misma no tiene ningún efecto en su contenido, a menos que tú le des estilos.

Las etiquetas *span* generalmente envuelven secciones de texto con fines de estilo o para agregar atributos a una sección de texto **sin crear una nueva línea de contenido**. Se utiliza normalmente en conjunto con los atributos **style** y **class** .

Antes de HTML5, su uso muy común, era utilizarlas para el diseño del documento mediante hojas de estilo. Aunque actualmente con la nueva versión de HTML5 que proporciona nuevas etiquetas para establecer dicho diseño, se recomienda la utilización de las nuevas etiquetas en sustitución de div y span.

Ejemplo de uso de la etiqueta :

```
<p>Párrafo con formato <span style="color: blue">aplicado</span>
a un elemento en línea</p>
```

La etiqueta **div** se suele acompañar de uno o dos atributos destinados a identificar ese bloque:

> **id:** permite establecer un identificador único para el bloque. Así podremos referirnos al bloque de forma inequívoca.
> **class**: es similar a id, pero con la ventaja de que se puede repetir; así que podemos tener varios **div** diferentes con la misma clase.

Usando **id** y **class** podemos diferenciar unos **div** de otros y así aplicarles estilos CSS diferentes o hacer que actúen de forma distinta.

Otro uso de div es para crear niveles dentro de section, article, header, footer o aside. Por ejemplo, si queremos tres niveles dentro de una sección podemos anidar tres artículos o también usar DIV con un CLASS para crear el tercer nivel.

Etiqueta <hgroup>

La etiqueta <hgroup> que se solía utilizar junto con el HEADER ha quedado obsoleta en HTML5.

Si bien el elemento <hgroup> se eliminó de la especificación HTML5 (W3C), todavía se mantiene en la versión WHATWG de HTML, por lo que es improbable que desaparezca.

Estándares de HTML

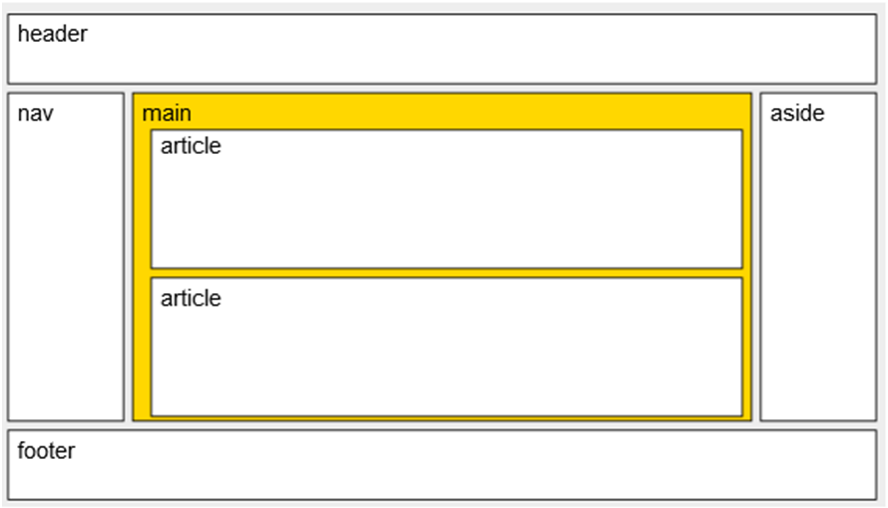
W3C

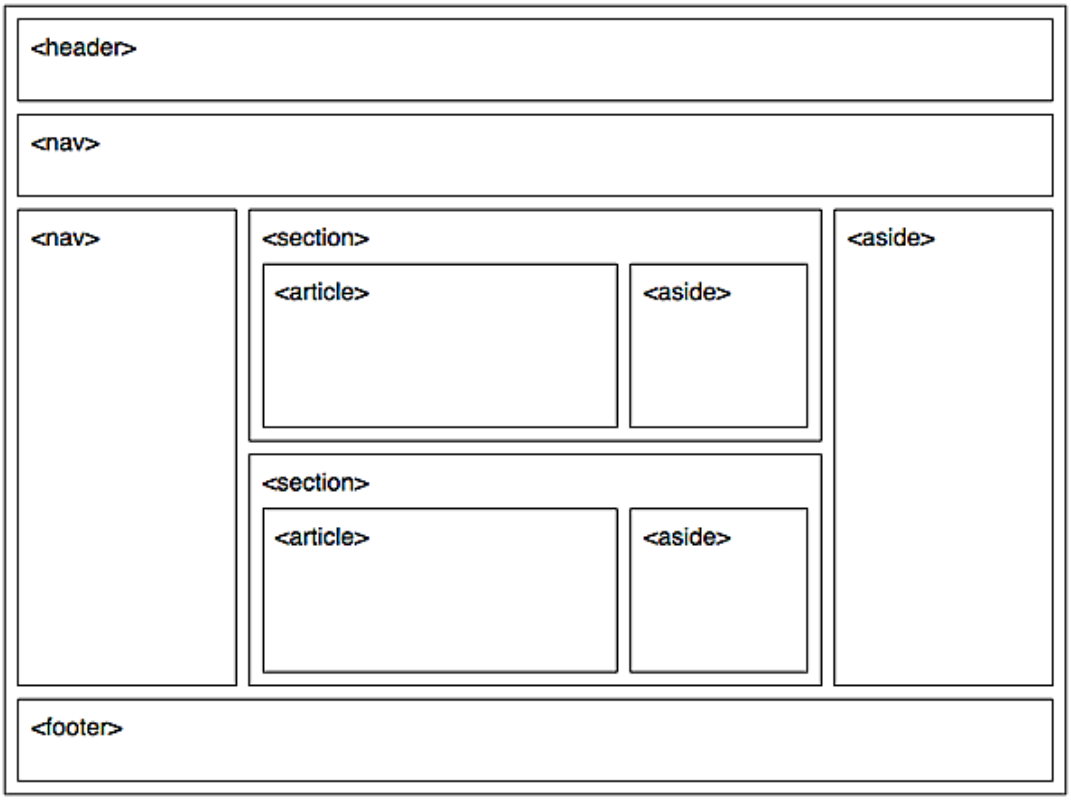
El Consorcio World Wide Web (W3C) formado en 1994, es la comunidad donde las organizaciones, los desarrolladores y el público en general trabajan en conjunto desarrollando recomendaciones y estándares que aseguran el crecimiento de la World Wide Web a largo plazo.

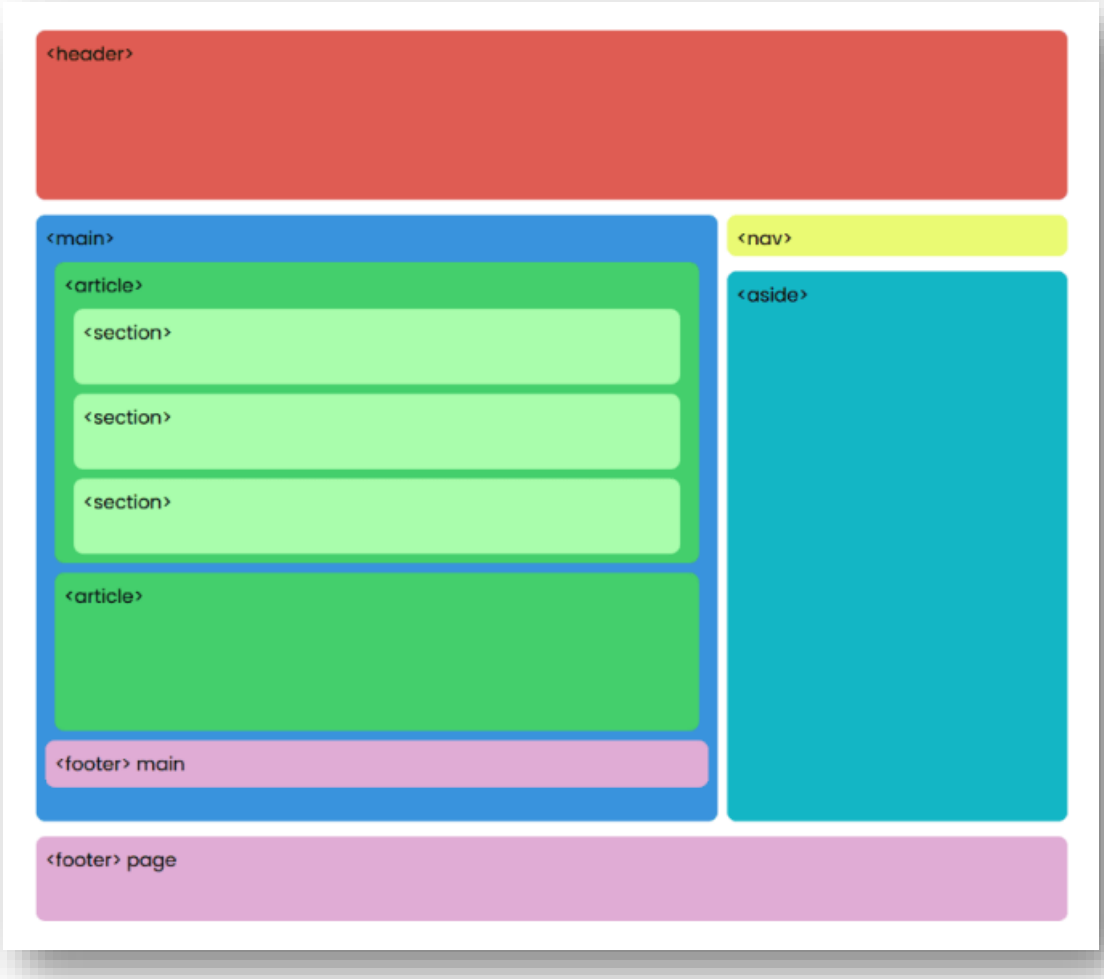
WHATWG

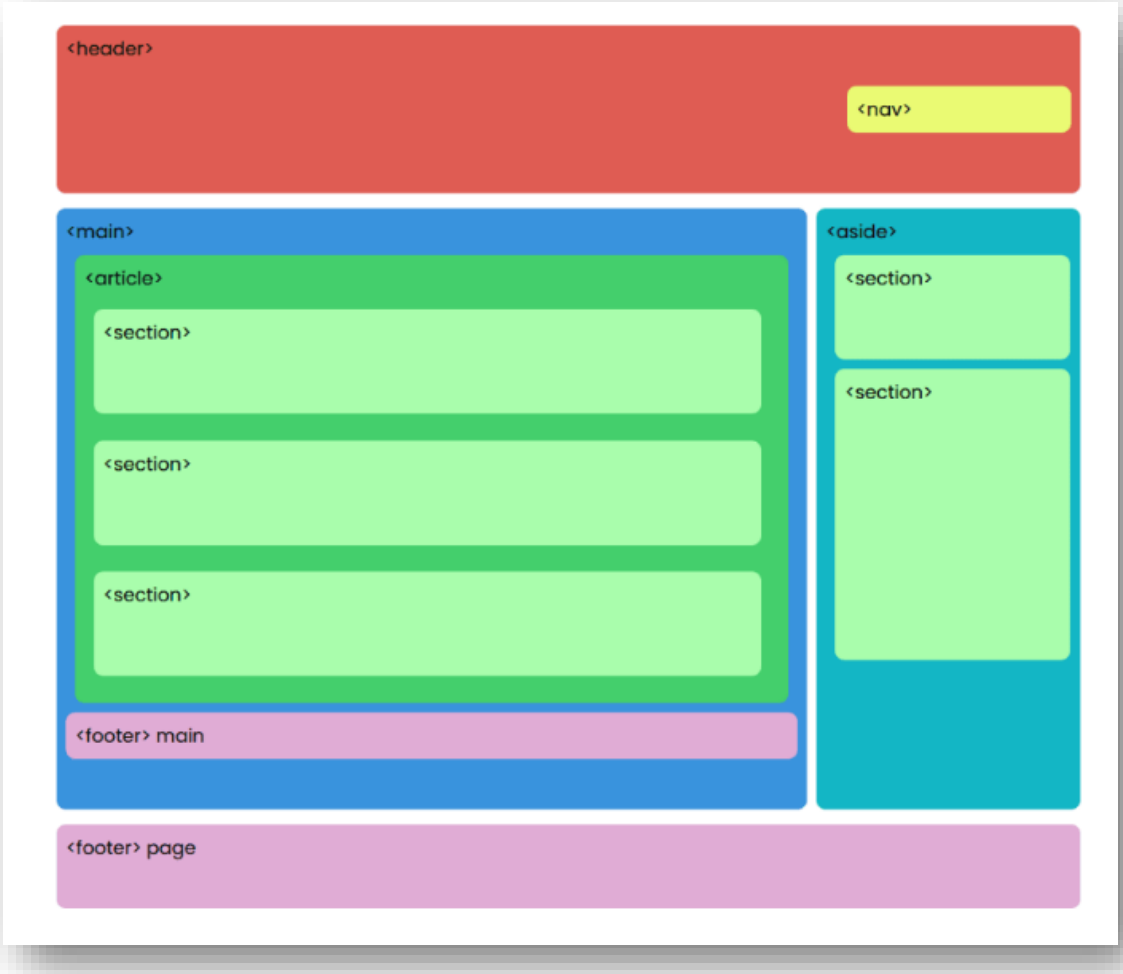
El Web Hypertext Application Technology Working Group (WHATWG) es un consorcio formado en 2004. **Constituido a raíz de un desacuerdo con W3C, después que se anunciara la decisión de pasar a centrarse en XHTML como evolución de HTML.**

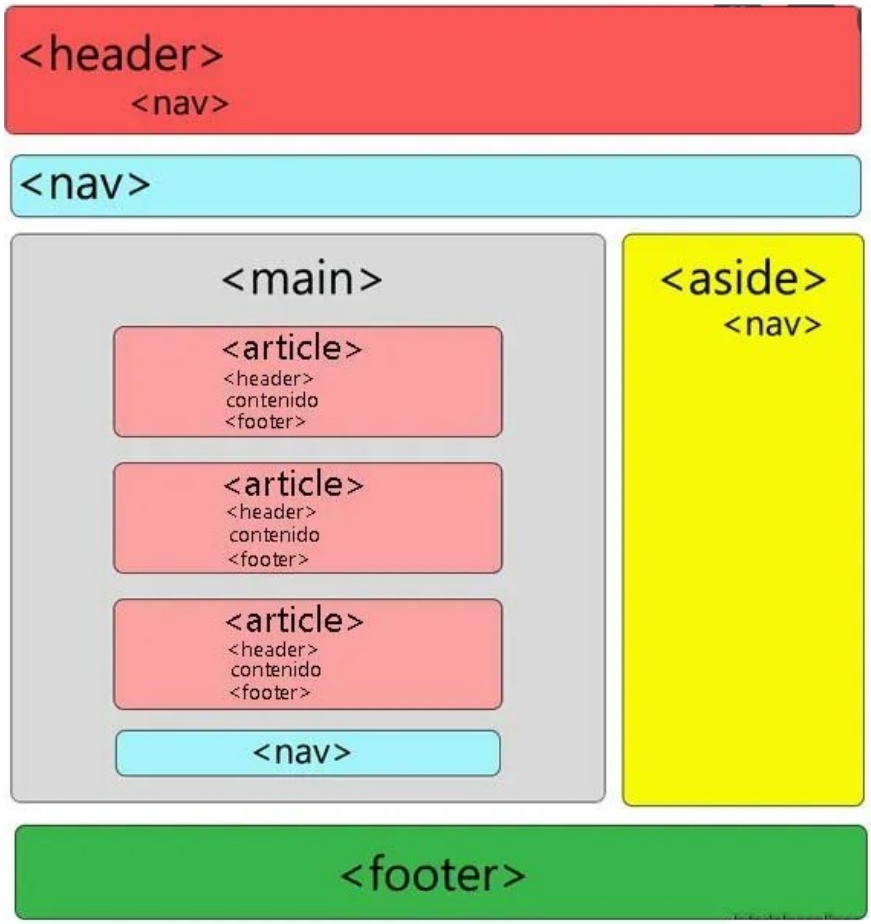
Desde entonces parece que tenemos dos rivales luchando por la estandarización web: WHATWG y W3C.

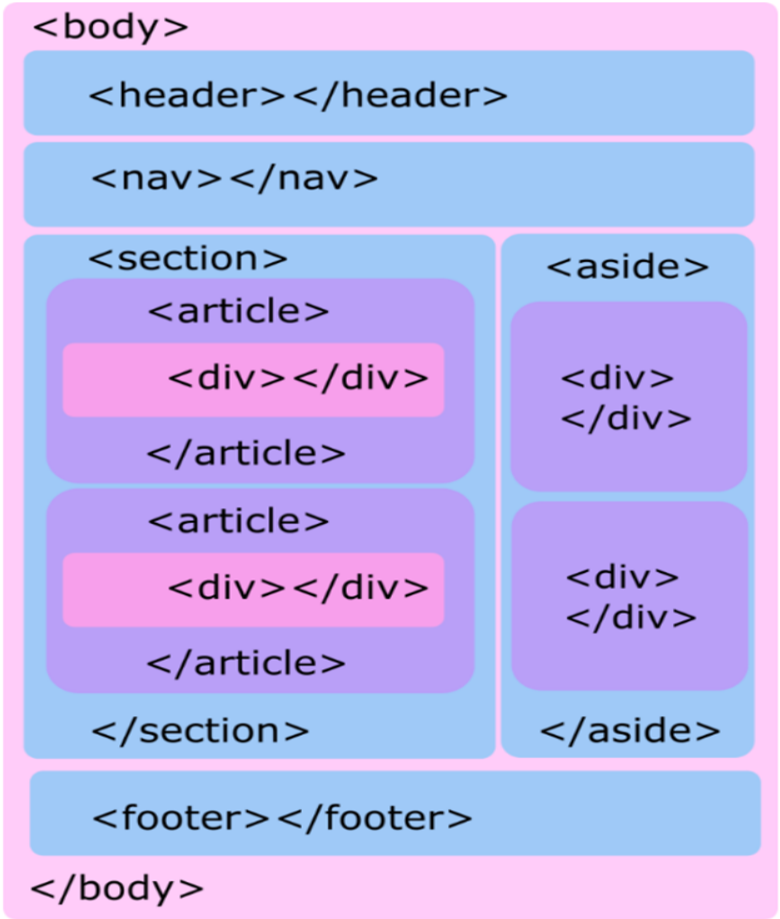

header
nav
main
article
article
aside
footer


<header>
<nav>
<nav>
<section>
<article>
<aside>
<section>
<article>
<aside>
<aside>
<footer>


<header>
<main>
<article>
<section>
<section>
<section>
<article>
<footer> main
<nav>
<aside>
<footer> page


<header>
<nav>
<main>
<article>
<section>
<section>
<section>
<footer> main
<aside>
<section>
<section>
<footer> page


<header>
<nav>
<nav>
<main>
<article>
<header>
contenido
<footer>
<article>
<header>
contenido
<footer>
<article>
<header>
contenido
<footer>
<nav>
<aside>
<nav>
<footer>

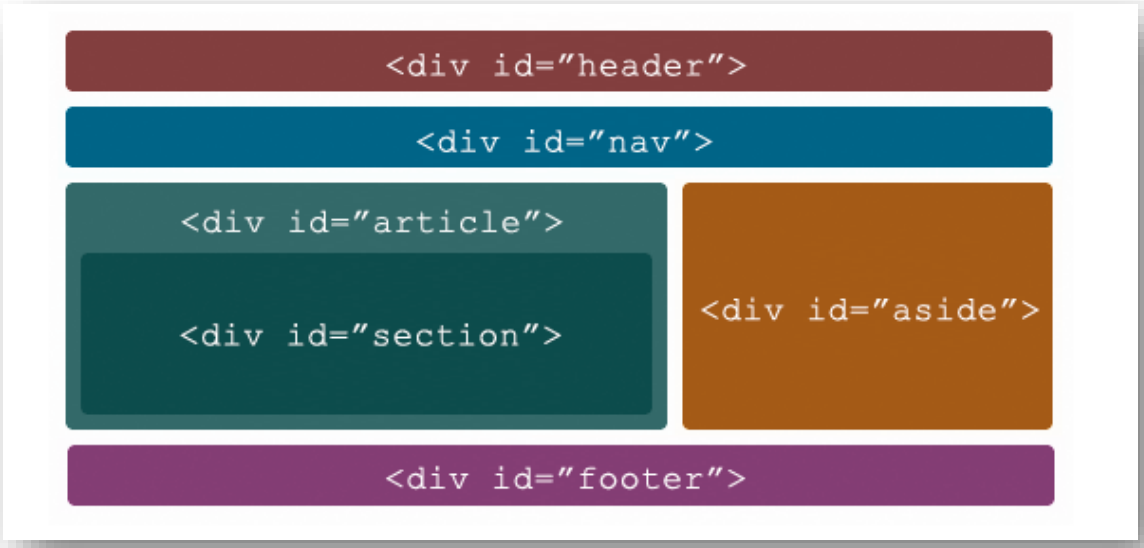

<body>
<header></header>
<nav></nav>
<section>
<article>

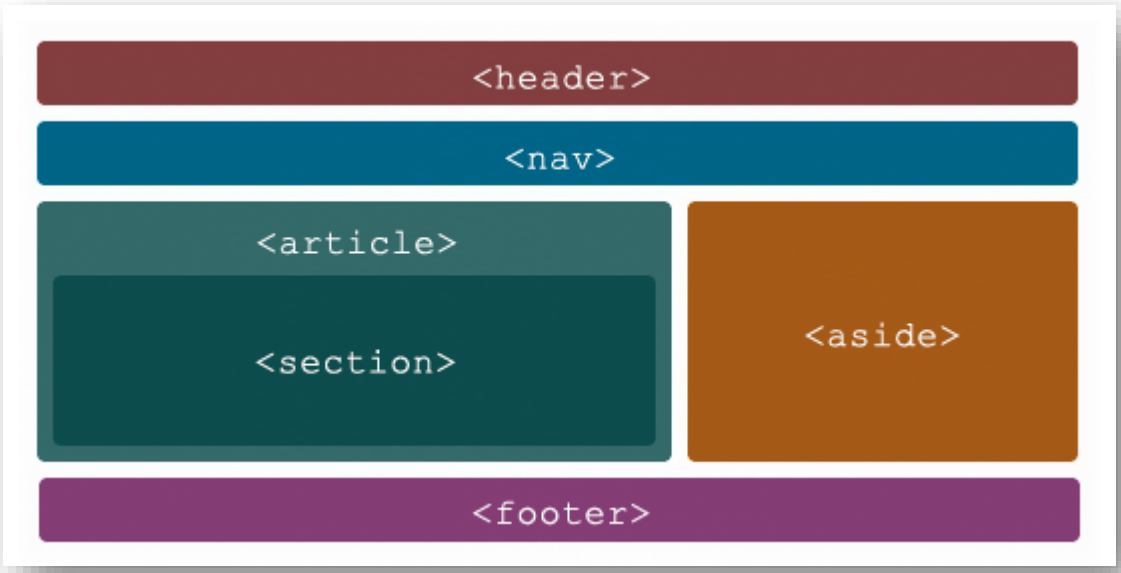
</article>
<article>

</article>
</section>
<aside>

</aside>
<footer></footer>
</body>

ESTRUCTURA HTML4 vs HTML5





ENTIDADES HTML

Una entidad HTML es un conjunto de caracteres que comienza con un *ampersand* (&) y termina con un punto y coma (;).

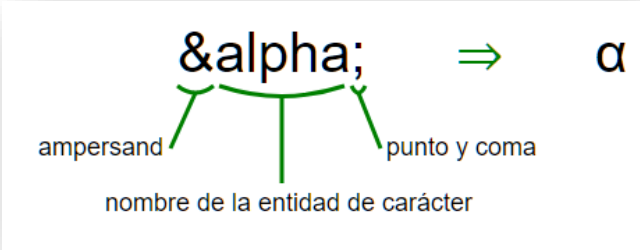
Las entidades son utilizadas para imprimir en pantalla caracteres reservados (aquellos que serían interpretados como HTML por el navegador) o **invisibles** (cómo tabulaciones). También se utilizan para representar **caracteres que no existan en los teclados**, por ejemplo, caracteres con tilde o diéresis, letras del alfabeto griego, emoticonos, …

Entidades de carácter y entidades numéricas

En las primeras versiones de HTML se especificaba que el juego de caracteres utilizado por las páginas web debía ser el juego de caracteres **ISO 8859-1**, más conocido como **Latin-1**.

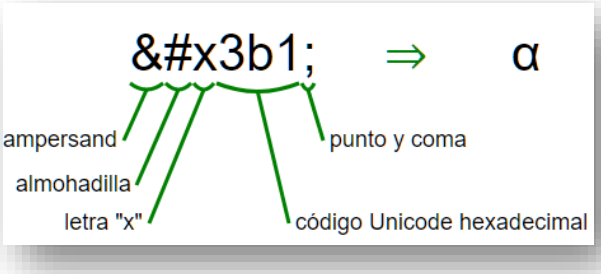
Para poder incluir en una página web caracteres no incluidos en ese juego de caracteres se debían utilizar las **entidades de carácter**. Una entidad de carácter es una referencia a un carácter y se escribe con la sintaxis **&nombre;** es decir, empieza por el símbolo & (en inglés, *ampersand*, en español, et), a continuación, se escribe el nombre del carácter y termina por un punto y coma.

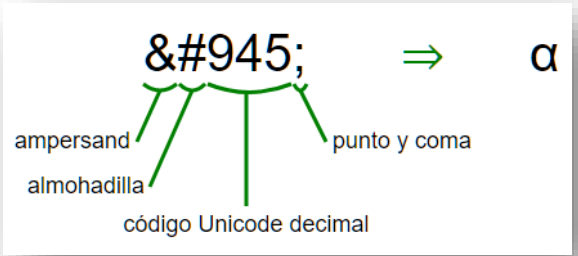
Por ejemplo, la letra griega alfa (α) se podía incluir en una página web con la entidad de carácter **α**



En enero de 1997, se permitió el uso de Unicode como juego de caracteres en HTML, de todos los juegos de caracteres de los que se compone Unicode, actualmente el más utilizado es UTF-8

Debido a la gran cantidad de carteres existentes en Unicode, se amplió el concepto de entidad de carácter a **entidad numérica**, para permitir la referencia a cualquier carácter Unicode, con la sintaxis **&número;** es decir empieza por el símbolo &, a continuación, se escribe el código Unicode del carácter (en notación decimal o en notación hexadecimal precedida por una letra "x") y termina por un punto y coma.





Caracteres Reservados

Algunos caracteres son reservados para uso en HTML, es decir, que no pueden utilizarse sin que el navegador los interprete como parte del código HTML.

Por ejemplo, al utilizar el símbolo para "menor a" (<), el navegador interpretara cualquier texto que siga como parte de una etiqueta.

Para utilizar estos caracteres como texto, deben reemplazarse por la entidad que les corresponda, debes recordar las siguientes:

Carácter	Entidad	Nota
&	&amp;	Interpretado como el comienzo de una entidad HTML.
<	&lt;	Interpretado como la apertura de una etiqueta
>	&gt;	Interpretado como el cierre de una etiqueta
"	&quot;	Interpretado como apertura o cierre del valor de un atributo
á	&aacute;	Vocal *a* acentuada
é	&eacute;	Vocal *e* acentuada
í	&iacute;	Vocal *i* acentuada
€	&euro;	Símbolo de la moneda euro
£	&pound;	Símbolo de la moneda libra
©	&copy;	Símbolo de copyright
®	&reg;	Símbolo de marca registrada
espacio		Interpretado como un espacio en blanco, alternativa

Páginas Web con Referencias de carácter y códigos UNICODE

- https://html.spec.whatwg.org/multipage/named-characters.html#named-character-references
- https://unicode-table.com/es/sets/symbols-for-youtube/#check-marks

Entidades HTML para caracteres especiales			
Símbolo	Entidad carácter	Entidad numérica	Nombre
★	★	★	Estrella
☆	☆	☆	Estrella blanca
½	½	½	Medio
♥	♥	&#9829	Corazón
♣	♣	&#9827	Picas
♠	♠	♠	Espadas
♦	♦	♦	Diamantes
∞	∞	∞	Infinito
∑	∑	∑	Sumatorio
∈	∈	∈	Pertenece
∫	∫	∫	Integral
√	√	√	Raíz cuadrada
⚜		ƒ	Flor de lis
♫		♫	Nota musical
☯		☯	Ying-Yang
🙂		🙂	Cara sonriente
😢		😢	Cara llorando
🤔		🤔	Cara pensativa
😍		😍	Cara con corazones

Entidad carácter	Entidad numérica	Nombre
★	★	Estrella
☆	☆	Estrella blanca
½	½	Medio
♥	♥	Corazón
♣	♣	Picas
♠	♠	Espadas
♦	♦	Diamantes
∞	∞	Infinito
∑	∑	Sumatorio
∈	∈	Pertenece
∫	∫	Integral
√	√	Raíz cuadrada
	♫	Nota musical
	☯	Ying-Yang
	⚜	Flor de lis
	🙂	Cara sonriente
	😢	Cara llorando
	🤔	Cara pensativa
	😍	Cara con corazones

Si los símbolos Unicode ***no se visualizan bien*** *en un sistema operativo es porque no tienen dentro de la fuente habitual que usa el sistema, dichos caracteres implementados. Por ejemplo, si tenemos texto escrito con el alfabeto chino o japonés, nuestro sistema operativo entiende que es un carácter de Unicode, pero si no están instaladas las fuentes en chino o japonés, no las podrá representar.*

Para agregar la fuente de un idioma en W10 debemos ir: ***Windows->Configuración -> Hora e idioma. -> Región e idioma -> Agregar un idioma.***

TIPOS MIME

El tipo Extensiones Multipropósito de Correo de Internet (Multipurpose Internet Mail Extensions) (MIME), es una forma estandarizada de indicar la naturaleza y el formato de un documento, archivo o conjunto de datos.

Los navegadores usan el tipo MIME (y no la extensión de archivo) para determinar cómo procesará un documento. Los tipos MIME son únicos y ésta es la principal razón para utilizar los tipos MIME siempre que sea posible, ya que una extensión puede asociarse a más de un formato.

En documentos HTML, los desarrolladores usan los tipos MIME en muchas etiquetas, a través del atributo "type". Algunos casos especiales de uso son los atributos "enctype" del elemento HTML <form>.

Sintaxis

La estructura de un tipo MIME es muy simple; consiste en un tipo y un subtipo, dos cadenas, separadas por un '/'. No se permite espacio. El tipo representa la categoría y el subtipo es específico para cada tipo.

Para documentos de texto sin subtipo específico, se debe usar **text/plain**. De forma similar, para los documentos binarios(aplicaciones) sin subtipo específico o conocido, se debe usar **application/octet-stream**.

Los tipos que indican la categoría del documento, puede ser uno de los siguientes:

Tipo	Descripción	Ejemplo de subtipos
text	Representa cualquier documento que contenga texto y es teóricamente legible por humanos	`text/plain`, `text/html`, `text/css`, `text/javascript`
image	Representa cualquier tipo de imagen. Los videos no están incluidos, aunque las imágenes animadas (como el gif animado) se describen con un tipo de imagen.	`image/gif`, `image/png`, `image/jpeg`, `image/bmp`, `image/webp`
audio	Representa cualquier tipo de archivos de audio	`audio/midi`, `audio/mpeg`, `audio/webm`, `audio/ogg`, `audio/wav`
video	Representa cualquier tipo de archivos de video	`video/webm`, `video/ogg`
application	Representa cualquier tipo de datos binarios.	`application/octet-stream`, `application/vnd.mspowerpoint`, `application/xhtml+xml`, `application/xml`, `application/pdf`

ETIQUETAS **HTML** QUE ADMITEN EL ATRIBUTO **TYPE** CON **MIME**

<base>: Define una URL base para los enlaces de la página y puede indicar el tipo MIME de los recursos enlazados.

```
<base href="https://www.ejemplo.com/" type="text/html">
```

<script>: Define scripts y puede tener el atributo **type** para especificar el tipo MIME del script.

```
<script type="text/javascript" src="script.js"></script>
```

<link>: Vincula recursos externos y puede especificar el tipo MIME del recurso.

```
<link rel="stylesheet" href="estilos.css" type="text/css">
<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
```

<style>: Define estilos en línea o internos al documento. Aunque el atributo type es opcional en HTML5 para **<style>,** se usaba históricamente para indicar el tipo de estilo (generalmente text/css).

```
<style type="text/css">
    /* Estilos CSS aquí */
</style>
```

<a>: Permite especificar el tipo MIME del recurso al que enlaza.

```
<a href="archivo.pdf" type="application/pdf">Enlace a archivo PDF</a>
```

<area>: Se utiliza en conjunto con la etiqueta **<map>** para definir áreas sensibles en imágenes y puede especificar el tipo MIME del recurso.

```
<map name="mapa">
  <area shape="rect" coords="0,0,20,20" href="pagina.html" type="text/html">
</map>
```

<object>: Se utiliza para incrustar objetos multimedia como imágenes, videos, archivos Flash, etc. El atributo **type** define el tipo MIME del recurso incrustado.

```
<object data="video.mp4" type="video/mp4" width="400" height="300">
    Tu navegador no soporta el elemento de video.
</object>
```

<source>: Se utiliza dentro de elementos multimedia como **<video>** o **<audio>** para proporcionar diferentes fuentes de medios según el tipo MIME especificado.

```
<video controls>
        <source src="video.mp4" type="video/mp4">
        <source src="video.webm" type="video/webm">
        Tu navegador no soporta el elemento de video.
</video>
```

<track>: Utilizado dentro de elementos multimedia como **<video>** o **<audio>** para proporcionar pistas de texto como subtítulos o descripciones de audio y puede definir el tipo MIME del recurso de pista.

```
<video controls>
        <source src="video.mp4" type="video/mp4">
        <track src="subtitulos.vtt"
               kind="subtitles"
               srclang="es"
               label="Subtítulos en español"
               type="text/vtt">
</video>
```

<embed>: Se utiliza para incrustar varios tipos de contenido, y el atributo **type** define el tipo MIME del recurso incrustado.

```
 <embed src="archivo.mp3" type="audio/mpeg">
```