
TEMA 2.I:HTML



¿QUÉ ES HTML?

- HTML es un **lenguaje de marcado**, usado para decirle al navegador cómo **estructurar la páginas**.
- Puede ser tan complejo o tan simple como desee el desarrollador web.
- Consiste en una **serie de elementos** que se utilizan para crear contenidos, organizarlos y estructurarlos.
- Cada elemento (etiquetas) envuelven un determinado contenido realizando una función concreta.
- En páginas sucesivas iremos viendo la función de cada uno de estos elementos.

ANATOMÍA DE UN ELEMENTO HTML



- **Etiqueta de apertura:** consiste en el nombre del elemento (en este caso, p) encerrada en paréntesis angulares de apertura y cierre. Esto indica que el elemento comienza, o empieza a tener efecto.
- **Etiqueta de cierre:** Es lo mismo que la etiqueta de apertura, excepto que incluye una barra diagonal antes del nombre del elemento. Esto indica donde termina el elemento, en este caso donde finaliza el párrafo.
- **El contenido:** es el contenido del elemento, en este caso solo texto.
- **El elemento:** El elemento lo conforman la suma de la etiqueta de apertura, la etiqueta de cierre y el contenido.

ELEMENTOS VACÍOS

- No todos los elementos de HTML siguen el patrón **<etiqueta_apertura>** contenido **</etiqueta_cierre>**
- Existen elementos que no tienen contenido y se denominan *elementos void*. Estos elementos, **al carecer de contenido, carecen de etiqueta de cierre.**
- Por ejemplo la etiqueta **
** que sirve para incluir saltos de línea en nuestros documentos.
- En XHTML lo que se hacía era utilizar el “**autocierre**”, que consistía en añadir una barra el final de la etiqueta de apertura. Siguiendo el mismo ejemplo:
`
`
- En HTML 5 no es necesario cerrar los elementos void, se aceptan las dos formas, pero **se considera una buena práctica cerrarlas todas.**

DOCTYPE

- Lo primero que debemos hacer al crear un documento HTML es indicar que versión de HTML vamos a utilizar. Esto lo hacemos mediante un **DOCTYPE**.

- El DOCTYPE más sencillo de todos es el de **HTML 5**:

```
<!DOCTYPE html>
```

- Para HTML 4.01:

```
<!DOCTYPE HTML PUBLIC  
“-//W3C//DTD HTML 4.01 Transitional//EN”  
“http://www.w3.org/TR/html4/loose.dtd”>
```

ELEMENTOS ANIDADOS

- Los elementos pueden ir dentro de otros elementos, esto se llama anidamiento.
- Debemos asegurarnos que los elementos estén correctamente anidados, es decir, la **etiqueta a cerrar** en un determinado punto **será la última que se ha abierto**. Por ejemplo:

```
<p>Esto es <strong>Correcto</strong></p>
```

```
<p>Esto es <strong>Incorrecto</p></strong>
```

ANATOMÍA DE UN DOCUMENTO HTML

```
<!DOCTYPE html>
```

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

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title> Título de la página </title>
```

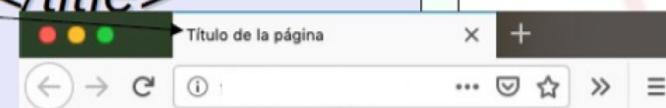
```
  </head>
```

```
<body>
```

```
  Cuerpo
```

```
</body>
```

```
</html>
```



Cuerpo

ANATOMÍA DE UN DOCUMENTO HMTL

- **<!DOCTYPE html>**: El doctype. En sus inicios, cuando el HTML llevaba poco tiempo (alrededor de 1991/2), los doctypes servían como enlaces al conjunto de reglas que la página HTML debía seguir para considerarse buen HTML. En la actualidad se ignoran los enlaces pero hay que incluirlo para que distinga la versión.
- **<html lang="es"></html>**: Este elemento engloba todo el contenido de la página y es conocido en ocasiones como el elemento raíz. El atributo lang permite especificar el lenguaje en el que está escrita la página.
- **<head></head>**: El elemento **<head>** (cabecera). Este elemento actúa como contenedor para otros parámetros que queramos incluir en el documento HTML que NO SERÁ visible a los visitantes de la página. Incluye cosas como palabras clave y la descripción de la página que quieras mostrar en los resultados de búsqueda, así como la hoja de estilo para formatear nuestro contenido, declaraciones de codificación de caracteres y más.
- **<meta charset="utf-8">**: establece que nuestro documento HTML usará la codificación utf-8, que incluye la gran mayoría de caracteres de todos los lenguajes humanos conocidos.
- **<title></title>**: Este elemento establece el título de tu página, que aparece en la pestaña/ventana de tu navegador cuando la página se carga y se utiliza para describir la página cuando la agregas a tus marcadores o la marcas como favorita.
- **<body></body>**: El elemento **<body>**. Contiene todo el contenido a mostrar a los usuarios cuando visitan nuestra página, ya sea texto, imágenes, videos, juegos, pistas de audio reproducibles o cualquier otra cosa.

ESTRUCTURA VS FORMATO

- Como ya hemos comentado, HTML es actualmente un lenguaje que define la **estructura lógica** de un documento, **NO su aspecto** (que queda a la elección del navegador).
- En las primeras versiones de HTML existían etiquetas del tipo negrita (****) pero se han marcado como **obsoletas** ya que no tendrían sentido en un navegador gps, pero en cambio una etiqueta destacado (****) si.
 - **, ...**
- Para definir el aspecto que tendrán nuestras páginas web utilizaremos **hojas de estilo** que nos permitirán separar el **contenido del diseño**.
- Además separando contenido de diseño podremos **actualizar fácilmente el diseño de nuestra web**, simplemente tendremos que cambiar un archivo. En cambio si damos formato desde los archivos html que forman la página web, deberemos editar cada uno de los archivos de nuestro portal web.

VALIDAR EL CÓDIGO HTML

- Los **navegadores no son estrictos con las normas HTML**; es decir, aunque tengamos algunos fallos, van a ignorarlos en aras de mostrar la mejor versión del código que hagamos.
- El problema es que si escribimos mal el código los resultados que podemos obtener pueden diferir de un navegador a otro.
- Para evitar problemas, el W3C pone a nuestra disposición un servicio de validación para **comprobar que nuestras páginas cumplen con los estándares**.
- El validador oficial de la W3C (<http://validator.w3.org>) permite validar todas las versiones de HTML, incluida la versión 5.

ENTIDADES DE CARÁCTER

- Sintaxis:
 - &codigo;
- Donde “codigo” es una palabra reservada o un número que indica el carácter a representar.
- Permiten poner **caracteres especiales** en un documento:
Letras acentuadas, etc.

á	á	"	“
é	é	&	&
...		<	<
Á	Á	>	>
...		 	
ñ	ñ	©	©
Ñ	Ñ	€	€
¿	¿	&rdash;	-
á	á	"	“

ELEMENTOS DE BLOQUE VS ELEMENTOS EN LÍNEA

Existen dos categorías de elementos en HTML:

- Los **elementos de bloque** forman un bloque visible en la página, aparecerán en una nueva línea después de cualquier contenido anterior, y cualquier contenido que vaya después también aparecerá en una nueva línea. Los elementos de bloque tienden a ser **elementos estructurales en la página** que representan por ejemplo párrafos, listas, menús de navegación, pies de página, etc.
- Los **elementos en línea** son aquellos que están **contenidos dentro de elementos de bloque** y rodean solo pequeñas partes del contenido del documento, no párrafos enteros ni agrupaciones de contenido. Un elemento en línea no hará que una nueva línea aparezca en el documento; normalmente aparecen dentro de un párrafo o texto, por ejemplo es el caso de un elemento `<a>` (hipervínculo) o elementos de énfasis como `` o ``.

ESPACIOS EN BLANCO EN HTML

- No importa cuantos espacios en blanco se utilicen (se pueden incluir tanto caracteres de espacio como saltos de línea) el analizador de HTML reduce cada uno a un único espacio, al representar el código.
- ¿Entonces por qué utilizar espacios en blanco? La respuesta es **legibilidad**: es mucho más fácil entender el código si está bien formateado en lugar de estar desordenado.
- Observemos el siguiente ejemplo:

```
<p>En un lugar de la Mancha ...</p>
<p>En      un      lugar
      de  la      Mancha ...</p>
```

- Los dos párrafos mostrarán exactamente lo mismo.
- En nuestro HTML cada elemento anidado se sangra con dos espacios más que el que está fuera, pero de cada desarrollador.

ATRIBUTOS

- Los elementos pueden tener atributos que tienen el siguiente aspecto:



```
<p class="editor-note">Mi gato es muy gruñon</p>
```

- Los atributos **contienen información extra**. Es este caso, el atributo class asigna permite indicar la clase a la que pertenece el elemento. Muy útil, como veremos en el siguiente tema, al combinarlo con hojas de estilo.
- Un atributo **debe tener**:
 - Un espacio entre él y el nombre del elemento.
 - El nombre del atributo, seguido por un signo igual.
 - El valor del atributo entrecomillado (habitualmente comillas dobles, pero también se pueden utilizar simples. Eso si, no se pueden combinar en un mismo elemento).

COMENTARIOS

- En HTML existen mecanismos para escribir comentarios en el código.
- Los comentarios son **ignorados por el navegador** y son invisibles para el usuario. Su propósito es permitir **comentar el código** para aclarar su funcionamiento, explicar lo que hacen las diferentes partes del código, etc. Esto es muy útil como recordatorio del trabajo realizado si vuelve a ver código en el que no ha trabajado durante meses o si entrega su código a otra persona para que trabaje en él.
- Para convertir una sección de contenido dentro de su archivo HTML en un comentario, debe envolverlo en los marcadores especiales **<!-- y -->**, por ejemplo:

```
<p>Esto no está dentro de un comentario</p>
<!-- <p>Esto sí!</p> -->
```

METADATOS: EL ELEMENTO <META>

Los Metadatos son datos que describen datos, y HTML tiene una forma "oficial" de agregar metadatos a un documento, el elemento **<meta>**.

Atributos más comunes:

- **charset**: permite indicar la codificación de caracteres usada en nuestra página. `<meta charset="utf-8">`
- Muchos elementos **<meta>** incluyen los atributos **name** y **content**:
 - **name** permite especificar el tipo de elemento
 - **content** permite especificar el contenido
 - Para indicar el autor de la página: `<meta name="author" content="Jordi Benimeli">`
 - Para indicar una descripción de la página: `<meta name="description" content="Página dedicada a bla bla ...>`
 - Para indicar palabras clave (actualmente la mayoría de buscadores lo ignoran): `<meta name="keywords" content="enseñanza, aprendizaje, formación, informática, cursos">`

ICONO PERSONALIZADO

- Para mejorar la experiencia de nuestros usuarios podemos incluir un icono que se mostrará en la pestaña y en la barra de búsqueda.
- Para ello debemos crear un archivo llamado **favicon.icon** que contenga un ícono de 16 x 16 pixel y guardarlo en el mismo directorio que la página index.html de nuestro sitio.
- Para indicarle a nuestra página que lo incluya, loharemos dentro de la etiqueta **<head>** de la siguiente forma:
`<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">`

ELEMENTOS DE BLOQUE

- Cuerpo del documento (body)
- Títulos (h1, h2... h6)
- Párrafos (p)
- Salto de línea (br)
- Listas:
 - Sin relación de orden (ul, li)
 - Con relación de orden (ol, li)
 - De definición (dl, dt, dd)
- Barra o separador horizontal (hr)
- Preformato (pre)
- Citas de bloque (blockquote)
- División (div)
- Tablas (table)
- Formularios (form)

ELEMENTOS DE LÍNEA

- Elementos lógicos (em, strong, cite, code...)
- Enlaces (a)
- Imágenes (img)
- Contenedores en línea (span)
- ~~Elementos físicos (sub, sup, i, b, u) (no recomendables)~~

TÍTULOS, PÁRRAFOS Y SALTOS DE LÍNEA

- Cabeceras o títulos (los tamaños van desde 1 hasta 6):
 - <h1>Título 1 </h1>
 - <h2>Título 2</h2>
 - <h3>Título 3</h3>
 - <h4>Título 4</h4>
 - <h5>Título 5</h5>
 - <h6>Título 6</h6>
- Párrafos
 - <p>Esto es un párrafo</p>
 - <p>Esto es otro párrafo</p>
- Saltos de línea
 -



¿POR QUÉ NECESITAMOS UNA BUENA ESTRUCTURA?

Diseñar las páginas web con una buena estructura nos ayudará en los siguientes aspectos:

- **Atraer y mantener la atención:** los usuarios tienden a escanear rápidamente el contenido de una página para encontrar el contenido relevante. Si no pueden ver nada útil en segundos, es probable que se frustren y se vayan a otra página.
- **SEO:** los motores de búsqueda indexan las páginas considerando los títulos como palabras clave importantes.
- **Accesibilidad:** las personas con discapacidad visual severa habitualmente no leen las páginas, en lugar de ello, las escuchan con screen readers. Este tipo de software suele proporcionar un esquema del documento para permitir a los usuarios encontrar la información más rápidamente.
- **Aplicar CSS y JS:** con una buena estructura es mucho más fácil aplicar un diseño bonito y agradable.

ÉNFASIS Y DESTACADO

- La etiqueta **** sirve para indicar que queremos poner **énfasis** sobre un determinado contenido. Por ejemplo:

Lo hago así porque quiero

La forma habitual en que lo muestran los navegadores es con cursiva, pero no debemos confundirla con la etiqueta física **<i>** que ya hemos comentado que no deberíamos utilizar ya que indica un determinado formato específico.

- La etiqueta **** sirve para **destacar** un determinado contenido por encima del resto. Por ejemplo:

En 100 metros gire a la derecha

RUTAS

- **Ruta:** (en inglés *path*) es la forma de referenciar un recurso informático (habitualmente una archivo o directorio) en un sistema de archivos de un sistema operativo determinado.
- Una ruta señala la **localización exacta de un archivo o directorio** mediante una cadena de caracteres concreta. Ésta puede ser de diversas formas dependiendo del sistema operativo y del sistema de archivos en cuestión.
- En líneas generales se compondrá de los nombres de los directorios que conforman el camino hasta nuestro archivo o directorio a lo largo del árbol de directorios, y finalmente estará el nombre del archivo o directorio que se quiere referenciar.
- Estos nombres estarán separados por un carácter delimitador que usualmente será "\" en sistemas MS-DOS/Windows (Microsoft) o "/" en sistemas UNIX/Linux.

RUTAS

Una ruta estará formada por:

[*Unidad (sólo windows)*] + { delimitador + *carpeta* } + [*archivo*]

- [] son campos opcionales
- { } conjunto/repetición de campos

Ejemplos:

- En Windows
 - C:\Archivos de Programa\Steam\Steam.exe
 - C:\Windows\notepad.exe
- En Linux
 - /home/1smx/desktop
 - /home/2smx/documents/trabajo.doc

RUTAS

- **Ruta absoluta:** señala la ubicación de un archivo o directorio desde el directorio **raíz** del sistema de archivos. Por ejemplo:
 - **/home/dir1/foto.jpg** señala la ubicación de foto.jpg desde la raíz del sistema de archivos.
 - **C:\Archivos de programa\Steam\Steam.exe** señala la ubicación del archivo Steam.exe desde la raíz del sistema de archivos.
- **Ruta relativa:** señala la ubicación de un archivo o directorio **a partir de la posición actual** del sistema operativo en el sistema de archivos. Por ejemplo:
 - **dir1/foto.jpg** señala al archivo foto.jpg dentro del directorio dir1 en la ubicación actual.
 - **Steam\steam.exe** señala al archivo Steam.exe dentro del directorio Steam en la ubicación actual.

RUTAS

Consideraciones:

- Siempre que sea posible utilizaremos **rutas relativas**, ya que nos permitirá poder **alojar nuestra página web en cualquier servidor web**, sin depender del directorio que ocupe en él.
- Esto es válido para cualquier tipo archivo (imágenes, videos, sonidos, etc...) o carpeta.

HIPERENLACES

- Un **hiperenlace**, también llamado *enlace*, *link* o *vínculo*, es un elemento que permite **hacer referencia a otro recurso**, por ejemplo, otra página web.
- Estructura general de un enlace: `yyy`
 - xxx: destino del enlace (Nombre del recurso)
 - yyy: texto que aparecerá hiperenlazado
- Tipos de hiperenlaces:
 - Hiperenlaces a otra página
 - Hiperenlaces a un punto dentro de una página
 - Hiperenlaces a una página fuera de nuestro sistema
 - Hiperenlaces a una dirección e-mail

HIPERENLACE A OTRA PÁGINA

- En este tipo de hiperenlaces el destino se encuentra en nuestro portal web, habitualmente será otro documento html de nuestro portal.
- La forma de referenciarlo es la siguiente:
 - **Ejemplo 2**
 - Al tratarse de un destino dentro de nuestro sistema, el atributo href contendrá una **ruta relativa**.
- Otro ejemplo que nos lleva a una subcarpeta dentro de la carpeta actual:
 - **Producto 1**

HIPERENLACES A UN PUNTO DENTRO DE UNA PÁGINA

- En este tipo de hiperenlaces hacemos referencia a un punto concreto dentro de una página.
- Para ello debemos marcar dos puntos:
 - En primer lugar debemos **identificar el destino**. Para ello asignaremos el atributo **id** al elemento hacia el que queremos apuntar.
`<h3 id="email">j.benimelialemany@edu.gva.es</h3>`
 - Para hacer referencia a este id concreto, lo añadiremos al final de la url:
`email`
Si el destino se encontrara en la página actual escribiríamos simplemente:
`email`

HIPERENLACES A UNA PÁGINA FUERA DE NUESTRO SISTEMA

- En este tipo de hiperenlaces hacemos referencia un **recurso que no se encuentra hospedado en nuestro portal web**.
- En este caso, debemos indicar la dirección completa de la página web, incluyendo el protocolo http.
- Por ejemplo:
`Universitat Politècnica de València`

HIPERENLACES A UNA DIRECCIÓN EMAIL

- En este tipo de hiperenlaces hacemos referencia a una dirección de correo electrónico.
- Al pulsar sobre uno de estos hiperenlaces, el navegador abrirá la aplicación local para enviar correo electrónico con la dirección destino indicada.
- Para ello debemos indicar como valor de href “**mailto:direccion**”
- Por ejemplo:

```
<a href="mailto:ejemplo@gmail.com">Contacto</a>
```

HIPERENLACES – ATRIBUTOS ÚTILES

- Atributo de descarga **download**:

Si queremos hacer referencia a una descarga en lugar de a algo abierto por el navegador, podemos utilizar el atributo download indicando el nombre del archivo a descargar. Por ejemplo:

```
<a href="https://download.mozilla.org/?product=firefox"  
download="firefox-installer.exe">Descargar Firefox</a>
```

- Atributo de ventana objetivo **target**:

Si queremos forzar a que **el link destino se cargue en una nueva ventana o pestaña** en lugar de sustituir a la actual, podemos utilizar el atributo target con el valor “_blank”. Por ejemplo:

```
<a href="http://www.nasa.gov" target="_blank">Página de la NASA</a>
```

LISTAS

- Las listas están en todos los aspectos de nuestra vida, desde la lista de compra a la lista de direcciones que sigues inconscientemente para llegar a tu casa cada día.
- Las listas están **en todos lados en la Web** también y tenemos tres tipos diferentes para preocuparnos:
 - Listas sin relación de orden, también llamadas desordenadas por la traducción “unorder list” o listas de viñetas.
 - Listas con relación de orden, también llamadas ordenadas por la traducción “order list” o numeradas.
 - Listas de definiciones.

LISTAS SIN RELACIÓN DE ORDEN

- Son listas en las que no existe un orden intrínseco entre los elementos que forman parte de la lista.
- Por ejemplo, en una lista de la compra, no importa cuál va antes o después, queremos comprarlos todos.
- Para definir una lista de este tipo se utiliza la etiqueta **** “unorder list”.
- Cada uno de los elementos que componen la lista irá contenido en etiquetas ****.
- Por ejemplo:

```
<ul>
  <li>leche</li>
  <li>arroz</li>
  <li>pan</li>
  <li>aceite</li>
</ul>
```



- leche
- arroz
- pan
- aceite

LISTAS CON RELACIÓN DE ORDEN

- Son listas en las que sí importa el orden en las que sean enumeradas, como por ejemplo las instrucciones para llegar a un sitio.
- Para definir una lista de este tipo se utiliza la etiqueta **** “order list”.
- Cada uno de los elementos que componen la lista irá contenido en etiquetas ****
- Por ejemplo:

```
<ol>
  <li>Conduzca hasta el final de la calle</li>
  <li>Gire a la derecha</li>
  <li>En la rotonda salga en la segunda salida</li>
  <li>Continue 100 metros</li>
  <li>El colegio está a su derecha</li>
</ol>
```



1. Conduzca hasta el final de la calle
2. Gire a la derecha
3. En la rotonda salga en la segunda salida
4. Continue 100 metros
5. El colegio está a su derecha

LISTAS DE DEFINICIONES

- Permiten crear listas de definición indicando el término y su definición.
- Para crear una lista de definición utilizamos la etiqueta **<dl>** “definition list”.
- Para crear cada uno de los términos utilizamos la etiqueta **<dt>** “definition term”
- Para crear las definiciones de cada término utilizamos la etiqueta **<dd>**
- Ejemplo:

```
<dl>
  <dt>Informática</dt>
  <dd>Conjunto de conocimientos científicos y
      técnicas que hacen posible el tratamiento
      automático de la información por medio
      de computadoras</dd>
  <dt>Ordenador</dt>
  <dd>Máquina electrónica que, mediante
      determinados programas, permite
      almacenar y tratar información, y resolver
      problemas de diversa índole</dd>
</dl>
```



Informática

Conjunto de conocimientos científicos y técnicas que hacen posible el tratamiento automático de la información por medio de computadoras

Ordenador

Máquina electrónica que, mediante determinados programas, permite almacenar y tratar información, y resolver problemas de diversa índole

LISTAS ANIDADAS

- Todas las listas que hemos visto pueden ser anidadas, de forma que un elemento de una lista, puede a su vez, contener otra lista.

- Por ejemplo:

```
<ul>
  <li>Casa</li>
    <ol>
      <li>Gire a la derecha</li>
      <li>En la rotonda salga en la segunda salida</li>
    </ol>
  <li>Trabajo</li>
    <ol>
      <li>Continúe 100 metros</li>
      <li>Gire a la izquierda</li>
    </ol>
</ul>
```



- Casa
 1. Gire a la derecha
 2. En la rotonda salga en la segunda salida
- Trabajo
 1. Continúe 100 metros
 2. Gire a la izquierda

TABLAS

- Las tablas **<table></table>** permiten mostrar datos tabulados. Históricamente también se han usado también para organizar contenido, pero actualmente es considerado un grave error, por lo que su uso debería quedar restringido a mostrar datos en forma tabulada.
- Funcionamiento:
 - Definimos las **filas** con las etiquetas **<tr></tr>** (“table row”)
 - Definimos cada **columna** de la fila con etiquetas **<td></td>** (“table data”). **El número de columnas debe coincidir en todas las filas.**

TABLAS

No es recomendable utilizar el atributo border. Lo utilizamos aquí solo para ver los límites de la tabla, pero es mejor hacerlo desde hojas de estilo

```
<table border="1">
```

```
    <tr>
```

```
        <td> Texto1 </td>
```

```
        <td> Texto2 </td>
```

```
        <td> Texto3 </td>
```

```
    </tr>
```

```
    <tr>
```

```
        <td> aa </td>
```

```
        <td> bb </td>
```

```
        <td> cc </td>
```

```
    </tr>
```

```
    <tr>
```

```
        <td> dd </td>
```

```
        <td> ee </td>
```

```
        <td> ff </td>
```

```
    </tr>
```

```
</table>
```

Texto1	Texto2	Texto3
aa	bb	cc
dd	ee	ff

TITULAR DE LA TABLA

Para indicar el título de la tabla podemos utilizar la etiqueta
<caption>Título de la tabla</caption>.

Título de la tabla

Texto1	Texto2	Texto3
aa	bb	cc
dd	ee	ff

CELDA DE CABECERA DE LA TABLA

- En ocasiones puede ser interesante incluir **encabezados de columna** en la tabla.
- Para ellos podemos utilizar la etiqueta **<th></th>**.

Cabecera1	Cabecera2	Cabecera3
aa	bb	cc
dd	ee	ff

FUSIONAR COLUMNAS

- Podemos unir varias columnas de una tabla mediante el atributo **colspan** de la etiqueta **td**. El **valor** que proporcionemos a colspan será el **número de columnas fusionadas**.
- Es importante recordar que **la suma de columnas debe coincidir en todas las filas**.

A		B
C	D	E
F	G	

```
<table>
<tr><td colspan="2">A</td><td>B</td></tr>
<tr><td>C</td><td>D</td><td>E</td></tr>
<tr><td>F</td><td colspan="2">G</td></tr>
</table>
```

FUSIONAR FILAS

- Podemos unir varias filas de una tabla mediante el atributo **rowspan** de la etiqueta **td**. El **valor** que proporcionemos a rowspan será el **número de filas fusionadas**.
- La columna que tiene el atributo rowspan “consume” una columna de la fila o filas posteriores. El nº de columnas debe ser igual para todas la filas.

	B	C
A	D	E
	F	G

```
<table>
<tr><td rowspan="3">A</td><td>B</td><td>C</td></tr>
<tr><td>D</td><td>E</td></tr>
<tr><td>F</td><td>G</td></tr>
</table>
```

FUSIÓN DE FILAS Y COLUMNAS (COMBINADAS)

- Podemos combinar **colspan** y **rowspan** para obtener cualquier tipo de tabla.
- En este ejemplo la tabla tiene 3 filas y 3 columnas. El número de columnas de una tabla será siempre el de la fila con más columnas.

A	B	
	C	
	D	E

```
<table>
<tr><td rowspan="3">A</td><td colspan="2">B</td></tr>
<tr><td colspan="2">C</td></tr>
<tr><td>D</td><td>E</td></tr>
</table>
```

IMÁGENES

- Podemos insertar imágenes en nuestros documentos mediante la etiqueta ``.
- Se trata de una **etiqueta void** por tanto tiene autocierre.
- El atributo **src** permite indicar la ruta al archivo de imagen.
- El atributo **alt** permite indicar una descripción de la imagen para que en el caso de que no se cargue, se disponga de una breve descripción. Muy importante para que nuestras páginas sean accesibles.
- Ejemplo:
 - ``

FORMULARIOS

- Los formularios son posiblemente la **herramienta más utilizada en Internet para obtener datos e información** acerca de la gente que navega nuestro sitio.
- La idea de los formularios es recolectar información online en la interacción con el usuario y luego ejecutar una determinada acción con la misma. Por ejemplo, quizás el caso más utilizado, un formulario de venta que el usuario completa y luego es enviado vía email al vendedor en forma encriptada.
- Para construir formularios utilizaremos las siguientes etiquetas básicas **<form>**, **<label>**, **<input>**, **<textarea>**, **<select>**, **<option>**, **<button>**
- Para procesar la información que se recibe de un formulario es necesario poseer un **lenguaje de script en el servidor**. Con HTML sólo podemos definir la interfaz de comunicación.

FORMULARIOS – ETIQUETA FORM

- Todo formulario debe estar encerrado entre el par de etiquetas **<form>** y **</form>**, y debe ser ubicado en el cuerpo de cualquier documento HTML, es decir, entre el par de etiquetas **<body>** y **</body>**. Esta etiqueta **<form>** presenta **tres atributos posibles**:
 - **action**: valor de este parámetro es la URL del programa o guión en el servidor Web utilizado para procesar la información recolectada. **Es de uso obligatorio ya que establecen dónde enviar la información.**
 - **method**: puede asumir el valor GET o el valor POST, y definen la manera en la cual los datos son transferidos al servidor. **Es de uso obligatorio ya que establecen cómo enviar la información.**
 - **enctype**: reservado para indicar como debería ser codificada la información del formulario. Útil cuando queremos enviar archivos. **Es optativo.**

FORMULARIOS – ETIQUETA FORM

Todos los elementos que pertenecen a un formulario irán entre la apertura y cierre del elemento **form** de la siguiente manera:

```
<form action="archivo_destino.php" method="get">
```

Aquí irán etiquetas de componentes para entrada de datos, como por ejemplo input, select, textarea, etc.

```
</form>
```

FORMULARIOS – LA ETIQUETA INPUT

- La etiqueta **<input>** es la segunda etiqueta más importante de los formularios. Se la puede definir como una etiqueta multifunción, ya que con la misma podemos crear "push buttons", "radio buttons", "check boxes", cuadros de texto, y un largo etc.
- Los atributos comunes más importantes son:
 - autocomplete, autofocus, disabled, form, id, list, name, readonly, required, tabindex, type, value
- Los atributos que siempre figurarán en cualquier etiqueta input son **name** y **type**.

FORMULARIOS – ETIQUETA INPUT. ATRIBUTOS COMUNES

- **autocomplete**, cadena de caracteres indicado el tipo de autocompletado.
- **autofocus**, valor booleano que indica que el input recibirá el foco cuando el formulario sea presentado al usuario.
- **disabled**, valor booleano que indica si el input está desactivado o no.
- **form**, indica el id del formulario al que pertenece el input. Si no se especifica, pertenecerá al más cercano que contenga a la etiqueta input.
- **id**, identificador único del input.
- **list**, permite indicar el id del <datalist> que proporciona la lista de valores sugeridos para el input.
- **name**, asocia un nombre al input que permite identificar el valor enviado.
- **readonly**, valor booleano que indica si el componente es solo de lectura.
- **required**, valor booleano que indica si el input debe tener un valor antes de poder ser enviado el formulario.
- **tabindex**, valor numérico que indica el orden en el que input recibirá el foco cuando el usuario pulse la tecla Tab.
- **type**, cuyos valores más comunes son text, password, checkbox, radio, hidden, file, reset, submit, image, color, date, datetime-local, email, month, number, range, search, tel, time, url, week; de acuerdo al tipo de elemento que querremos representar. Más tarde veremos algunos ejemplos.
- **value**, valor por defecto que tomará la variable que se creará al enviar el formulario con nombre igual al valor del atributo name.

FORMULARIOS –ETIQUETA LABEL

- Representa una etiqueta para un elemento en una interfaz de usuario.
- Este puede estar asociado con un control ya sea mediante la utilización del atributo **for**, o ubicando el control dentro del elemento **label**.

```
<label for="Name">Click me</label>
```

```
<input type="text" id="Name" name="Name" />
```

```
<label>Click me <input type="text" id="Name" name="Name" /></label>
```

FORMULARIOS – INPUT TYPE="TEXT"

- Permite la introducción de texto en una caja de texto.
- Al ser enviado el formulario se creará una **variable con el nombre** igual al valor del atributo **name** y con el valor introducido en el campo.
- Podemos asignar un **valor por defecto** al campo añadiendo el atributo **value**.

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

Nombre:

FORMULARIOS – INPUT TYPE=’PASSWORD’

- Permite la introducción de texto en una línea **ocultando los caracteres tecleados**.
- Al ser enviado el formulario se creará una **variable con el nombre** igual al valor del atributo **name** y con el valor introducido en el campo. (=text)
- Podemos asignar un **valor por defecto** al campo añadiendo el atributo **value**. (=text)
`<label for="inputPassword">Contraseña:</label>
<input id="inputPassword" name="passwd" type="password" />`

Password:

FORMULARIOS – INPUT TYPE="NUMBER"

- Permite la introducción de valores numéricos.
- Mediante los atributos **min** y **max** podemos establecer los valores mínimos y máximos que aceptará.

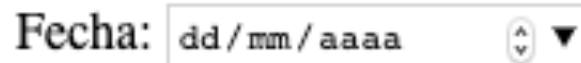
```
<label for="edad">Edad: </label>
<input id="edad" type="number" name="edad" />
```

Edad:

FORMULARIOS – INPUT TYPE="DATE"

- Permite la introducción de fechas.
- Se trata de una etiqueta relativamente nueva introducida en html 5 y por tanto algunos navegadores no la soportan (la interpretan como type="text").

```
<label for="idfecha">Fecha: </label>
<input id="idfecha" type="date" name="fecha" />
```



- Existen la variante type="datetime-local" que permite introducir fecha y hora.

FORMULARIOS – INPUT TYPE="CHECKBOX"

- Permite la introducción de **selecciones múltiples**. Suele utilizarse cuando se le quiere ofrecer al usuario que elija entre una o varias opciones.
- Al ser enviado el formulario si tiene el atributo **checked**, se creará una variable con el nombre igual al valor del atributo **name**.
- Podemos hacer que un checkbox esté seleccionado por defecto utilizando el atributo **checked="checked"**.

```
<input id="checkRemember" type="checkbox" name="remember" />
<label for="checkRemember">Recordar contraseña</label>
```

Recordar contraseña

FORMULARIOS – INPUT TYPE="RADIO"

- Permite la introducción de un único dato procedente de **múltiples opciones** que son **mutuamente excluyentes entre sí**. Para que los radios sean mutuamente excluyentes entre si **deben tener el mismo nombre** (atributo name) pero con distintos valores (atributo value) por defecto.

```
<input id="rh" type="radio" name="sexo" value="h" />
```

```
<label for="rh">Hombre</label>
```

```
<input id="rm" type="radio" name="sexo" value="m" />
```

```
<label for="rm">Mujer</label>
```

Hombre Mujer

FORMULARIOS – SELECT

- La etiqueta **select** crea un control mediante el cual los usuarios pueden seleccionar de una lista de opciones.
- Está especialmente indicado cuando queremos **limitar la entrada del usuario**.
- Para **cada una de las opciones** que queramos que sea visible en la lista utilizaremos la etiqueta
<option value="valor">Texto</option>.
- El valor (value) que tenga cada una de las opciones es importante ya que es el valor que será enviado de la forma **name_del_select=valor_del_option**
- Es importante que el número de elementos de una etiqueta <select>, es decir, el número de <option>, no sea demasiado elevado.
- Otra característica importante de esta etiqueta es la posibilidad de crear grupos de opciones dentro de un mismo <select> mediante la etiqueta
<optgroup label="nombre grupo">...</optgroup>.

```
<select name="ciudades">
  <optgroup label="Comunidad Valenciana">
    <option value="ali">Alicante</option>
    <option value="cas">Castell&oacute;n</option>
    <option value="val">Valencia</option>
  </optgroup>
  <optgroup label="Comunidad de Madrid">
    <option value="mad">Madrid</option>
    <option value="get">Getafe</option>
    <option value="alc">Alcorc&oacute;n</option>
  </optgroup>
</select>
```

NOTA:

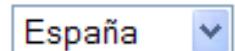
Observar la diferencia entre el texto que aparece en cada una de las opciones de la lista y el valor que realmente será enviado.



FORMULARIOS – SELECT CON OPCIÓN POR DEFECTO

- La opción seleccionada por defecto es el primer `<option>` que hemos introducido en el `<select>`, pero en ocasiones puede ser interesante cambiar esta opción por defecto para facilitarle la tarea al visitante de nuestra web (porque es la opción más probable de ser escogida).
- Podemos conseguirlo añadiendo a la opción (`option`) el atributo **selected="selected"**.

```
<select name="pais">
  <option value="ar">Argentina</option>
  <option value="co">Colombia</option>
  <option value="es" selected="selected">Espa&ntilde;a</option>
  <option value="mx">M&eacute;xico</option>
  <option value="ve">Venezuela</option>
</select>
```



FORMULARIOS – SELECT

- Otra opción que puede ser interesante es poder mostrar varios elementos de una lista a la vez.
- Podemos conseguirlo añadiendo a la etiqueta <select> el atributo **size="n"** donde n será el número de filas que deseamos que se muestren por defecto.

```
<select name="pais" size="5">
  <option value="ar">Argentina</option>
  <option value="co">Colombia</option>
  <option value="es">Espa&ntilde;a</option>
  <option value="mx">M&eacute;xico</option>
  <option value="ve">Venezuela</option>
</select>
```



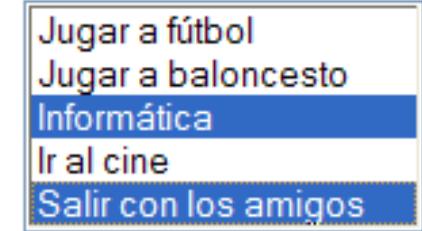
FORMULARIOS – SELECT CON SELECCIÓN MÚLTIPLE

- Veamos como podemos seleccionar varios elementos de una lista a la vez. Esto es especialmente interesante cuando las opciones que podemos seleccionar no son mutuamente excluyentes entre si.
- Para activar la selección múltiple hay que añadir a la etiqueta select el atributo **multiple="multiple"**.
- El atributo **multiple="multiple"** muestra por defecto las primeras 4 opciones de la lista. Si nuestra lista tiene más de 4 elementos podemos añadir el atributo size para mostrarlas todas.
- Para seleccionar más de una opción de la lista hay que utilizar la tecla **Ctrl+Pulsación izquierda de ratón**.

FORMULARIOS – SELECT CON SELECCIÓN MÚLTIPLE

Ejemplo, campo de selección de aficiones:

```
<select name="aficiones" multiple="multiple" size="5">
  <option value="jf">Jugar a f&uacute;tbol</option>
  <option value="jb">Jugar a baloncesto</option>
  <option value="in">Inform&aacute;tica</option>
  <option value="ic">Ir al cine</option>
  <option value="sa">Salir con los amigos</option>
</select>
```



FORMULARIOS – INPUT TYPE="RANGE"

- Permite la introducción de valores numéricos que deben estar entre un mínimo y un máximo pero saber su valor exacto no es realmente importante.
- Suele representarse visualmente como una barra deslizante.
- Mediante los atributos **min** y **max** podemos establecer los valores mínimos y máximos que aceptará.
- Mediante el atributo **step** podemos indicar como variará su valor al mover un paso la barra deslizante.

```
<label for="vol">Volumen</label>
<input id="vol" type="range" name="volumen" min="0" max="100"
value="60" step="10" />
```



FORMULARIOS – INPUT TYPE="HIDDEN"

- Este tipo de input no es visible desde el navegador, no produce ningún espacio y no puede ser modificado por el usuario.
- Suele ser utilizado **para enviar parámetros de configuración que no tiene porque ver el usuario.**
- Al ser un campo que no se puede modificar, un input de tipo hidden **siempre deberá tener un valor por defecto.**

```
<input type="hidden" name="idioma" value="es" />
```

FORMULARIOS – INPUT TYPE=”FILE”

- Permite subir ficheros desde un cliente al servidor.
- Un ejemplo claro de este tipo de campo es el que utilizáis para subir las prácticas (vuestra navegador actúa como cliente y AULES como servidor).
- Para que el archivo pueda ser enviado el formulario debe utilizar el **método post** y el atributo **enctype** de la etiqueta **form** cobra sentido.
- Hasta ahora no era necesario modificar éste atributo, pero con el envío de ficheros, **enctype** debe tener el valor “**multipart/form-data**”.
- Por seguridad se puede definir un campo oculto para limitar el tamaño del fichero que el cliente puede enviar. Este campo oculto debe tener como nombre **MAX_FILE_SIZE** y el valor del campo será el **tamaño máximo en bytes**.

FORMULARIOS – INPUT TYPE=”FILE”

- Al crear un campo de tipo file aparece automáticamente el botón Examinar ... que permite buscar en el sistema de ficheros.

- Ejemplo:

```
<form action="destino.php" method="post" enctype="multipart/form-data">  
...  
    <label for="idfichero">Fichero: </label>  
    <input id="idfichero" type="file" name="fichero" />  
    <input type="hidden" name="MAX_FILE_SIZE" value="1024000" />  
...  
</form>
```

Fichero: Ningún archivo seleccionado

FORMULARIOS – INPUT TYPE=’RESET’

- Este campo crea un botón para reiniciar un formulario a los valores que cada campo tiene por defecto.
- El valor que le proporcionemos en el atributo value será el texto que aparecerá escrito sobre el botón.

```
<input type="reset" value="Limpiar datos" name="breiniciar" />
```

Limpiar datos

FORMULARIOS – INPUT TYPE="SUBMIT"

- Crea un botón para enviar un formulario al servidor.
- Al enviar un formulario pulsando sobre el botón se crea una variable con el nombre (name) y valor (value).
- También podemos enviar un formulario pulsando la tecla intro estando dentro de un input tipo text. En este caso la variable con el nombre del botón no será creada.

```
<input type="submit" name="enviar" value="Enviar Datos" />
```

Enviar Datos

FORMULARIOS – BUTTON

- Representa un botón sobre el cuál se puede hacer click.
- Este elemento puede ser usado en cualquier parte del documento y por tanto no está restringido solo a formularios.
- Mediante el atributo **type** podemos indicar la funcionalidad del botón:
 - **submit** para enviar formularios.
 - **reset** para borrar el contenido de los controles de un formulario.
 - **button** no tiene un comportamiento predefinido. Útiles en combinación el atributo **onclick** que permite definir acciones a realizar mediante el lenguaje JavaScript.

```
<button type="button" onclick="alert('Hola');">Saluda</button>
```

Saluda

FORMULARIOS – TEXTAREA

- Otro elemento que suele utilizarse para enviar información mediante formularios es el área de texto.
- Son similares a las cajas de texto `input type="text"` pero se diferencian de estas en que el área de texto suele utilizarse para enviar grandes cantidades de texto.
- Para crear un área de texto utilizaremos la etiqueta:
`<textarea name="nombre">...Contenido...</textarea>`.
- Podemos cambiar el tamaño por defecto mediante los atributos **rows** (número de filas de texto) y **cols** (número de caracteres por fila).

FORMULARIOS – TEXTAREA

- Ejemplo: Formulario para enviar un correo electrónico. El asunto en una caja de texto y el cuerpo del mensaje en una área de texto.

```
<label for="asunto">Asunto: </label>
<input id="asunto" type="text" name="asunto" size="50" /><br />
<label for="mensaje">Mensaje: </label><br />
<textarea id="mensaje" name="mensaje" rows="6" cols="60"></textarea><br />
<input type="submit" value="Enviar Mensaje" />
```

The image shows a screenshot of a web-based form. At the top, there is a label "Asunto:" followed by a text input field. Below it, there is a label "Mensajes:" followed by a large text area for entering the message content. At the bottom of the form is a button labeled "Enviar Mensaje".