

HTML

- HyperText Markup Language
- Lenguaje de Marcado estándar para hacer páginas Webs.
- Recomendado por la W3C

HTML



DOCTYPE

```
<!DOCTYPE html>
```

DOCTYPE

- Tiene que ser lo primero en el HTML
- No es una etiqueta!!
- Es una instrucción al navegador para que sepa la versión HTML de la página Web.
- En este caso indica que la página web están en HTML5
- Siempre debemos de ponerlo
- Se aconseja usar

DOCTYPE

La versión HTML 4.01 teníamos que poner la siguiente instrucción:
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
["http://www.w3.org/TR/html4/strict.dtd"](http://www.w3.org/TR/html4/strict.dtd)>

DTD: Definición del tipo de documento

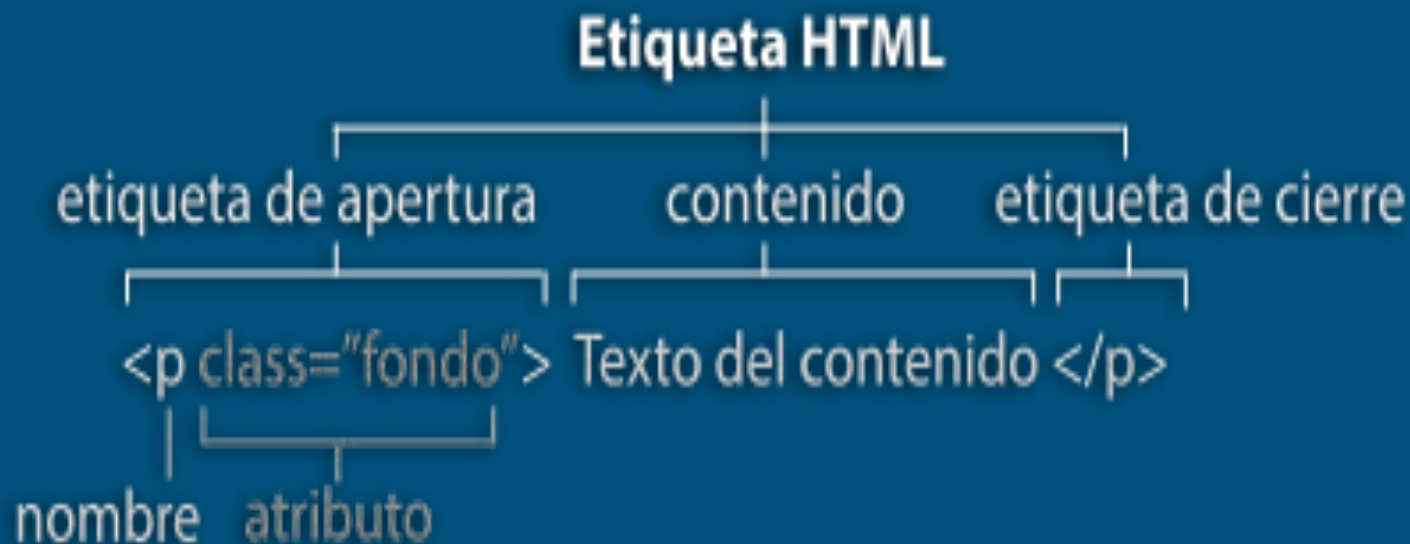
Era necesario porque HTML 4.01 porque estaba basado en SGML.

SGML: Lenguaje de marcado estándar o generalizado

HTML

- `<HTML></HTML>`
- Sirve para crear la estructura principal de la página HTML
- Todo lo que está entre estas etiquetas será interpretado como código HTML
- Si no usamos esta etiqueta, es posible que los navegadores interpreten incorrectamente el código HTML.
- Indicar atributo “lang” para especificar el lenguaje del documento.
`<html lang=“es”>`

HTML



Atributos

- Son valores adicionales que configuran los elementos.
- Se aconseja que los nombres de los atributos siempre en minúscula.
- Se aconseja usar siempre “dobles ” o ‘ simples’ para los valores de los atributos (No es obligatorio).
- Se aconseja no dejar espacios entre el nombre, el = y el valor.

Atributos

- Atributos Globales:

Los atributos globales son atributos comunes a todos los elementos HTML; pueden usarse en todos los elementos, aunque pueden no tener efecto en algunos de ellos:

- **Id**: Identificador del elemento
- **Class**: Definir clases de estilo
- **Accesskey**: atajo de teclado

- Lista de todos los atributos globales

https://developer.mozilla.org/es/docs/Web/HTML/Atributos_Globales

HEAD

- <head></head>
- Contiene los metadatos de la página web, enlaces a, o definiciones de scripts y hojas de estilo:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/st
[- <html>
  [- <head profile="http://www.w3resource.com/profiles.html">
    <meta content="text/html; charset=iso-8859-1" http-equiv="Content-Type">
    <title>html head tag example</title>
    [+ <link type="text/css" href="test.css" rel="stylesheet">
  </head>
  [+ <body>
</html>
```

HEAD

<title> Título de la página

<style> Dar estilo a la página

<base> Especifica la URL base de todas las URLs relativas del documento

<link> => Define la relación entre la página y archivos externos. (CSS)

<meta> => Provee información sobre el documento HTML que no se ve en la página, pero sirve para navegadores, motores de búsquedas y servicios Webs.

HEAD

`<script>` Es usado para definir un script del lado del cliente.(Javascript)

`<noscript>` Para ofrecer un contenido alternativo para navegadores que tienen desactivado usar scripts o no soportan la tecnología de script que se utiliza. Dentro de esta etiqueta podemos incluir todo lo que puede añadirse dentro de la etiqueta body.

Cuando obtenemos scripts, css, imagenes, videos, etc, desde una página externa, siempre es aconsejable que incluyamos el protocolo **HTTPS**, en el caso de que no fuera accesible por HTTPS, usaríamos HTTP, pero siempre hay que especificar el protocolo.

HEAD

- No es necesario incluir el valor del atributo type, para las etiquetas script y rel.
- **<META>** Es recomendable indicar siempre la codificación de las páginas con esta etiqueta:
 - `<meta charset="utf-8">`
- Por defecto si no se indica nada, el valor será utf-8.

BODY

- `<body></body>`
- Contiene todo el contenido de una página HTML

Párrafos

- `<p></p>` => Para incluir párrafos en el documento.
- Los navegadores aplican un estilo predeterminado a esta etiqueta.
- Este estilo puede modificarse con CSS.

DIV

- **<DIV>** Document división
- Sirve para dividir el documento en secciones.

Salto de línea

- **
** Salto de línea
- No es aconsejable usarlo para manipular las posiciones de los elementos en el documento HTML

TAG SELF-CLOSED

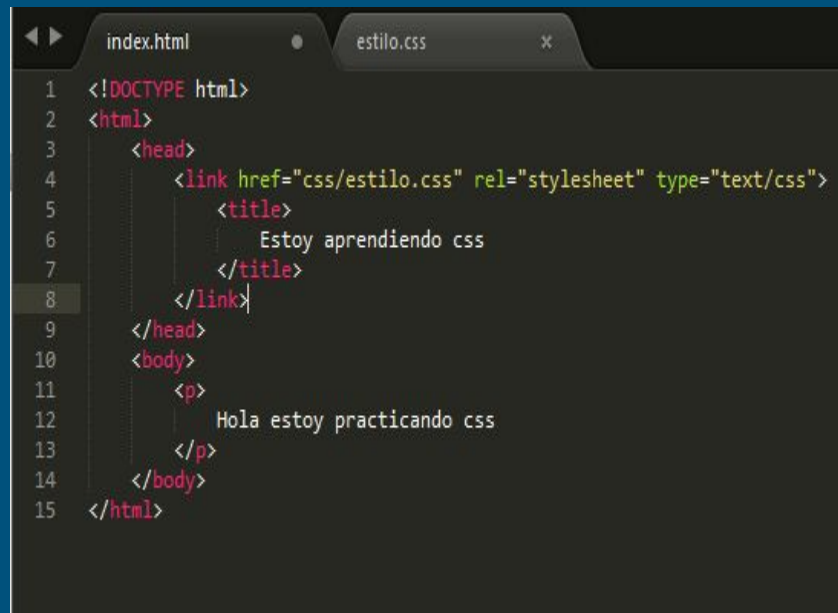
- Elementos “void”, con auto-cierre de etiqueta:
 - `<input />` => No hay que poner `<input></input>`
 - `<input type="text" id="name" />`
- Listado:
 - `area`, `base`, `br`, `col`, `embed`, `hr`, `img`, `input`, `keygen`, `link`, `menuitem`, `meta`, `param`, `source`, `track`, `wbr`.
- En HTML5 no es necesario incluir la barra de cierre en las etiquetas, pero si aconsejable hacerlo en todas. (HTML5 y los diferentes navegadores tienen una gran capacidad para adaptarse a errores de sintaxis en el código, pero eso no significa que sea código válido.)

NIVELES

- Niveles de anidamiento (Nesting levels) : Se usan relaciones padre e hijos para decir que esta anidado con otro.
- `<div id="div1">` Es el padre de `div2`
 - `<div id="div2">` Es el padre de `p3` y el hijo de `div1`
 - `<p id="p3">` Es el hijo de `div2`.
 - "Hola Mundo" => No es hijo de `p3!!!!`, sino el contenido HTML de `p3`
 - `</p>`
 - `</div>`
- `<div>`

INDENTACIÓN Y FORMATEO

- Indentación y Formateo:
 - Siempre incluir cada elemento en una nueva línea y con su indentación correspondiente: siempre incluirla cuando un elemento está dentro de otro.
- W3c aconseja usar dos espacios para indexar , ya que así permite más caracteres por línea.



The screenshot shows a code editor with two tabs: 'index.html' and 'estilo.css'. The 'index.html' file is open, displaying the following code with proper indentation (two spaces per level):

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <link href="css/estilo.css" rel="stylesheet" type="text/css">
5     <title>
6       Estoy aprendiendo css
7     </title>
8   </link>
9 </head>
10 <body>
11   <p>
12     Hola estoy practicando css
13   </p>
14 </body>
15 </html>
```

COMENTARIOS

- Comentarios: Para comentar el código

`<!-- Todo lo que hay entre estos caracteres será ignorado por el navegador -->`

(útil para depurar)

HEADINGS

- Headings: Para capturar la atención del lector en los textos, por ejemplo titulares de noticias. Organizamos nuestro contenido.
 - <h1>
 - <h2>
 - <h3>
 - <h4>
 - <h5>
 - <h6>

SPAN

- Span => Sirve para mostrar pequeños textos, sin tener que dividir las secciones del documento.
- Si queremos poner en negrita el texto de un párrafo podemos utilizar un span con un estilo concreto para ello:
 - `<p>Este es el texto y quiero resaltar la siguiente palabra SPAN, las demás no la resaltamos</p>`
 - El elemento div, no podríamos usarlo dentro del párrafo, pero el span si.

Etiquetas de formateo

`` texto en Itálica

`` texto en negrita

`` texto en negrita

`<i>` texto en itálica

Es aconsejable usar `` y `` porque para los motores de búsquedas, estas etiquetas si tienen valor, las otras no.

LISTAS

Listas

- ul => Listas desordenadas
- ol => Listas ordenadas
- li => Cada elemento de la lista

Ejemplo:

```
<ul>
```

```
<li>HTML 5</li>
```

```
<li>CSS 3</li>
```

```
<li>Javascript</li>
```

```
</ul>
```


Imágenes

- **** para mostrar imágenes
- Usamos el atributo **“src”** para indicar la ruta de la imagen.
- Usar el atributo **“alt”** , para mostrar un texto alternativo, en el caso de que no pudiera cargarse la imagen. También es una buena práctica para hacer más **accesible nuestra Web**, los navegadores para personas invidentes, que leen el contenido de las páginas, utilizan este atributo para describir la imagen al usuario.
- Si la imagen no da información importante al usuario, no es conveniente incluir dicho atributo.

Videos (1/2)

- Etiqueta para mostrar videos
- Usamos el atributo “src” para indicar la ruta de la imagen.
- Con el atributo “controls” indicamos al navegador que incluya los botones básicos para controlar el video.
- Los vídeos de youtube no pueden usarse con esta etiqueta

Videos (2/2)

- Podemos incluir varios formatos de nuestro video por si el navegador no soporta alguno de ellos:

```
<video controls>
  <source src="movie.mp4" type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"' />
  <source src="movie.webm" type='video/webm; codecs="vp8, vorbis"' />
</video>
```

Audio

- Igual que la etiqueta video, pero en este caso incluimos sonido a la página.

```
<audio controls="controls">
  <source src="https://mainline.i3s.unice.fr/mooc/horse.ogg" type="audio/ogg" />
  <source src="https://mainline.i3s.unice.fr/mooc/horse.mp3" type="audio/mp3" />
</audio>
```

Hyperlinks (Etiqueta <a>)

- Vincular a otras páginas
- Atributo “**href**” indicamos el destino.
- Podemos usar imágenes como link en vez de un texto.
- Podemos hacer links a elementos de nuestra página

```
<a href="https://www.ciclismoyrendimiento.com" >Ir al blog de ciclismo</a>
```

```
<a href="https://www.google.es"></a>
```

```
<a href="#parte1">Vamos al div parte1</div>
```

```
<div id="parte1"></div>
```

Hyperlinks atributos

- **target:** Especifica donde abrir la página:
 - Sólo deberíamos usar “_self”(misma página) y “_blank” para una nueva pestaña. Los otros casos son para marcos.
- **rel:** indicar la relación que tienen ambas páginas
- **hreflang:** para indicar el lenguaje de la página
- **MIME_type:** para indicar el MIME type del documento
- **media:** especifica que dispositivo esta la página optimizada. Lo conveniente sería “all”, pero en algunos casos, sería aconsejable especificar.
- **Download:** indica que va a descargar un archivo
- ***MIME** => Multipurpose Internet Mail Extensions (Especificaciones para el intercambio de archivos en internet)

Buenas prácticas en hiperlinks

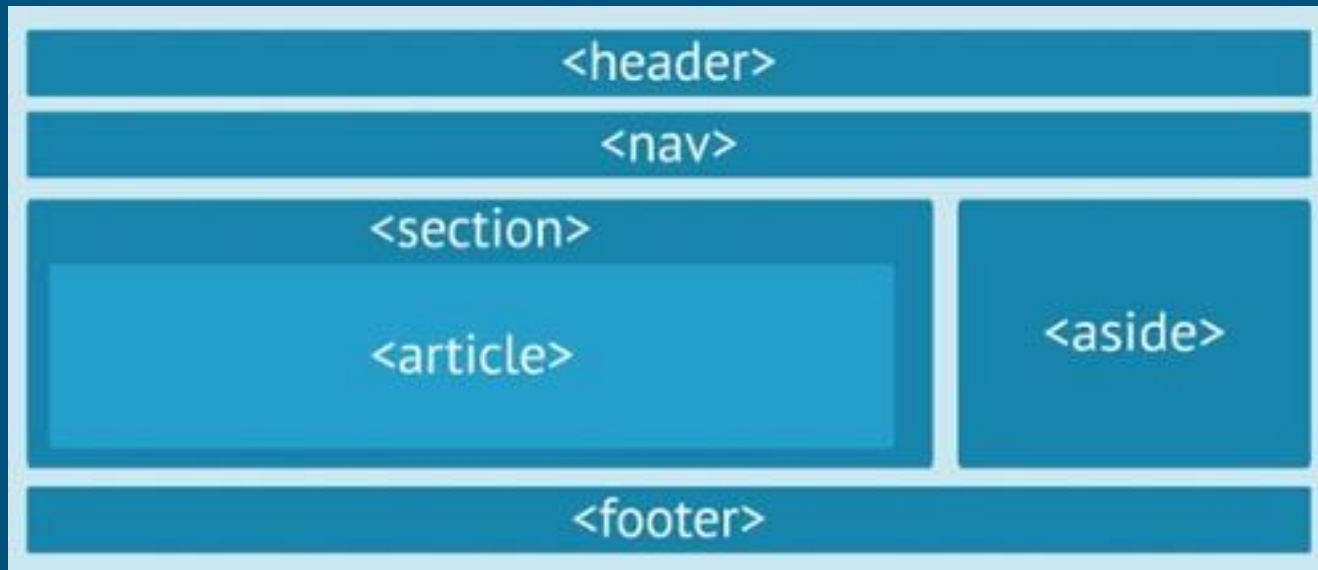
- No repetir la url como parte del texto.
- Usar siempre un texto claro y conciso.
- Reducir el número de veces que usamos el mismo texto para ir a sitios distintos.
- Usar rutas relativas siempre que estemos en local.
 - Más fácil de mantener si cambiamos de dominio

Buenas prácticas en hiperlinks

- Para descarga de archivos usar el atributo “download”. (Sólo se pueden archivo del mismo dominio)
- Usar esta etiqueta para enlazar páginas, no ejecutar código javascript o enviar correos.

Etiquetas semánticas

<article>
<aside>
<details>
<figcaption>
<figure>
<footer>
<header>
<main>
<mark>
<nav>
<section>
<summary>



Section

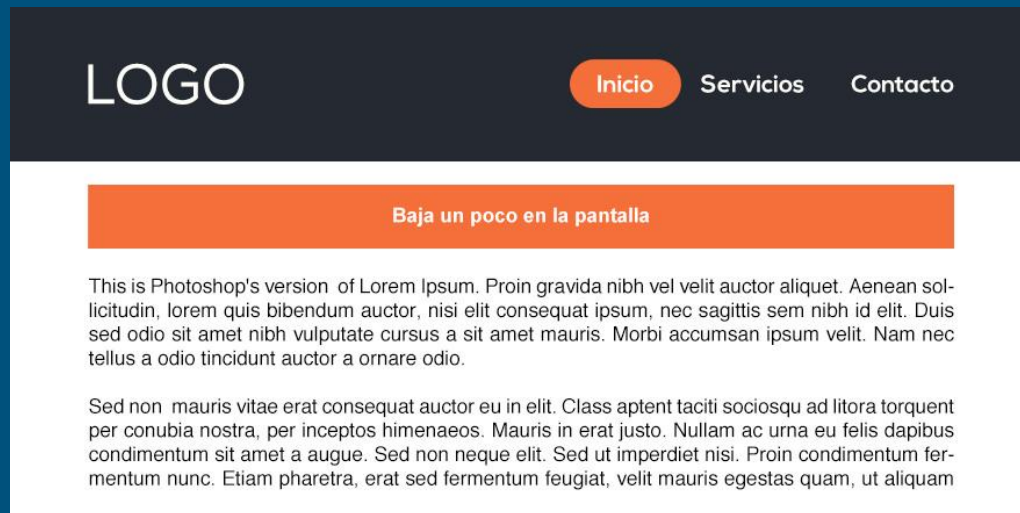
- Para agrupar contenidos:
 - Introducción
 - Información de contacto
 - Suele ser casi todo que contenga una cabecera
- Es una buena práctica incluir etiquetas de cabeceras después de esta etiqueta.

Article

- Para mostrar una información que tiene sentido por sí misma, y si fuera posible debería leerse de forma independiente a la página web:
 - Comentarios en los foros
 - Comentarios en blogs
 - Artículos de noticias.
- En internet suelen confundirse article y section. Pero en algunos casos incluimos section dentro de article y article dentro de section.

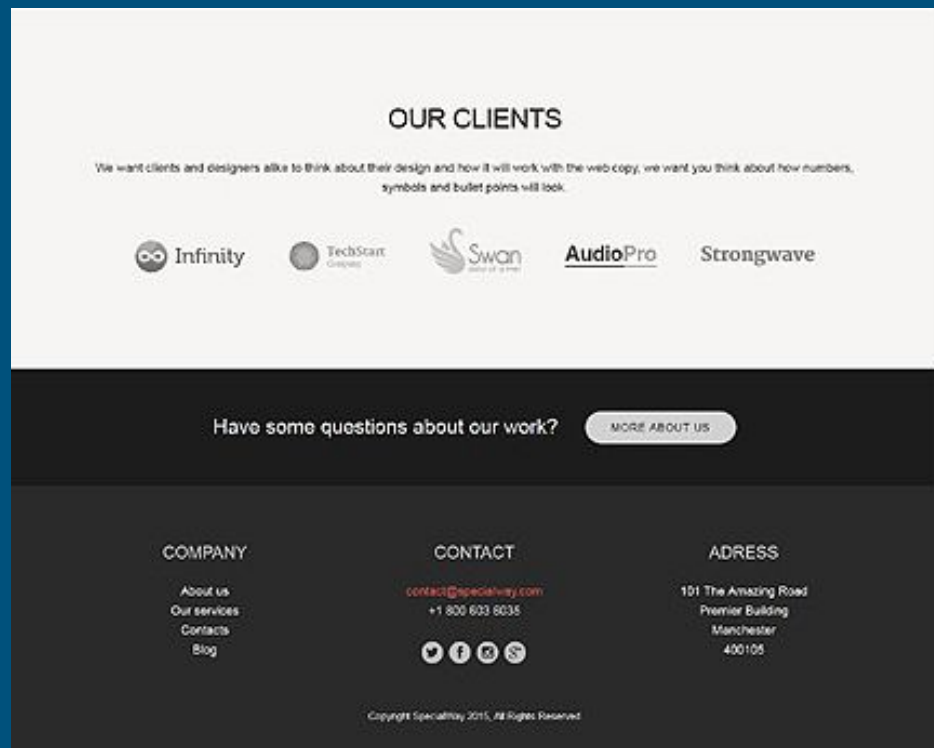
Header

- Cabecera del documento
 - Para incluir los elementos de identificación e introducción.



Footer

- Pié de página:
 - Información de contacto
 - Copyright
 - Los links a los términos de uso
 - Autor de la página



Nav

- Menú de navegación

Aside

- Contenido adicional y separado del principal:
 - Anuncios
 - Promociones
 - Artículos más leídos

Figure and figcaption

- Para incluir una explicación visual a la imagen que se incluye:

```
<figure>
  
  <figcaption>Fig1. – El oso de Canada.</figcaption>
</figure>
```


Details and Summary

- Para contenido adicional que el usuario puede ver o ocultar

```
<details>
  <summary>DIW- HTML5 + CSS.</summary>
  <p> Ejemplo de HTML5.</p>
  <p>HTML5 es súper entretenido</p>
</details>
```

Main

- Para especificar el contenido principal del documento

Time

- Para definir una fecha u hora en la página

```
<p>Los viernes terminamos a las <time>18:10</time> DIW</p>  
<p>Me tocará la primitiva el <time datetime="2019-12-28 20:00">Día de  
los inocentes</time>.</p>
```

- Los motores de búsquedas usan estas etiquetas para obtener más información.

Mark

- Para marcar un texto relevante:

```
<p>La siguiente palabra <mark>relevante</mark> vamos a marcarla</p>
```

- Cuando sea posible, es preferible utilizar **<mark>** a **** o ****.
- El estilo de mark se puede cambiar con css.

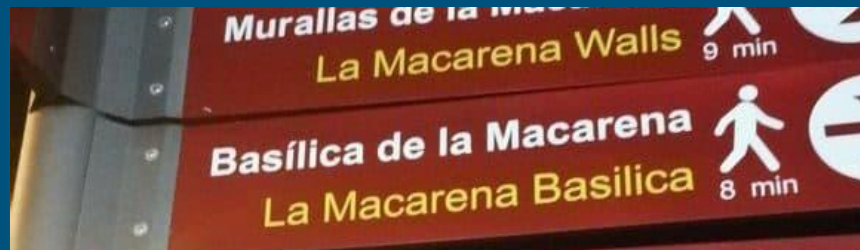
Atributo Translate

- Sirve para indicar si un texto se debe traducir o no por los traductores de los navegadores:

```
<span translate="no" class="monumento">Basílica de la Macarena</span>
```

Ejemplos de uso:

- Páginas Webs que contienen código fuente que no debe traducirse
- Nombres de calles y autores.



Atributo data-*

- EL atributo data-* sirve para guardar información privada en los elementos utilizados sin que se visible por el usuario que entra en la página.
- Este atributo nos permite dar información personalizada a cada uno de los
- elementos.
- Para usar este atributo debemos rellenar las siguientes dos partes:
 - El nombre del atributo personalizado: data-{nombre_del_atributo}
 - El valor del atributo
- Además estos atributos serán ignorados por los agentes de usuario (navegadores).

Atributo data-*

- Ejemplo:

```
<h1>Alumnos de clase</h1>
<ul>
  <li data-edad='23' data-email='juanito@gmail.com'>Juanito</li>
  <li data-edad='18' data-email='pepito@gmail.com'>Pepito</li>
  <li data-edad='15' data-email='jaimito@gmail.com'>Jaimito</li>
</ul>
```

Formularios

- Usamos la etiqueta de `<form>` para definir el elemento del formulario, que se encargará de enviar la información de una página a otra.
- Atributos:
 - **action:** donde enviaremos los datos
 - **enctype:** para definir cómo debe ser encriptado los datos cuando se envía por el método "POST"
 - **method:**
 - GET:
 - Los datos serán visibles en la url.
 - Añade datos de formulario en la URL en pares nombre/valor
 - Nunca usar GET para enviar datos confidenciales!
 - GET es mejor para datos no seguros, como las consultas en Google

Formularios

- Atributos:
 - **method:**
 - POST:
 - Siempre usar POST si los datos del formulario contienen información sensible o personal. El método POST no muestra los datos que se envían en el formulario.
 - **novalidate** no se valida al ejecutarse
 - **name:** nombre del formulario
- Para ejecutar un formulario podemos usar un botón de tipo submit.

Formularios

- Todos los elementos del formulario deben identificarse con una etiqueta.
- En algunos casos usaremos <label>

```
<label for="color">Color casco</label>  
<input type="color" name="casco_color_control" /><br>
```

- Otras veces ya estarán incluidas dentro del elemento:
 - <button type="submit">Aceptar</button>
- Si todo está bien etiquetado, podemos acceder a ellos a través del teclado, voz o usando lectores de pantalla.

Formularios

- Para formularios grandes y complejos usar la etiqueta `<fieldset>` y `<legend>` para agrupar elementos del mismo tipo. [Ejemplo](#).

```
<fieldset>
  <legend>Datos Personales</legend>
  <label for="nombre">Tu nombre</label>
  <input id="fnombre" type="text" name="fname"/>

  <label for="apellidos">Tus apellidos</label>
  <input id="fapellido" type="text" name=" fapellido "/>
</fieldset>
```

Inputs

- Los inputs son los elementos que se utilizarán para incluir información y enviarla al servidor para que este las procese:
 - `<input />`
- Existen varios tipos, que se definirán con el atributo “**type**”.
- Dependiendo del **tipo**, se mostrará de **una forma u otra en el navegador**.
- El valor del input viene definido en el atributo “**value**”, en la mayoría de los tipos.
- Los datos se identifican con el atributo “**name**”, debe ser único en el formulario
 - Si se omite el atributo name, los datos de ese input no se enviarán. [Ejemplo](#)

Tipos de inputs

- **text:** introducir texto
- **Hidden:** Crea un campo con datos ocultos a la vista del usuario.
- **Password:** El texto introducido aparece oculto.
- **Button:** para crear botones que interactúan con la aplicación
- **Submit:** para enviar los datos del formulario
- **Image:** botones de submit con imagen, sólo que hay que incluir la ruta de la imagen con el atributo “src”.

Para subir ficheros

- **File:** Para subir ficheros. [Ejemplo](#)

```
<label for="contrato">Contrato</label>  
<input type=file value="Subir fichero" />
```

- Para que puedan subirse ficheros al servidor debe indicarse en la etiqueta **<form>** el atributo `enctype="multipart/form-data"`.

Radio

Selecciona sólo un valor de un conjunto. [Ejemplo](#)

```
<input type=radio value="madrid" name="equipo"/>  
<input type=radio value="betis" name="equipo"/>  
<input type=radio value="sevilla" name="equipo"/>  
<input type=radio value="Cadiz" name="equipo"/>  
<input type=radio value="Recreativo" name="equipo"/>
```

CheckBox

- Para marcar una casilla con un valor. Si está marcada, su valor es el que se indica en el atributo “value”, sino se marca esa casilla, su valor es NULO.

Formatos

- Facilitan la introducción de estos tipos de formatos específicos, desde cualquier dispositivo.
 - `<input type="email">` Email
 - `<input type="tel">` Teléfono
 - `<input type="URL">` Una URL

```
<label for="hora">Hora entrada al trabajo</label>  
<input type="time" name="hora_control" /><br>
```

```
<label for="satisfaccion">Satisfacción (min 0, max 10)</label>  
<input type="range" name="satisfaccion_control" min="0" max="10" value="0" /><br>
```

```
<label for="telefono">Teléfono</label>  
<input type="tel" name="tel_control" /><br>
```

Búsqueda

- Para facilitar las búsquedas, aparece un listado de las búsquedas recientes realizadas. [Ejemplo](#)

```
<input type="search">
```

Números

- Para facilitar el añadir números en un formulario:
 - **Max:** es el valor máximo
 - **Min:** el valor mínimo
 - **Step:** Indica que los valores deben ser múltiplos de ese número.

Ejemplo

```
<label for="vacaciones">Dias de vacaciones disfrutados (min 0, max 23)</label>  
<input type="number" name="number_vacaciones" min="0" max="23" value="0" /><br>
```

Rangos

- Especifica un rango de valores seleccionables
- Usamos “datalist” y el atributo “list” para mostrar marca del rango

Ejemplo

```
<input type="range" value="5" min="0" step="2" max="10" list="ticks2"/>
<datalist id=ticks2>
  <option>0</option>
  <option>2</option>
  <option>4</option>
  <option>6</option>
  <option>8</option>
  <option>10</option>
</datalist>
```

Select

Para seleccionar un valor de una lista. [Ejemplo](#)

```
<select name="navegador">
  <option value="firefox">Firefox</option>
  <option value="ie">IE</option>
  <option value="chrome">Chrome</option>
  <option value="opera">Opera</option>
  <option value="safari">Safari</option>
</select>
```

Textarea

- Por si queremos incluir un texto grande, como un artículo. [Ejemplo](#)

```
<label for="observaciones">Observaciones</label>  
<textarea name="observaciones"></textarea><br>
```

Atributos elementos del formulario

- **Autocomplete:** Permite que el navegador pueda autocompletar un campo o no. SU valor es "ON" o "OFF";
- **Autofocus:** se sitúa en el elemento con dicho atributo
- **Pattern:** para especificar expresiones regulares que deben cumplirse
- **Múltiple:** Para emails y ficheros, nos permite introducir varios emails separados por coma, o añadir varios ficheros.
- **Placeholder:** Para mostrar un ejemplo del texto que puede introducirse.

```
<label for="nombre">Nombre</label>  
<input type="text" name="name_control" autocomplete="OFF" autofocus required /><br>
```

Listas

- EL usuario puede seleccionar un elemento de la lista o introducir el valor manualmente. [Ejemplo](#)

```
<input list="navegadores" id="minavegador" />
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
```


Canvas

- Para dibujar gráficos en la Web.
- Ejemplo:

https://www.w3schools.com/graphics/tryit.asp?filename=trycanvas_draw

Animaciones y otras apis.

- **requestAnimationFrame API:** Para realizar animaciones en la Web.
- **Web Storage:** Para guardar información localmente, sin necesidad de usar el servidor.
- **File API:** para permitir a las aplicaciones trabajar con archivos locales
- **Geolocation API:** Para obtener información de la longitud, latitud, altitud y velocidad.
- **Microdata:** Para agregar información semántica a elementos de la página, incluso permitiendo añadir grupos.

