



HTML5

M04 - Lenguaje de Marcas

Alicia Vázquez



Un poco de historia

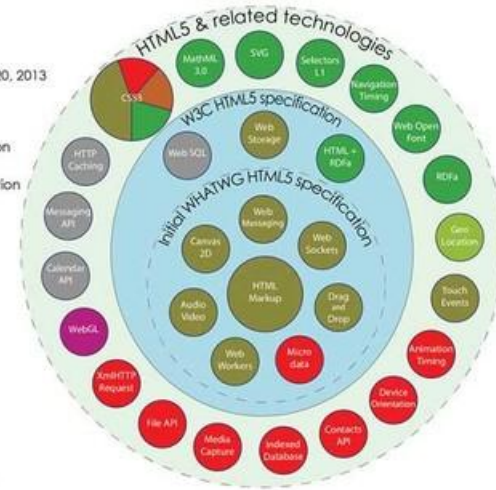
- HTML se remonta a 1980 Tim Berners-Lee
- La primera descripción de HTML se publicó en **1991**
- 1993 primer standard [IETF](#) (*Internet Engineering Task Force*)
- **1995** HTML 2.0 standard oficial
- 1997 HTML 3.2 y es la primera recomendación de HTML publicada por el **W3Consortium**.
- **1998** HTML 4.0
- W3C se centró en el desarrollo del estándar XHTML, que no tuvo el éxito esperado.
- 2004 WHATWG retoma la iniciativa y define estándar **HTML 5**

Pero qué es HTML5 ? [Definición](#)

HTML5

Taxonomy & Status on January 20, 2013

- W3C Recommendation
- Proposed Recommendation
- Candidate Recommendation
- Last Call
- Working Draft
- Non-W3C Specifications
- Deprecated



by Sergey Mavrodov CC BY-SA

Html5 (infografía)



HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML.

Nació en 2004, cuando se fundó el grupo de trabajo WHAT con miembros de **Apple**, la **Fundación Mozilla** y **Opera Software**. Dos años después consiguieron uno de sus principales apoyos, pues el **W3C** (World Wide Web Consortium) decidió abandonar XHTML y comenzó a trabajar con WHAT.

En 2008 se finalizó su primera versión. Esto permitió que **Firefox 3** y **Mozilla**, fuese compatible con **HTML5**. Más adelante se sumarían **Internet Explorer**, **Google Chrome** y **Safari**.

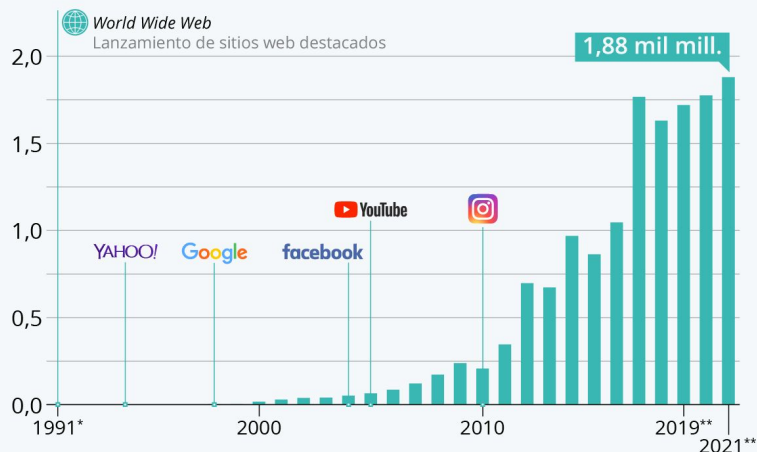
Poco a poco se fueron sumando compañías y se fue abandonando el uso de tecnologías como **Flash**. De hecho, en septiembre de **2011**, 34 de las 100 páginas con más tráfico según el ranking de Alexa utilizaban este lenguaje.

HTML5 ina es un lenguaje estable desde **2013**.

Los datos

¿Cuántas páginas web existen?

Número de sitios web existentes en Internet (en miles de mill.)



Un sitio web se entiende aquí como un "hostname" único.

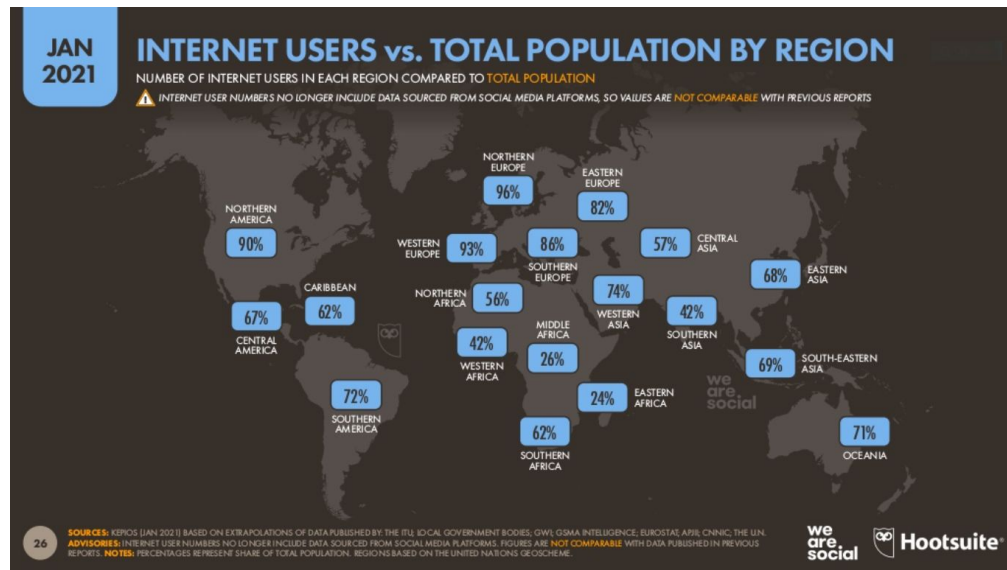
* Datos del 1 de agosto de 1991

** Últimos datos disponibles 2019: 28 de octubre, 2020: 2 de junio, 2021: 6 de agosto.

Fuente: Internet Live Stats



statista



Modem a 56K en 1994


Html5

HTML no es un lenguaje de programación, es un lenguaje de marcas. Sirve para **estructurar** la información y permite que el contenido sea fácilmente analizable.

Define el **QUÉ** y **DÓNDE**

HTML es un **XML** pero con una sintaxis propia.

CSS



CSS (Cascading Style Sheets), en español "Hojas de estilo en cascada", es un **lenguaje de diseño gráfico** para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado.

Es muy usado para establecer el diseño visual de los documentos web, e interfaces de usuario escritas en HTML o XHTML; el lenguaje puede ser aplicado a cualquier documento XML, incluyendo XHTML, SVG, XUL, RSS, etcétera. También permite aplicar estilos no visuales, como las hojas de estilo auditivas.

CSS está diseñado principalmente para marcar la separación del contenido del documento y la forma de presentación de este, características tales como las capas o layouts, los colores y las fuentes.

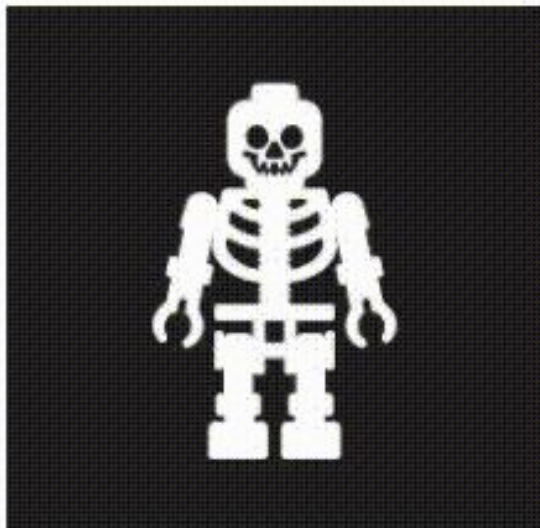
CSS3



CSS no es un lenguaje de **programación**. Sirve para dar **formato** al contenido estructurado con un lenguaje de marcas.

Define el **CÓMO**

HTML
structure



CSS
presentation/appearance



JavaScript
dynamism/action





HTML5

Sintaxis de HTML

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



Ejemplo:

```
<p id="primer" class="grande">
```

Hola Mundo!

```
</p>
```

class → hará referencia a un estilo definido en el CSS.

id → es el identificador del elemento en el DOM (Document Object Model)

Estructura básica

A diagram illustrating the basic structure of an HTML document. It consists of several nested rectangular boxes. The outermost box is light gray and contains the text '<html>' at the top left and '</html>' at the bottom left. Inside this box is a blue box containing '<!DOCTYPE html>'. Below the blue box is an orange box containing '<head>' and '</head>'. Below the orange box is a dark blue box containing '<body>' and '</body>'.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>  
</head>
```

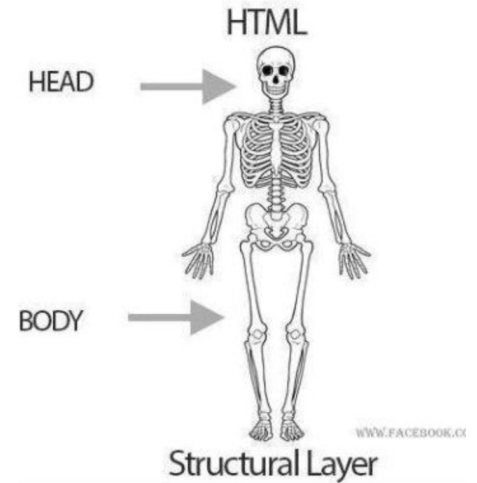
```
<body>  
</body>
```

```
</html>
```


HTML

Estructura

- Head
- Body



Estructura HTML (head)



```
<!DOCTYPE html>
<html lang="en">
  <head> <!-- información de la página y será lo primero que se cargue-->
    <meta charset="UTF-8">
    <title>Document</title>
    <!--
    enlazar archivos css
    enlazar archivos js
    añadir fuentes externas
    metadatos (keywords, descriptions, robots, author, owner)
    -->
  </head>
  <body> <!-- contenido de la página -->
  </body>
</html>
```

head ([html-head](#))

La etiqueta **<head>** contiene información que se precarga en el navegador y **NO** se visualiza. Por lo que nunca encontraremos contenido propio de la página sino información necesaria para su renderización (javascript, css, metadatos u otros).

- ★ **<title>** es donde indicaremos el nombre de la página que queremos que salga en la pestaña del navegador.
- ★ **<link>** es la etiqueta que nos permite “incluir” diversos ficheros entre ellos el icono de la página.

```
<link rel="icon" type="image/gif" href="favicon.gif">
```

Pero también usaremos **<link>** o **<script>** para hacer referencia a ficheros css y js que use nuestra web.

```
<script src="[file_name].js"></script>
```

```
<link rel="stylesheet" type="text/css" href="[file_name].css">
```

head

- ★ **<meta>** es la etiqueta que nos permite definir metadatos de nuestra página; por ejemplo el **autor**, una **descripción** para navegadores, etc.

```
<meta charset="utf-8"/> <!-- para que reconozca las ñ's-->
```

```
<meta name="author" content="Chris Mills">
```

```
<meta name="description" content="The MDN Learning Area aims to provide complete beginners to the Web with all they need to know to get started with developing web sites and applications.">
```

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



HTML

Estructura

- ❏ Head

- ❏ Body

- ❏ header

- ❏ main y div

- ❏ section y article

- ❏ aside

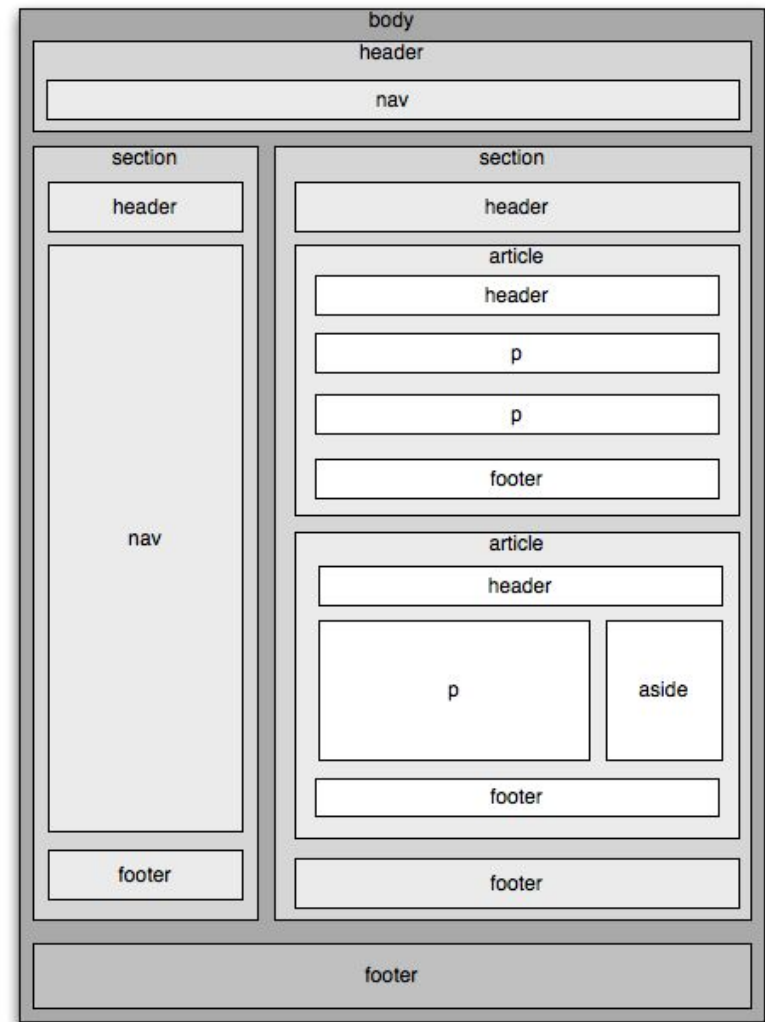
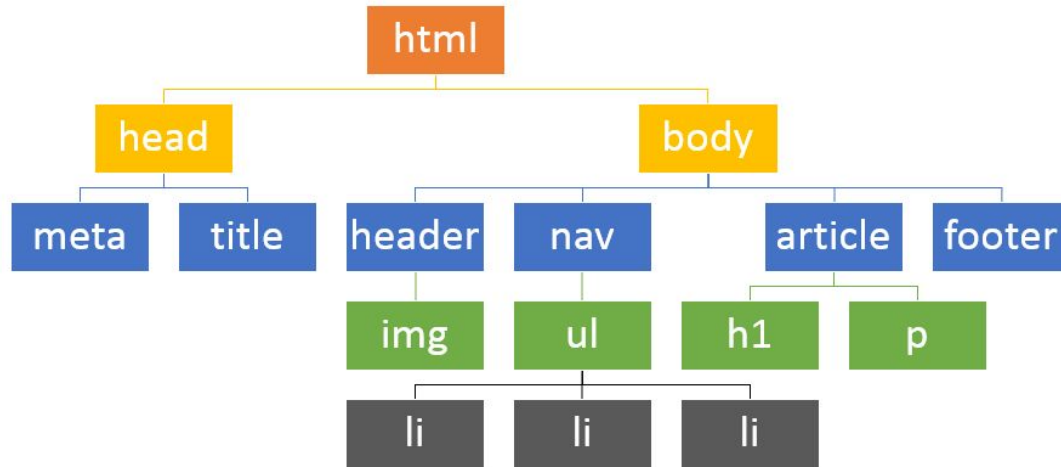
- ❏ footer

body ([tag-body](#))

El **body** es el elemento principal, el **contenedor** de toda la información que será renderizada en el navegador y por lo tanto aquí es dónde irá todo nuestro **contenido**, la información a mostrar.



Estructura



Estructura HTML (body)

```
<!DOCTYPE html>
<html lang="en">
<head><title></title></head>
<body>
  <header><!--Logo, título principal, menús de navegación, etc...-->
    <h1>Título principal </h1>
    <h2>Subtítulo</h2>
    <nav>Menú navegación</nav>
  </header>
  <main><!-- Pondremos todo el contenido -->
    <section><!--Sección del contenido-->
      <article><!--artículo dentro de la sección-->
        <h3>Título del artículo</h3>
        <p>Contenido del artículo</p>
      </article>
    </section>
  </main>
  <footer>Document</footer>
</body>
</html>
```

header (tag-header)

El **header** no confundir con head, es el elemento que engloba a todos los elementos que pertenecen a la cabecera de nuestra página.

En el encontraremos el **título** de la página (`h1` y `h2`) el **logo** (en una `img`), el **menú de navegación** (`nav`) y quizás **un buscador**, enlaces a **rss** usando **iconos** y/o cambios iconos que nos permiten cambiar de idioma nuestra web.

Son el conjunto de los múltiples elementos que nos permitirán navegar por nuestra web e identificarla, muchas veces será este elemento el que queremos que esté siempre visible.

h1, h2, h3, etc (tag-hn)

Cada **h1, h2, h3,..., h6** representa un título dentro de nuestra página que será la manera de organizar nuestra página.

SEO - Posicionamiento.

- **H1**, debe estar dentro del header puesto que es lo primero que analiza Google para posicionar nuestra web. Solo deberíamos tener un solo **H1**.
- De la misma manera, cuanto más títulos del resto pongamos más se diluye su importancia a la hora de posicionarse.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Ejemplo de h1</title>
6  </head>
7  <body>
8      <header>
9          <h1>Título principal</h1>
10     </header>
11
12     <h2>Título secundario 1</h2>
13     <h2>Título secundario 2</h2>
14     <h2>Título secundario 3</h2>
15     <h3>Título h3</h3>
16     <h4>Título h4</h4>
17     <h5>Título h5</h5>
18 </body>
19 </html>
```

Ejercicio 1 - Hello World

Crea tu primera página html, **index.html** usando VSC, instalate els plugin de HTML5.

- Ponle un título.
- Un autor
- Una descripción
- Añadir un icono ([favicon](#))
- Y que salga en el body lo siguiente:

Hello World

We are here

Empezamos a trabajar HTML

nav (tag-nav)

La etiqueta **nav** es la que indica que hay un **menú de navegación** y agrupa el resto de elementos que permiten movernos dentro de la página o enlazar a otras páginas del site.

Dentro de **nav** podemos encontrar:

- **ul**: Lo usamos para definir una lista de elementos, desordenada [unordered] (puntos).
- **ol**: Define una lista ordenada [ordered] de elementos (números)
 - **li**: Representa un elemento de la lista.
 - **a**: Enlace (anchor).

- Opcion1
- Opcion2
- Opcion3
- Opcion4

1. Opcion1
2. Opcion2
3. Opcion3
4. Opcion4

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Menú</title>
6  </head>
7  <body>
8      <header>
9          <nav>
10             <ul>
11                 <li>Opcion1</li>
12                 <li>Opcion2</li>
13                 <li>Opcion3</li>
14                 <li>Opcion4</li>
15             </ul>
16         </nav>
17         <nav>
18             <ol>
19                 <li>Opcion1</li>
20                 <li>Opcion2</li>
21                 <li>Opcion3</li>
22                 <li>Opcion4</li>
23             </ol>
24         </nav>
25     </header>
26 </body>
27 </html>
```

anchor (tag-header)

Dentro de un elemento de la lista (**li**) podemos definir el enlace que tenga, para ello usaremos:

- **a**: Todo su contenido será el que nos envíe al enlace definido.
 - **href**: Atributo en el que indicaremos la dirección del enlace (url, directorio, identificador).
 - **target**: Si se abre en una nueva pestaña o no, por defecto se abrirá en la pestaña actual. En caso de que queramos que se abra en una nueva pestaña deberemos poner el valor **_blank**

NOTA: **<a>** debe estar siempre dentro del elemento de la lista, no al revés. En ambos casos funciona, pero no es correcto.

```
<a href="#" target="_blank"><li>enlace</li></a>
```



```
<li><a href="#" target="_blank">enlace</a></li>
```




```
<body>
  <header>
    <nav>
      <ul>
        <li>
          <a href="http://google.com">Vamos a otra página url</a>
        </li>
        <li>
          <a target="_blank" href="http://google.com">Abrimos la otra página en otra pesataña</a>
        </li>
        <li>
          <a href="#ejemplo">Navegamos dentro de nuestra propia página y vamos a una zona especifica</a>
        </li>
        <li>
          <a href="../ejemplos/12 - hgroup.html">Navegamos dentro de nuestro directorio</a>
        </li>
      </ul>
    </nav>
  </header>
  <section>
    <h4>Esta zona no quiero que vengas</h4>
    <p>
      Lorem Ipsum
    </p>
  </section>
  <section id="ejemplo">
    <p>Esta es la zona que quiero que llegues desde el menú</p>
  </section>
</body>
```

Ejercicio 2

A la página del ejercicio anterior añade un menú de navegación, una lista de elementos con enlaces que permitirán en un futuro añadir más páginas.

- Menú de navegación con mínimo 4 enlaces (externo, interno con referencia a propia página)
- 2 secciones con 1/2/3 artículos. Cada artículo con título y texto.

Hello World!

Mi primera página en HTML

Vamos a parender muchas cosas

- [Opción 1](#)
- [Opción 2](#)
- [Opción 3](#)
- [Opción 4](#)

sección 1

artículo 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Iure cumque quam nostrum vitae ad, at quis dicta perspiciatis necessitatibus voluptatibus sapiente unde ipsum libero corporis, perferendis sint delectus hic? Ea.

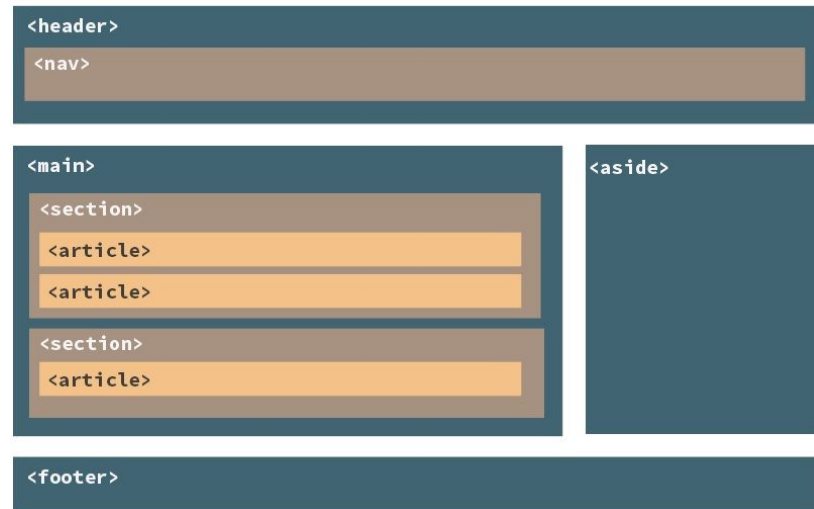
Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, labore reprehenderit nam dolorum! Ipsam vitae animi ab, atque illum repellendus vel.

artículo 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Iure cumque quam

main y div ([tag-header](#))

- ★ **<main></main>** es dónde está la **información principal** del cuerpo de nuestra página. Solo puede ser hijo de **body** y solo puede aparecer una vez. No lo podremos encontrar en ningún otro lugar pero dentro definiremos una amplia estructura de contenido.
- ★ **<div></div>** es la etiqueta más empleada puesto que permite agrupar elementos, esto nos permite organizarnos. Todos los elementos que pueden ser agrupados que vayan juntos, los meteremos en un **div**. Cuidado de no abusar puesto que existen tags específicos par organizar la información.



article (tag-header), section (tag-header)

- ★ Aunque **<article></article>** también engloba muchos elementos diferentes que juntos dan un contenido, que forman un solo elemento, un post.

→ article, engloba elementos, que todos ellos representan un elemento de información, sueltos, no tienen sentido.

- ★ La etiqueta **<section></section>** permite **englobar** un conjunto de contenido que tienen relación entre sí. Podemos tener dentro de una sección, **<section>**, muchos otros elementos, lo importante es que todos están relacionados entre ellos a nivel de **temática**.

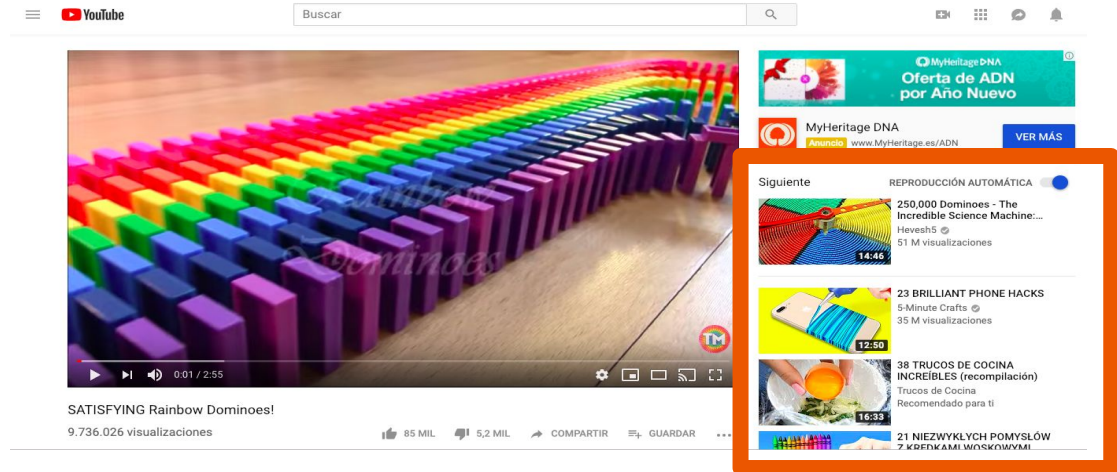
→ section une elementos a nivel conceptual, elementos que tratan el mismo contenido, pero sueltos tienen sentido.



aside (tag-header)

El `<aside>``</aside>` contiene los elementos que no son parte del contenido principal, pero que sí están relacionado con él. Pondremos dentro de un `<aside>`, sugerencias, contenido que puede interesar y enlaces externos.

Para los buscadores, el contenido que esté dentro de `<aside>`, será menos relevante.



p (tag-p)

- ★ La etiqueta `<p></p>` de **párrafo** es esencial para introducir contenido dentro de la página. Toda información de texto deberá estar contenida en este elemento. Cada vez que cerremos la etiqueta se creará automáticamente un salto de línea.
- ★ para poder añadir voluntariamente un salto de línea podemos usar el elemento `
` que no se cierra y es elemento **empty**.

```
<article>
  <h1>Mi primera página</h1>
  <!-- Aquí va un comentario que no es interpretado por el navegador →
    <p>Hola mundo, esta es un página con titular,
      que tiene también un párrafo y <br>unos cuantos saltos de línea.</p>
</article>
```




Mi primera página

Hola mundo, esta es un página con titular, que tiene también un párrafo y unos cuantos saltos de línea.



HTML

Estructura

-  Head
-  Body
-  Footer

footer

`<footer></footer>` es la etiqueta que indica el pie de página, la encontraremos por lo tanto al final del código.

Normalmente nos encontraremos un solo footer, aunque en páginas muy grandes podemos encontrarnos alguno más, para acabar una sección, pero no es lo habitual y no debe sorprendernos.

`<adress></adress>` es una etiqueta usal del **footer** en la cual encontraremos la dirección postal de la entidad de la página.



SI VOLS SEGUIR LES NOVETATS

ENVIA

Copyright © 2019 Institut Jaume Balmes Design by SimpleWpThemes & FPJaumeBalmes


Totes les fotos i continguts d'aquest web estan fets per estudiants, professorat i personal de la secretaria de l'Institut Jaume Balmes

Batx · ESO · FP Informàtica - C/ Pau Claris 121 Barcelona 08009 - 93.487.03.01 - secretaria@jaumbalmes.net

Validar HTML

Para asegurarnos de que nuestro código está correctamente escrito, podemos utilizar un **Validador HTML**, que no es más que un sistema que analiza nuestro código y nos dice el número de errores que tenemos, junto a una breve descripción del mismo.

Validador

**Markup Validation Service**
Check the markup (HTML, XHTML, ...) of Web documents

Validate by **URI**

Validate by File Upload

Validate by Direct Input

Validate by URI

Validate a document online:

Address:

[▶ More Options](#)

Check

Ejercicio 3

1. Sigue con el mismo ejercicio anterior y siguiendo la estructura vista (html, head, body, main, section, article, p, footer, address) añade al fichero html:
 - Main dónde esten las 2 secciones con sus artículos.
 - Aside: zona para publicidad con dos artículos y enlaces externos.
 - Un pie de página con información del site y la dirección de contacto.
2. Haz el árbol de la estructura

Hello World!

Mi primera página en HTML

Vamos a aprender muchas cosas

- [Opción 1](#)
- [Sección 2 - Artículo 3](#)
- [Aside](#)
- [Footer](#)

Main

sección 1

artículo 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, lab

artículo 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, lab

sección 2

artículo 1

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, lab

artículo 2

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, lab

artículo 3

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, lab

Publicidad - Aside

Artículo de Publicidad

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

[Youtube](#)

Artículo de Publicidad

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Lure cumque quam nostrum vitae ad, at quis dicta perspicatis necessitatibus voluptatibus s

Lorem ipsum, dolor sit amet consectetur adipisicing elit. Similique nemo quasi voluptates autem repellendus fugiat inventore fuga voluptatibus, lab

[Amazon](#)

Footer

Política de privacidad

Mapa de la web






Contacto

Institut tecnològic de Barcelona
info@itb.cat
c/ aiguablava 121, 08033 Barcelona
Tlf: 937 070 020



HTML

Elementos específicos

-  Énfasis
-  Imágenes
-  Audio
-  Vídeo
-  Tablas

span, strong, em ([text-formatting](#))

Etiquetas para **enfatar** el contenido:

- ★ `` nos permite hacer más énfasis a ese texto, darle más fuerza semántica, destacarlo. Visualmente se verá en negrita (o no), pero este no es su cometido, en CSS podríamos cambiarlo. Su propósito es destacar un texto frente a los demás para que los buscadores puedan encontrarlo.
- ★ `` es una equitea que igual que `` permite dar más relevancia a la parte del texto que contiene, pero esta vez su formato será en itálico, que podremos cambiar con nuestro CSS.
- ★ La etiqueta `` que no hace saltos de línea, visualmente en el navegador no hay ningún cambio, no es un bloque independiente. Lo usamos para poder cambiar el estilo diferente en el CSS.

```
<p>Este es un parafo y quiero destacar con <strong>strong este trozo de texto</strong>  
y <em>quiero destacar este otro</em>, puesto que son muy importantes.  
Cuando sepamos CSS, con span podremos cambiar el estilo de esta última frase.</p>
```

Este es un parafo y quiero destacar con strong este trozo de texto y *quiero destacar este otro*, puesto que son muy importantes. Cuando sepamos CSS, con span podremos cambiar el estilo de esta última frase.

etiquetas para formatear ([text-formatting](#))

Son aquellas etiquetas que no dan valor (semántico) al contenido sino que simplemente las usamos para cambiar la manera **visual** del texto en cuestión.

```
<b> - Bold text</b>
<strong> - Important text</strong>
<i> - Italic text</i>
<em> - Emphasized text</em>
<mark> - Marked text</mark>
<small> - Smaller text</small>
<del> - Deleted text</del>
<ins> - Inserted text</ins>
<sub> - Subscript text</sub>
<sup> - Superscript text</sup>
```

- **Bold text**
- **Important text**
- *Italic text*
- *Emphasized text*
- **Marked text**
- Smaller text
- ~~Deleted text~~
- Inserted text
- Subscript text
- Superscript text

figure, figcaption, img

`<figure></figure>` Es la etiqueta específica para poner una imagen dentro de nuestro html.

Usaremos la etiqueta `` y usaremos los atributos:

- `src` (source) para indicar la ruta **relativa** de la imagen.
- `alt*` para indicar el texto informativo que aparece cuando mouse hover.

Añadiremos la etiqueta `<figcaption></figcaption>` para poner el texto que acompaña a la imagen, su leyenda.

*El atributo `alt` debe estar siempre informado, ya que los buscadores lo usan para entender que contenido tiene la imagen.

Nota: Para poner imagenes random (<https://source.unsplash.com/1600x900/?nature,water>)

audio



La etiqueta `<audio></audio>` es la que nos permite añadir un audio dentro de nuestra página.

Además deberemos informar los siguientes atributos:

- **src** (source) para indicar la ruta **relativa** del archivo.
- **type** para indicar de qué tipo de archivo se trata (audio/extensión: *audio/wav*, *audio/mp3*, etc)
- **controls** para activar los controles de reproducción de audio (play, stop, etc).

video

La etiqueta `<video></video>` es la que nos permite añadir un vídeo dentro de nuestra página. Además deberemos informar los siguientes atributos:

- `src` (source) para indicar la ruta **relativa** del archivo.
- `type` para indicar de qué tipo de archivo se trata (video/extensión: *video/mov*, *video/mp4*, etc)
- `controls` para activar los controles de reproducción de audio (play, stop, etc).

NOTA: Sin embargo, para evitar problemas de reproducción según el navegador del cliente y de peso de archivo, lo mejor es subirlo a youtube (el código nos lo da youtube):

Usar `<iframe></iframe>` para añadir un enlace al vídeo en youtube:

- `width, height` para marcar la anchura y altura del reproductor, en píxels.
- `src` (source) para indicar la ruta del archivo, en este caso una **url**
- `frameborder="0"` para activar los controles de reproducción de audio (play, stop, etc).
- `allowfullscreen` permite reproducir el video a pantalla completa (no hay que darle valor, solo añadir esta propiedad)..

audio, video (atributos)

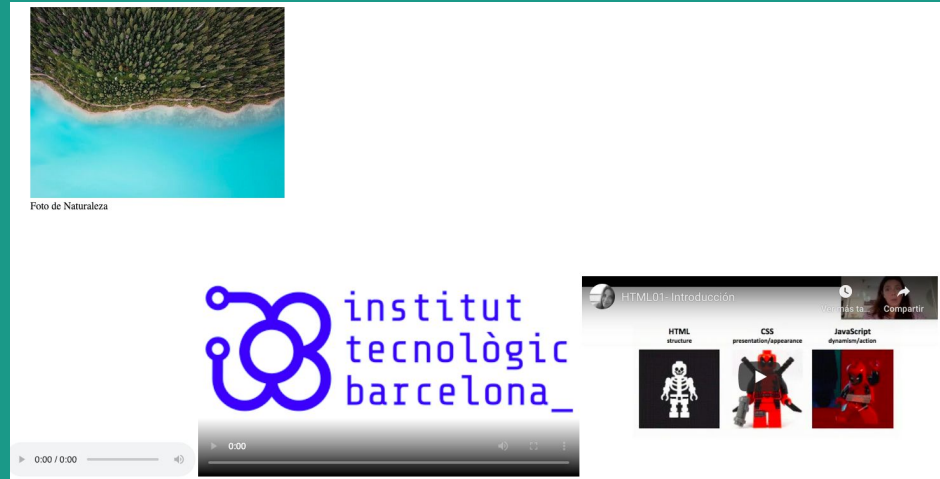
Las etiquetas **<audio>** y **<video>** pueden tener los siguientes atributos (propiedades).

- **controls** Atributo booleano que activa un conjunto de controles de reproducción integrados. (pausa, buscar y ajustar volumen).
- **poster** Cadena que representa una imagen de marcador de posición y que se muestra en el reproductor de vídeo. Esta imagen sólo se muestra cuando un vídeo no está disponible.
- **loop** Atributo booleano que reproduce el vídeo repetidamente hasta que se presiona el botón de pausa en los controles.
- **muted** Atributo booleano que reproduce vídeo con la pista de audio desactivada.
- **autoplay** Atributo booleano que inicia la reproducción de vídeo automáticamente cuando el reproductor tiene suficiente contenido almacenado en búfer.
- **preload** Atributo booleano que define una clave sobre cuánto almacenamiento en búfer es necesario.

Ejercicio 4

Crea una nueva página **Ej04.html** que contenga una imagen, un audio y dos vídeos.

- La imagen del video sea el logo del ITB (busca como hacerlo)
- Insertar un vídeo de youtube (desde youtube está la opción de <iframe>, sólo debes copiar el código propuesto..



PD: Añadir imágenes CCC y random: <https://source.unsplash.com/400x300/?nature>

table, tr, td, th

Las tablas más sencillas de HTML se definen con tres etiquetas:

- `<table>` para crear la tabla
- `<tr>` para crear cada fila
- `<td>` para crear cada columna.

Además, HTML define la etiqueta `<th>` para indicar que una celda es cabecera de otras celdas. Los atributos de la etiqueta `<th>` son idénticos que los atributos definidos para la etiqueta `<td>`.

Existen los atributos `rowspan` y `colspan` que nos permiten decir cuantas filas o columnas ocupa esa celda.

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

table, th, caption

La etiqueta **<caption>** sirve para establecer el título de una tabla.

```
<table>
<caption>Titulo de la tabla</caption>
...
</table>
```

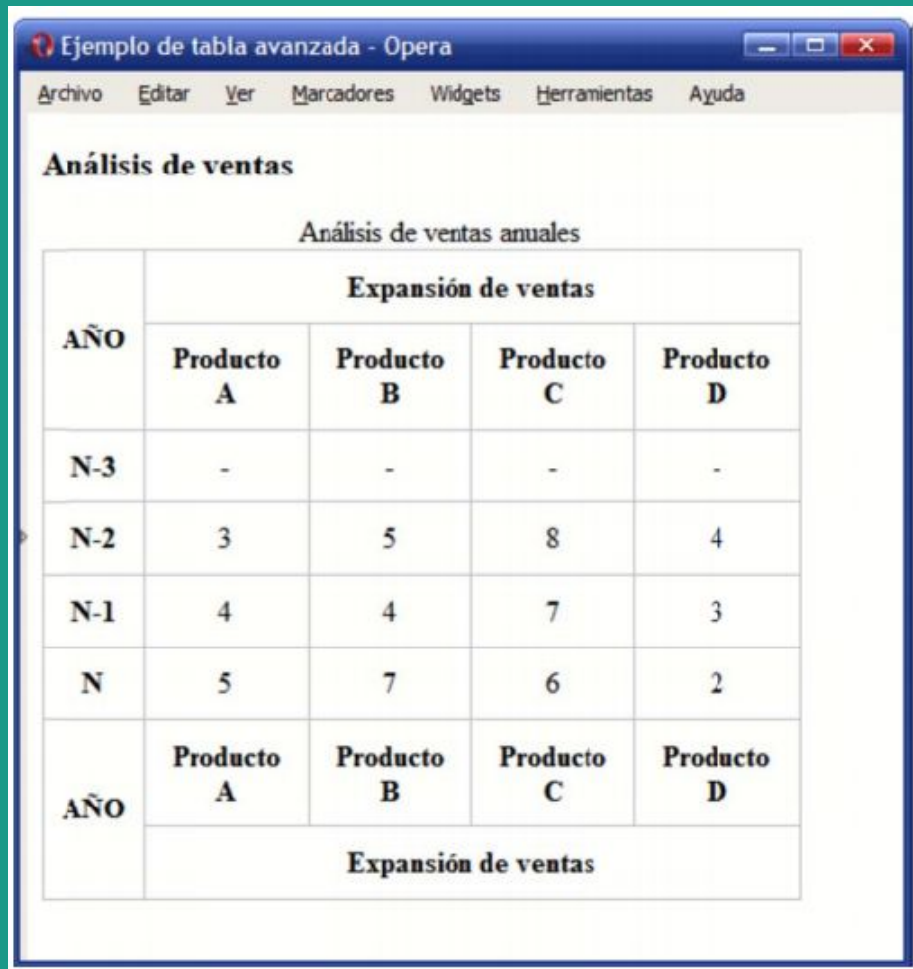
Las partes que componen las tablas complejas se definen mediante las etiquetas:

- **<thead>**: la cabecera de la tabla
- **<tbody>**: sección de datos
- **<tfoot>**: el pie de la tabla

Ejercicio 5

Al ejercicio anterior añade una nueva página **Ej05.html** que contenga la tabla de Análisis de Venta. Haz que podamos llegar a esta página desde el menú de navegación.

Utiliza todos los tags de tablas (thead, tbody, tfoot)



The screenshot shows a web browser window with the title "Ejemplo de tabla avanzada - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The main content area displays a table titled "Análisis de ventas".

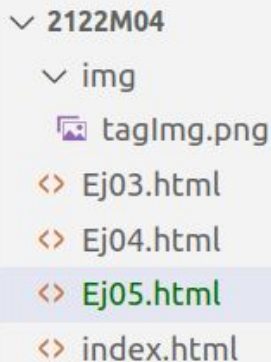
The table has a main title "Análisis de ventas anuales" and is structured as follows:

AÑO	Expansión de ventas			
	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2
AÑO	Producto A	Producto B	Producto C	Producto D
	Expansión de ventas			

Index

Crea una página **index.html** que tenga un menú con un enlace a todos los ejercicios (3, 4 y 5).

Este debería ser el árbol de trabajo (una carpeta *img* si has puesto imagen de directorio):



```
▼ 2122M04
  ▼ img
    tagImg.png
  <> Ej03.html
  <> Ej04.html
  <> Ej05.html
  <> index.html
```

M04 Lenguajes de marcas

Años 2021/20222

ITB: DAWe, DAMr, DAMi

Aprenderemos Xml, HTML, CSS y JS

- [Ejercicio 3: Estructura básica](#)
- [Ejercicio 4: Textos y Multiemdia](#)
- [Ejercicio 5: Tablas](#)

Publicar

Toca publicar nuestro proyecto, para eso usaremos **GitHub**.

- Crea una cuenta GitHub usando el correo del ITB y haz que la carpeta sea Tu **NombreApellido**
- Vídeo de como sincronizar Visual Studio Code con GitHub (el proyecto será **M04Ejercicios**) <https://youtu.be/ngow7sPfSDQ>



Nota: Desde este vídeo hay todos los vídeos para aprender GitHub.

- Publicar la web, os dejo un documento que os explica como se publica una web.
- Ahora simplemente con compartiendo vuestra URL ya se puede ver vuestro trabajo:



HTML

Interacción

-  Formularios
-  Más elementos

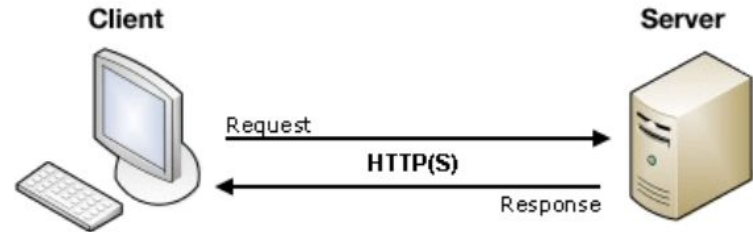
form

Para crear un formulario en HTML, debemos usar el elemento `<form>`. A partir de aquí se irán añadiendo los elementos que queremos que estén en nuestro formulario usando `<input type="">`.

El atributo `action` del `<form>` nos permite indicar qué acción se llevará a cabo en el momento de hacer “submit” en el botón del formulario.

`<label>` es la etiqueta que permite poner un nombre/título al campo del formulario.

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



type

En el atributo **type** de `<input>` será dónde de indicaremos cómo será el campo a introducir:

- ❑ **text:** Entrada de texto
- ❑ **radio:** Selector de una opción.
- ❑ **checkbox:** Selector de múltiples opciones
- ❑ **submit:** Botón que lanza la acción especificada de enviar el formulario.
- ❑ **reset:** Borrar todos los valores introducidos dejando los valores por defecto.
- ❑ **button:** Botón al que le podemos asignar una acción.
- ❑ **email:** de esta manera podemos validar el formato (no si la dirección existe)

type

- ❑ **url:** Podemos comprobar que es una dirección url y añade **http://** en caso que el usuario no lo haya puesto.
- ❑ **date:** de cara a evitar problemas con los diferentes formatos de fechas, ofrece al usuario un calendario.
- ❑ **time:** como de hora sobre 24h, ofreciendo un selector de hora al usuario y evitando así problemas de formato.
- ❑ **range:** muestra un control deslizante.
- ❑ **datetime, tel, search, color**



type

El elemento `<input>` también tiene otros atributos que podemos usar:

- **id** identificador único en toda la página HTML, en el DOM
- **name** nombre que le asignamos al elemento y que nos permite agrupa label/input o varios radio
- **value** valor que tiene el elemento.
- **required** indica que es obligatorio rellenar el control.
- **disabled** muestra el elemento deshabilitado.
- **readonly** solo de lectura.
- **placeholder** es un mensaje que sale en el input informando cómo debemos rellenar el campo y desaparece.
- **selected** indica que la opción está seleccionada.
- **tabindex** permite usar el tabulador entre los diferentes elementos (ojo con el orden)
- **pattern** indicamos un patrón, igual que en XSD.

Methode: Get / Post

Supongamos este formulario.

```
<form action="/action_page.php" method="XXX">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>
```

A visual representation of the HTML form code. It shows a vertical green line on the left. To its right, the text "First name:" is followed by a text input field. Below that, "Last name:" is followed by another text input field. At the bottom, there is a "Submit" button.

First name:

Last name:

Get



Supongamos este formulario. Si usamos el método **GET**, los datos que se recogen del formulario viajan por la **URL**, es decir que añadiremos toda la información que esté rellenando el usuario en la URL, de tal manera que queda visible al usuario, queda marcado en el historial y por lo tanto es muy sensible a ser modificada por cualquier persona que tenga acceso al navegador.

https://aliciaitb.github.io/action_page.php?fname=Alicia&lname=Vazquez

Para identificar los campos usamos el **ID** de cada elemento, su identificador, que debe ser único en toda la página, único en el **DOM** (Document Object Model).

Ejemplo: Formularios de altas.

Post

Si usamos **POST**, la información se encapsula y envía al servidor de manera **oculta**, invisible al usuario, la URL no se modifica. No queda constancia en el Historial de navegación de que se ha enviado un formulario ni el usuario puede modificar nada puesto que la URL corresponde simplemente al fichero.

Este es un método mucho más seguro para la protección de los datos, sin embargo, los datos pueden enviarse de nuevo si recargamos la página. Este problema deberá ser controlado en el servidor.

Ejemplo: Compras.



Recap Get / Post



	GET	POST
Visibilidad	Visible en la barra de direcciones para el usuario	Invisible para el usuario
Marcadores e historiales de navegación	Los parámetros URL se guardan junto al URL	Los parámetros URL no se guardan junto al URL
Caché y registro del servidor	Los parámetros URL se guardan sin cifrar.	Los parámetros URL no se guardan automáticamente
Comportamiento al actualizar el navegador o retroceder	Los parámetros URL no se envían de nuevo	El navegador advierte de que los datos del formulario se enviarán de nuevo
Tipo de datos	Solo caracteres ASCII	Caracteres ASCII y datos binarios
Longitud de datos	Limitado al máximo del URL (2048 caracteres)	Ilimitado



HTML

Interacción

-  Formularios
-  Más elementos

datalist

Podemos crear una lista de opciones para el **type="list"**, asignado el mismo identificador, **id**, al **input** que al **datalist**.

```
<input id="form-person-title" type="text" list="mylist">
<datalist id="mylist">
  <option label="Mr" value="Mr">
  <option label="Ms" value="Ms">
  <option label="Prof" value="Mad Professor">
</datalist>
```

progress

El elemento **progress** es utilizado para representar un *avance* o *progreso* en la ejecución de una tarea, como puede ser la descarga de un fichero o la ejecución de una tarea compleja. En el **value** indicamos cual es el progreso

```
<!DOCTYPE html>
<html>
<body>

<h1>The progress element</h1>

<label for="file">Downloading progress:
</label>
<progress id="file" value="32" max="100">
32% </progress>

</body>
</html>
```

The progress element


Downloading progress: 

meter

El elemento **meter** es muy similar a **progress**. Se usa para representar escalas de medidas conocidas, como la longitud, masa, peso, uso de disco, entre otras. Define los siguientes atributos:

- **value**: representa el valor actual. Si no se especifica, se toma como valor el primer número que aparece en el contenido del elemento. Su valor por defecto es 0.
- **min**: el mínimo valor permitido. El valor por defecto es 0.
- **max**: el mayor valor permitido. Si no se especifica, su valor por defecto es 1, a menos que el valor mínimo definido sea mayor que 1, en cuyo caso el valor de max será igual a min.
- **low**: es considerado el límite inferior del rango de valores.
- **high**: es considerado el límite superior del rango de valores.
- **optium**: representa el valor óptimo del elemento, y se encuentra entre min y max.

```
<p>Your score is:  
  <meter value="91" min="0" max="100"  
    low="40" high="90" optimum="100">A+</meter>  
</p>
```

Your score is: 

Ejercicio 6

Al proyecto anterior añade una nueva página **Ej06.html** que contenga un formulario con todas las opciones que hemos visto en clase. Añade el enlace a esta páginas desde el menú de navegación del **index.html**

¿Qué te parece hacer un formulario como carta a los Reyes? **¿Habéis sido buenos?**



```
<form action="index.html"
method="get">
```

...

```
</form>
```

De esta manera verás todos los campos en la URL, pero no habrá respuesta si no hay nada en JS.

Referencias

W3CSchools: un sitio de referencia que viene muy bien para consultar puntualmente dudas técnicas muy concretas.

Elementos: Todos los elementos de HTML5 clasificados.

Como estructurar una página web: Más allá del HTML hay una distribución lógica, humana que nos dice cómo distribuir la información. Eso y el uso correcto de los elementos que ofrece HTML5 es garantía de éxito.

PildorasInformaticas: https://youtu.be/ANF1X42_ae4