



UT 2:

Elaboración de páginas web con lenguajes de marcas

Aplicaciones Web

Grado Medio - Sistemas Microinformáticos y Redes

Introducción a HTML

1 - Introducción

En este tema, vamos a empezar a trabajar con el lenguaje de marcado **HTML (HyperText Markup Language)** para crear páginas web.

A modo de contexto, recordemos el lugar de estos en Internet:

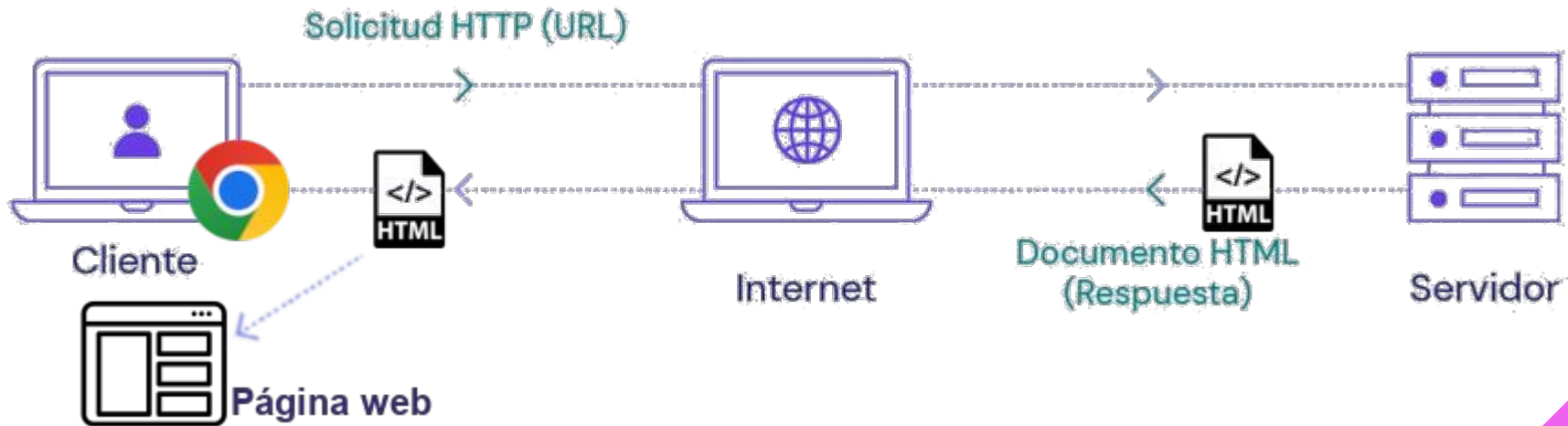
- **Internet es una red** con una enorme cantidad de equipos conectados.
- Una **parte de internet es la WWW** (World Wide Web), donde se comparte una ingente cantidad de información.
- Esta **información** se estructura en la forma de **páginas web**.
- Estas **páginas** se escriben en **lenguaje HTML** (en su forma más simple).

1 - Introducción

Estos **archivos .html** suelen estar almacenados en un **servidor web**.

Al acceder con un **nombre de dominio** o IP, **solicitamos la página web al servidor**. Lo que estamos pidiendo es el **.html** de la página.

El **archivo .html** (en texto) **lo traduce el navegador a la vista de página**.



1 - Introducción

La característica principal de las páginas web es que son **hipertextos**.

Con esto nos referimos a textos que enlazan a otros, en este caso, las páginas web pueden **enlazar a otras**, de modo que queden conectadas y accesibles entre sí. Algunos términos a tener claros son:

- **Hipertexto**. Documento basado en texto en el que mediante vínculos, se puede acceder a otros documento similares. P. ej. archivos .html.
- **Hipervínculo** entre hipertextos. Los links/enlaces que conocemos.
- **HTTP**. Protocolo de Transmisión de Hipertextos.
- **HTML**. Lenguaje de Marcado de Hipertextos.

1 - Introducción

Siguiendo estos conceptos, podemos hacernos una idea de lo que es **.html**. Es un **lenguaje de marcado** en el que se escriben **hipertextos**, es decir, **páginas web**.

Pero, ¿qué es un lenguaje **de marcado**? ¿Qué son esas **marcas**?

Las **marcas** son una especie de **etiquetas** que acompañan a los textos e indican propiedades de estos. Si las escribimos siguiendo las reglas de **.html**, el navegador puede reconocerlas y mostrar la información.

P. ej. una etiqueta **<h1> Texto entre etiquetas </h1>** indica que el texto de en medio es el header o encabezado de la página.

1 - Introducción

Así, un **lenguaje de marcas** nos permite añadir estas etiquetas a los textos de la página para que el navegador las reconozca como textos, tablas, imágenes, enlaces, vídeos...

Aunque ambos se llamen **lenguajes**, los **lenguajes de marcas no son lenguajes de programación**, porque no definen sentencias a ejecutar. Sólo añaden información al texto, que traducimos a las páginas web.

```
<div  
  <h2>This is a heading in a div element</h2>  
  <p>This is some text in a div element.</p>  
</div>
```

Lenguaje de marcas

```
hello.py > ...  
1  msg = "Hello World"  
2  print(msg)
```

Lenguaje de programación

1 - Introducción

Los lenguajes de marcas aparecen a partir de los años 60 con IBM como propuesta de un **lenguaje estándar** para las marcas que describen el contenido en **diferentes documentos**.

Uno de los primeros a destacar fue **GML (Generalized Markup Language)** que permitía transmitir archivos entre sistemas sin perder información.

Después apareció **SGML (Standard Generalized Markup Language)** como estándar para este lenguaje.

Las normas estandarizadas por SGML todavía se aplican en los lenguajes de marcas actuales, entre los que destacamos **HTML** y **XML**.

2 - Fundamentos de HTML

Visto esto, **HTML** es un **lenguaje** que nos va a permitir escribir un **archivo de texto** en un formato que permita a los **navegadores web** convertirlo en una **página web** y **mostrarla al usuario**. Por esto los archivos .html:

- Se pueden manipular con **cualquier editor de texto**.
- **No contienen código**, sólo **textos con marcas**.
- Son **compatibles con cualquier navegador**.
- Son **accesibles en cualquier web** (guardar como).
- **No hace falta compilarlo**.



2 - Fundamentos de HTML

Existen **programas específicos** para **facilitar la creación de archivos .html**:

- **Editores de texto con asistentes**, plantillas de código, autocompletado...
P. ej. Atom, Sublime Text, Brackets, Visual Studio Code...
- Programas que pueden **exportar documentos a .html**. P. ej. Word, Writer...
- **Programas WYSIWIG (What You See Is What You Get)**, que permiten maquetar páginas de forma visual (con botones, cajas y similares) sin escribir código HTML. P.ej. Dreamweaver, Google Web Designer...

2 - Fundamentos de HTML

El **lenguaje HTML** establece unas **reglas a la hora de redactar archivos**.

Primero, hay que entender cómo funcionan las **etiquetas**:

- Se representan con **una palabra específica entre <>**. P.ej. `<body>`
- Las que tienen contenido (de bloque) se **abren y se cierran**.

P.ej. `<body>` y `</body>` Entre la apertura y cierre, se incluye el contenido.

Por ejemplo: `<body> Texto afectado por la etiqueta cuerpo </body>`

Este es el uso más básico, aunque luego veremos etiquetas con atributos.

P. ej. una foto y su tamaño: ``

2 - Fundamentos de HTML



Esquema de la sintaxis de una etiqueta.

2 - Fundamentos de HTML

Distinguimos dos tipos de etiquetas, de **línea** y de **bloque**:

- **De bloque**. Encierran contenido entre su inicio y su final. Así que hay que abrirlas y cerrarlas.

```
<h1> Este es el contenido del encabezado 1 </h1>
```

- **De línea**. No encierran contenido, así que sólo hay que abrirlas.

```
<br>
```

Etiqueta de salto de línea

3.1 - Estructura en HTML

Los **documentos HTML** siguen una **estructura relativamente estricta**, que debemos seguir al crearlos. Las principales son:

```
<!DOCTYPE html>
```

// Tipo de documento

```
<html>
```

// Etiqueta html

```
  <head>
```

// Cabecera de la página

```
  </head>
```

(Título, etiquetas, metadatos...)

```
  <body>
```

// Cuerpo de la página

```
  </body>
```

(Contenidos de la página)

```
</html>
```

// Cerramos etiqueta html

3.1 - Estructura en HTML - DOCTYPE

La primera sección que debemos incluir es en la que indicamos el **tipo de documento**. Esto lo hacemos en la **primera línea** y sirve para indicar al navegador qué **versión de HTML** utiliza la página web.

Conocer la versión ayudará al navegador a **mostrar la página web correctamente**, por las diferencias entre versiones.

En el caso de **HTML5**, que será el que más utilicemos, se escribe:

```
<!DOCTYPE html>
```

3.1 - Estructura en HTML - DOCTYPE

En las versiones **anteriores a HTML5** la declaración es **más compleja**. Esto es porque estas versiones están **basadas en SGML**, así que requieren que especifiquemos el **DTD** (*definition type document*).

Esto permite al navegador contrastar las normas en el DTD con lo que hay escrito en nuestro archivo .html.

Por ejemplo, en HTML 2, hay que escribir:

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

A continuación, vemos algunas versiones de HTML y sus declaraciones.

3.1 - Estructura en HTML - DOCTYPE

Las versiones más antiguas de HTML siguen siendo compatibles con los navegadores, pero no soportan muchas funcionalidades.

HTML 2.0. No soporta tablas, marcos ni internacionalización.

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML//EN">
```

HTML 3.2. Tiene uso limitado de hojas de estilo, no soporta marcos ni internacionalización

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

3.1 - Estructura en HTML - DOCTYPE

En **HTML 4.01** tiene la mayoría de funciones comunes. Además, diferenciamos tres tipos de DTD:

- **Strict.** Formato estándar anterior a HTML5, sin atributos en desuso, tiene buena compatibilidad con hojas de estilo CSS.

```
<!DOCTYPE      HTML      PUBLIC      "-//W3C//DTD      HTML      4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd
```

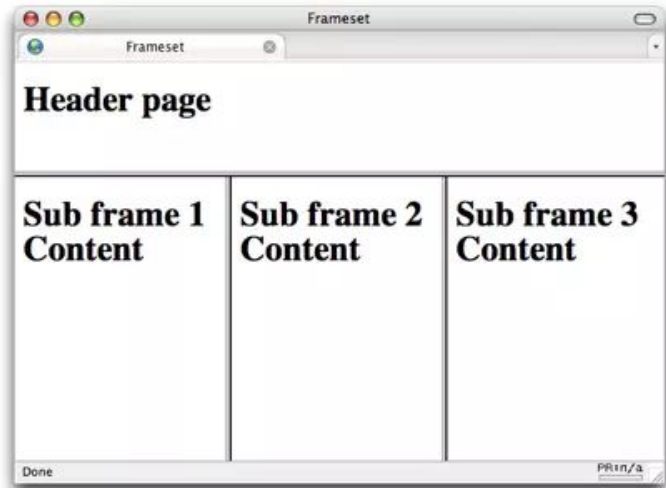
- **Transitional.** Modelo a medio camino del strict, con atributos en desuso.

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

3.1 - Estructura en HTML - DOCTYPE

Frameset. Para declarar marcos: *frames* o *framesets*.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```



Ejemplo de una página web con frames, esta funcionalidad pasó a hacerse de otra forma con HTML5.

3.1 - Estructura en HTML - DOCTYPE

HTML5. Como dijimos, HTML5 se basa en XHTML 1 y DOM, no en SGML. Por esto, la declaración es más sencilla, al no tener que indicar el DTD:

```
<!DOCTYPE html>
```



HTML5 es la versión más moderna de HTML, y la que utilizaremos en este módulo.

3.2 - Estructura en HTML - Etiqueta <html>

En segundo lugar se declara la **etiqueta <html>**, este es el **elemento raíz del documento** y **envuelve todo el código**: el **documento se termina** cuando se encuentra la etiqueta de **cierre </html>**.

```
<!DOCTYPE html>
```

```
<html>
```

```
// Aquí estaría el contenido de la página
```

```
</html>
```

3.3 - Estructura en HTML - Etiqueta <head>

En tercer lugar nos encontramos con la **etiqueta <head>**, esta define la **cabecera del documento** y contiene información sobre este.

La cerramos con </head>. Dentro, podemos incluir **otras etiquetas**:

```
<head>  
  <title>Título del documento</title>  
  <link rel="stylesheet" type="text/css" href="estilo.css">  
</head>
```

- El **título** del documento: **<title>** Título del documento **</title>**
- **Enlaces a hojas de estilo** (p.ej. CSS), utilizando la etiqueta **<link>**.

3.3 - Estructura en HTML - Etiqueta <head>

Las meta-etiquetas o etiquetas **<meta>**, sirven para describir **propiedades del documento**, que suelen utilizar los **navegadores y motores de búsqueda** (un buen uso de estas puede mejorar el SEO de la página).

Algunos ejemplos habituales son:

author	// Autor de la página
generator	// Programa utilizado para crear la página
classification	// Palabras para clasificar la web en buscadores
description	// Descripción del contenido de la página
keywords	// Palabras clave para la página

3.3 - Estructura en HTML - Etiqueta <head>

Con otros **atributos**, como **http-equiv** puede indicar al navegador algunas acciones que debe realizar al cargar la página.

Por ejemplo, para que la página se refresque cada 20 segundos se puede usar la acción **Refresh (actualizar)**:

```
<meta http-equiv="Refresh" content="20">
```

Con esta otra línea, se haría una redirección dentro de 3 segundos.

```
<meta http-equiv="Refresh" content="3;URL=www.misitio.es">
```


3.4 - Estructura en HTML - Etiqueta **<body>**

Contiene el **cuerpo de la página**, es decir, incluye las líneas con las que se genera la **parte visible de la página** (textos, tablas, imágenes, enlaces...). El cuerpo se cierra con la etiqueta **</body>**.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Mi primera página web</title>
  </head>
  <body>
    Este es el cuerpo de la página web
  </body>
</html>
```

4 - Normas y recomendaciones en HTML

A continuación vemos algunas **normas sobre cómo funciona la sintaxis en HTML y recomendaciones** para hacerlos más entendibles:

- Como hemos visto, los documento HTML siguen una **estructura**.

- Los **nombres de elementos y etiquetas no son case-sensitive**.

Es decir, no importan las mayúsculas, es igual `<html>` que `<HTML>`

- Los **nombres de atributos tampoco son case-sensitive**.

P.ej. `` es igual que ``.

- Aún así, por **estilo, se recomienda escribirlos en minúsculas**.

4 - Normas y recomendaciones en HTML

- Los **valores de atributos pueden ser case-sensitive** (según el caso).
P. ej. el nombre de un archivo `` puede ser distinto de ``
- Los **caracteres en blanco (espacios, intros, tabulaciones) fuera de las etiquetas** se interpretan como un espacio, **no afectan al contenido**.

```
<body>Cuerpo de la página</body><h1>Encabezado 1</h1>x
```

```
<body>  
    Cuerpo de la página</body>  
<h1>  
    Encabezado 1  
    </h1>
```

P. ej. estos dos textos se muestran igual en la web, aunque los espacios e intros sean distintos.

4 - Normas y recomendaciones en HTML

- Los **nombres de etiquetas** no pueden tener espacios en blanco.

P. ej. `` no es lo mismo que `<I M G>`

- Los **valores de los atributos** pueden contener **espacios y caracteres no alfanuméricos** si los escribimos **entre comillas**.

P.ej. ``

- Por esto, se recomienda poner los **valores de atributos entre comillas**.

`` y `` significan lo mismo y son válidos los dos, pero se recomienda la versión con comillas.

4 - Normas y recomendaciones en HTML

- Todas las **etiquetas que encierran contenido deben cerrarse**.

P. ej. `<html> </html>` o `<h1> Contenido de la cabecera </h1>`

- Las **etiquetas pueden anidarse** (incluir unas dentro de otras).

`<i> Este texto está en negrita y cursiva </i>`

- Los **exploradores ignoran elementos desconocidos**, sean etiquetas o atributos. Es decir, si están mal escritos, no los reconoce.
- La **primera página de un sitio se suele llamar index.html**, ya que es el archivo que los servidores web detectan como principal por defecto.

Etiquetas HTML

5 - Etiquetas HTML

En este tema vamos a ver algunas de las etiquetas más comunes a la hora de crear páginas web con HTML.

Muchas de ellas no suelen utilizarse en entornos profesionales: es más habitual gestionar el formateado de la página con **hojas de estilo CSS**.



Aún así, vamos a ver las más básicas para familiarizarnos con ellas y trabajar de una forma más sencilla antes de ver .css.

5.1 - Etiquetas para encabezados

Estas etiquetas de **bloque** establecen distintos **niveles de encabezados**:

```
<h1> Título de nivel 1 </h1>
```

```
<h2> Título de nivel 2 </h2>
```

```
<h3> Título de nivel 3 </h3>
```

```
<h4> Título de nivel 4 </h4>
```

```
<h5> Título de nivel 5 </h5>
```

```
<h6> Título de nivel 6 </h6>
```

H1 es el encabezado de mayor nivel (texto más grande, en negrita...).

Los siguientes son cada vez textos de menor nivel, de forma similar a los títulos en editores de texto.

Cuando veamos diferentes etiquetas, prueba a escribir estas líneas en un documento .html y ábrelo con el navegador.

5.2 - Etiquetas para formato de texto

Etiquetas de bloque que cambian la presentación del texto contenido:

`` Texto en negrita ``

`<i>` Texto en cursiva `</i>`

`<tt>` Tipo de letra máquina de escribir `</tt>`

`^{` Texto en superíndice `}`

`_{` Texto en subíndice `}`

`<big>` Aumentar tamaño de fuente `</big>`

`<small>` Reducir tamaño de fuente `</small>`

5.2 - Otras etiquetas para texto

Etiquetas de bloque que cambian la presentación del texto contenido:

```
<hr>
```

Inserta una barra horizontal.

```
<hr size="tamaño">
```

Con size se indica el tamaño de la línea.

```
<br>
```

Inserta un salto de línea.

```
<!-- Comentario -->
```

Esta línea es ignorada por el navegador

```
align="left"
```

```
align="center"
```

```
align="right"
```

Alinean el texto.

Estos tres últimos se aplican como **atributos**.

5.2 - Otras etiquetas para texto

Muchas etiquetas permiten incluir **atributos**, los atributos aportan información sobre esa etiqueta o modifican su comportamiento.

A los atributos hay que **darles un valor**, con el operador = (P. ej. *size = 10*)

P.ej. añadir la etiqueta ***align*** a los textos cambia su alineación.

```
<p align="left">Este párrafo se alinea a la izquierda</p>
```

```
<p align="right">Este párrafo se alinea a la derecha</p>
```

```
<p align="center">Este párrafo se alinea al centro</p>
```

5.3 - Etiquetas para párrafos

Etiquetas que definen párrafos, con espacios o a nivel semántico:

`<p> Párrafo </p>`

Delimita un párrafo, añade un salto de línea.

`<pre> Texto preformateado </pre>`

Permite insertar un texto preformateado, es decir, que se muestra tal y como aparece en el archivo, no se omiten espacios, saltos de línea...

P. ej., el texto de la derecha aparecerá tal cual.

```
<pre>
body{
    color:  red;
}
a    {
    color:green;
}
</pre>
```

5.3 - Etiquetas para párrafos

<div> sirve para agrupar etiquetas en un bloque o sección de la página para luego aplicar estilos. También podemos aplicar atributos como *align*.

```
<div>  
  <h2> Este header está dentro del div </h2>  
  <p> Este párrafo también </p>  
</div>  
<p> Este párrafo está fuera </p>
```

```
<div align="left"> ... </div>
```

Así, podemos crear divisiones con un alineado, color, fuente... aunque tengan varias etiquetas. Es una etiqueta **semántica**.

5.3 - Etiquetas para párrafos

<blockquote> crea una cita en bloque, con un formato concreto. Se suele utilizar para citar autores, otros documentos...

```
<blockquote> Este texto está justificado </blockquote>
```

<address> sirve para escribir direcciones de correo postal.

```
<address> C/ Hermanos Fernández Galiano, 6, 19004 </address>
```

Además de **visuales** (añadir contenido o modificar su aspecto), las etiquetas también son **semánticas** (comunicar qué es cada parte de la web).

P. ej. **<address>** permite a los buscadores encontrar una dirección fácilmente.

5.3 - Etiquetas para párrafos

**** permite agrupar elementos que vayan seguidos en una línea para darles formato con una hoja de estilo.

**** Este contenido se ve afectado por el span ****

En este ejemplo, definimos un span dentro de un párrafo:

<p> Esto es un párrafo, **** y esto un span dentro de un párrafo. **** **</p>**

Lo podemos ver como un **<div>** reducido, aplicado a elementos en línea.

5.4 - Etiquetas para fuentes

Podemos utilizar los atributos **color**, **size** y **face** con el elemento **font**:

```
<font color="código_color"> Texto de ese color </font>
```

El color se escribe en hexadecimal (p.ej. ``) o con su nombre si es de los genéricos (p.ej. ``)

```
<font size="tamaño"> Texto de ese tamaño </font>
```

Este atributo tiene valores entre 1 y 7, por defecto está en el 3.

```
<font face="nombre_fuente"> Texto en esa fuente </font>
```

Se escribe el nombre de una fuente instalada. P.ej. ``

5.4 - Etiquetas para fuentes

También podemos modificar el color y similares en toda la página, cambiando estas propiedades en el body:

```
<body text="Código color">
```

Color del texto por defecto.

```
<body bgcolor="Código color">
```

Color del fondo.

```
<body link="Código color">
```

Cambia el color de los enlaces.

```
<body vlink="Código color">
```

Color de los enlaces visitados

5.5 - Etiquetas para estilos lógicos

De forma similar a `<blockquote>` y `<address>`, existen otras etiquetas que modifican el estilo de un texto e indican información sobre estos.

Estas etiquetas semánticas son descriptivas de la información contenida:

` Texto importante `

Resalta texto (en **negrita**)

` Texto enfatizado `

Resalta texto (en *cursiva*)

No deben usarse de forma intercambiable con `` `<i>`, sino que servirán para hacer saber al navegador los elementos importantes de la web.

5.5 - Etiquetas para estilos lógicos

Otras etiquetas de este tipo se usan para tipos de información concreta:

<code> Líneas de código </code>

Fuentes tipo máquina de escribir.

<samp> Ejemplos </samp>

Ejemplos, estilo similar a <code>.

<var> Variables matemáticas </var>

Para ecuaciones y similares, en cursiva.

<cite> Citas de autor </cite>

Similar a blockquote, en cursiva.

<dfn> Definiciones </dfn>

Definir conceptos, en cursiva.

<kbd> Entradas de texto </kbd>

Estilo similar a <code>

5.6 - Etiquetas multimedia - Imágenes



Para insertar imágenes utilizamos la etiqueta **``**. Como vemos, hace falta el atributo `src`, con la ruta del archivo.

La ruta puede ser a otro sitio web o al sistema de ficheros de nuestro servidor, en este caso, la ruta es relativa a dónde está el archivo `.html`.

P. ej. si `web.html` está en la carpeta `htdocs` y escribo **``**, buscará el logo en la carpeta `htdocs`. No la encontrará en otras rutas..

```

```

Imagen en almacenamiento propio.

```

```

Imagen enlazada de sitio web.

5.6 - Etiquetas multimedia - Imágenes



La etiqueta img admite varios atributos:

```
<img src=foto.jpg width="ancho">
```

Ancho de la imagen

```
<img src=foto.jpg height="altura">
```

Altura de la imagen

```
<img src=foto.jpg align="alineación">
```

Alineación de la imagen

```
<img src=foto.jpg alt="texto_alt">
```

Texto alternativo/descriptivo

```
<img src=foto.jpg border="tamaño">
```

Tamaño del borde

5.6 - Etiquetas multimedia - Vídeo



Desde HTML5 se permite reproducir audio y vídeo sin necesitar plugins externos, como flash y similares.

Para vídeo, se utiliza la etiqueta **<video>**. Ejemplo:

```
<video width="320" height="240" controls autoplay>  
<source src="mi_video.mp4" type="video/mp4">  
</video>
```

Por defecto, se admiten los **formatos .mp4, .webm y .ogg**. Se debe indicar su **tipo MIME** (identificador de tipos de archivo en internet). Se escriben como: video/mp4, video/webm o video/ogg).

5.6 - Etiquetas multimedia - Vídeo



Antes de cerrar la etiqueta **<video>**, se deben añadir uno o más **<source>**:

- **Vídeo** tiene la **configuración del vídeo** con atributos como width, height, controls (si muestra los controles del vídeo), autoplay, loop, muted...
- **Source** especifica el **archivo** que del vídeo, se añaden para soportar varios formatos . El navegador elige el primero con el que sea compatible.

```
<video width="320" height="240" controls autoplay>  
<source src="mi_video.mp4" type="video/mp4">  
<source src="mi_video.webm" type="video/webm">  
El navegador no soporta el vídeo.  
</video>
```

5.6 - Etiquetas multimedia - Audio



Para reproducir archivos de audio, se utiliza la etiqueta **<audio>**.

```
<audio controls autoplay>  
<source src="mi_audio.mp3" type="audio/mpeg">  
</audio>
```

Tiene los mismos atributos que **<video>**, y debe contener **<source>**

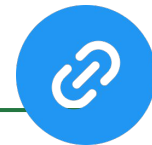
Los formatos aceptados y sus tipos MIME asociados son:

.mp3 -> **audio/mpeg**

.wav -> **audio/wav**

.ogg -> **audio/ogg**

5.7 - Etiquetas para enlaces



Sirven para enlazar a otros sitios web u a otras páginas web del mismo.

Como con las imágenes, distinguiremos **rutas locales y externas**.

```
<a href="http://www.google.com"> Enlace a Google </a>
```

La etiqueta es **<a>**, debe cerrarse con **** y el atributo de la ruta es **href**.

Es una etiqueta de contenido, **el contenido será la parte visible del enlace** y en la que podremos pulsar para desplazarnos a la ruta de href.

En este ejemplo, nos aparecería este texto, que llevaría a google.com.

[Enlace a Google](http://www.google.com)

5.7 - Etiquetas para enlaces



Algunos ejemplos de su uso serían:

- Enlazar a una página web externa:

```
<a href="http://www.google.com"> Enlace a Google </a>
```

- Enlazar a otra página web en nuestro directorio:

```
<a href="audio.html"> Página de prueba de audio </a>
```

```
<a href="/datos/contacto.html"> Contacto </a>
```

En el primer ejemplo, audio.html debe estar en la misma carpeta.

En el segundo, debe estar en una carpeta "datos" en la misma carpeta.

5.7 - Etiquetas para enlaces



- Enlazar a un archivo en nuestros directorios:

```
<a href="imagen.jpg"> Archivo de imagen </a>
```

```
<a href="entrega.zip"> Archivo comprimido </a>
```

```
<a href="music.mp3"> Archivo mp3 </a>
```

Podemos enlazar a archivos en nuestro almacenamiento, según el tipo de archivo, lo mostrará en el navegador o lo descargará directamente.

Recuerda que si el archivo está en una carpeta, debe incluirse en la ruta.

5.7 - Etiquetas para enlaces



También podemos **enlazar a partes concretas de la página web**.

Por ejemplo, volver al inicio de la página desde el final.

- Primero, se coloca un **<a>** (ancla) con el atributo **name**

```
<a name="parrafo">texto-anc1a</a>
```

- Después, podemos poner un link a esa parte con su nombre en un href.

```
<a href="#parrafo">texto-enlace</a>
```

Nota: En los enlaces, el símbolo #, representa el **propio documento**.

P.ej. si ponemos un href="#", refrescará la página.

En #parrafo, se busca un name="parrafo" en el documento actual.

5.7 - Etiquetas para enlaces



Ejemplo de página web con un enlace a otra parte de la página:

```
<h1> Título de la página </h1> <hr>
<a href="#PiePagina"> Ir a pie de página </a>
<p> Contenido de la página </p>
<hr size="1500">
<a name="PiePagina"> <h2> Pie de la página </h2> </a>
<p> Este es el pie de la página </p>
<p> Más contenido en el pie de página </p>
<hr size="1500" noshade>
```

5.7 - Etiquetas para enlaces



Como dijimos, el enlace se podrá abrir a partir del contenido que haya entre `<a>` y ``. Esto significa que también **podemos incluir otros elementos como imágenes o botones, y que actúen como enlace.**

Prueba a crear una página con una imagen enlace como en el ejemplo:

```
<a href="http://www.youtube.com">  
  
</a>
```

También podemos aplicar un enlace a un párrafo o a una sección entera.

5.8 - Etiquetas para listas

Otras etiquetas muy útiles en HTML son las de **listas**.

Son parecidas a las enumeraciones que generamos en programas como Word, permiten listar elementos, procesos, opciones, menús...

Veremos tres tipos de listas, el más básico es el de **lista no ordenada**:

```
<ul>  
  <li> 1º elemento </li>  
  <li> 2º elemento </li>  
  <li> 3º elemento </li>  
</ul>
```



- 1º elemento
- 2º elemento
- 3º elemento

5.8 - Etiquetas para listas - Listas no ordenadas

- Se abren y cierran con la etiqueta `` `` (unordered list).
- Entre medias, incluimos cada elemento con `` ``

```
<ul>  
  <li> Café </li>  
  <li> Tostadas </li>  
  <li> Zumo </li>  
</ul>
```

Abrimos lista

Elementos de la lista

Cerramos lista

- Café
- Tostadas
- Zumo

5.8 - Etiquetas para listas - Listas no ordenadas

Podemos modificar el aspecto de la lista con el atributo **type**. En listas no ordenadas, puede ser "Circle", "Square" o "Disc"

```
<ul type="Circle">
  <li> Lista con "Circle" </li>
</ul>
<ul type="Square">
  <li> Lista con "Square" </li>
</ul>
<ul type="Disc">
  <li> Lista con "Disc" </li>
</ul>
```

- Lista con "Circle"
- Lista con "Square"
- Lista con "Disc"

5.8 - Etiquetas para listas - Listas ordenadas

La diferencia es que sus elementos van ordenados, siguen un orden. Son parecidas a las enumeraciones en programas como Word (1,2,3... a,b,c...)

- Se crean con la etiqueta `` `` (ordered list).
- También utilizamos `` `` para los elementos.

```
<ol>  
  <li> Lunes </li>  
  <li> Martes </li>  
  <li> Miércoles </li>  
</ol>
```

```
1. Lunes  
2. Martes  
3. Miércoles
```

5.8 - Etiquetas para listas - Listas ordenadas

Al ser listas ordenadas, podemos cambiar la forma en que se muestra este orden con el atributo **type**. P. ej. "1,2,3", "I,II,III" o "a,b,c".

Type puede ser "1" (números), "A" (letras mayúsculas), "a" (letras minúsculas), "I" (nº romano mayúscula) o "i" (nº romano minúscula).

```
<ol type="i">  
  <li> Enero </li>  
  <li> Febrero </li>  
  <li> Marzo </li>  
</ol>
```

i. Enero
ii. Febrero
iii. Marzo

5.8 - Etiquetas para listas - Listas de definición

Las **listas de definiciones** se abren y cierran con `<dl>` `</dl>`.

La diferencia está en que en lugar de contener elementos sueltos, estos van por parejas de término y definición. Estos utilizan `<dt>` y `<dd>`.

```
<dl>
  <dt> Hardware </dt>
  <dd> Parte física del equipo </dd>
  <dt> Software </dt>
  <dd> Parte lógica del equipo </dd>
</dl>
```

```
Hardware
  Parte física del equipo
Software
  Parte lógica del equipo
```

5.8 - Etiquetas para listas - Listas ordenadas

También podemos anidar listas dentro de otras, en elementos:

```
<h2>ÍNDICE.</h2>
<ol type="I">
  <li>Internet</li>
  <li>Evolución</li>
  <ol type="A">
    <li>Internet 1.0</li>
    <li>Internet 2.0</li>
    <li>Internet 3.0</li>
  </ol>
  <li>Tendencias</li>
</ol>
```

ÍNDICE.

- I. Internet
- II. Evolución
 - A. Internet 1.0
 - B. Internet 2.0
 - C. Internet 3.0
- III. Tendencias

5.8 - Etiquetas para listas - Listas ordenadas

LISTAS ANIDADAS NUMERADAS:

- I. Elemento 1
 - A. Elemento 1.1
 - B. Elemento 1.2
 - 1. Elemento 1.2.1
 - 2. Elemento 1.2.2
 - 3. Elemento 1.2.3
- II. Elemento 2
 - a. Elemento 2.1
 - b. Elemento 2.2
- III. Elemento 3
- IV. Elemento 4
 - 1. Elemento 4.1
 - 2. Elemento 4.2
 - i. Elemento 4.2.1
 - ii. Elemento 4.2.2
 - 3. Elemento 4.3

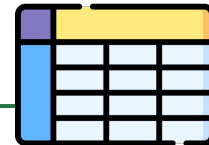
Intenta replicar esta página utilizando listas anidadas. Si te fijas, todas son listas ordenadas con distinto type.



Etiquetas HTML

Tablas

5.9 - Tablas en HTML

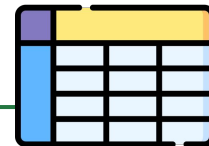


Las **tablas** se introdujeron en HTML 3.2, trayendo muchas ventajas al permitir organizar los elementos en **filas y columnas**.

Además de insertar tablas dentro de una página, pueden crearse páginas con todos los elementos contenidos en una tabla, jugando con alineación, anchos, altos...

Time Table					
Hours	Mon	Tue	Wed	Thu	Fri
	Math	Science	Math	Science	Arts
	Math	Science	Math	Science	Arts
	Lunch				
	Math	Science	Math	Project	
	Math	Science	Math		

5.9 - Tablas en HTML



Para crear una **tabla** utilizamos la etiqueta `<table>` `</table>`. Dentro hay:

- **Filas** (rows) con `<tr>` `</tr>` que contienen celdas.
- **Celdas** (data) con `<td>` `</td>` que forman columnas.
- **Títulos** de columnas (header) con `<th>` `</th>`

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

5.9 - Tablas en HTML

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

5.9 - Tablas en HTML

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

<table> (tabla)

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

</table>

5.9 - Tablas en HTML

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

<table> (tabla)

<tr>

columna primera	columna segunda
-----------------	-----------------

</tr>

<tr>

fila 1 – elemento 1	fila 1 – elemento 2
---------------------	---------------------

</tr>

<tr>

fila 2 – elemento 1	fila 2 – elemento 2
---------------------	---------------------

</tr>

</table>

5.9 - Tablas en HTML

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

<table> (tabla)

<tr>

<th> Título col </th>

<th> Título col </th>

</tr>

<tr>

fila 1 – elemento 1

fila 1 – elemento 2

</tr>

<tr>

fila 2 – elemento 1

fila 2 – elemento 2

</tr>

</table>

5.9 - Tablas en HTML

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

<table> (tabla)

<tr>

<th> Título col </th>

<th> Título col </th>

</tr>

<tr>

<td> Celda </td>

<td> Celda </td>

</tr>

<tr>

<td> Celda </td>

<td> Celda </td>

</tr>

</table>

5.9 - Tablas en HTML

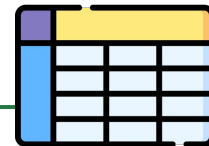


```
<table border="1">  
  <tr><th> Columna 1 </th> <th> Columna 2 </th> </tr>  
  <tr><td> Celda 1-1 </td> <td> Celda 1-2 </td> </tr>  
  <tr><td> Celda 2-1 </td> <td> Celda 2-2 </td> </tr>  
</table>
```

columna primera	columna segunda
fila 1 – elemento 1	fila 1 – elemento 2
fila 2 – elemento 1	fila 2 – elemento 2

Las etiquetas </th> y </td> pueden omitirse, pero son recomendables.

5.9 - Tablas en HTML



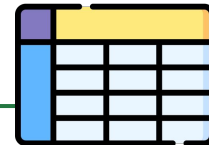
Vamos a ver que hay muchos más atributos para configurarlas:

border	Ancho del borde (px)
bordercolor	Color del borde (color)
align	Alineación del texto (left,center,right)
valign	Alineación vertical (top,bottom,middle)
background	Imagen de fondo (enlace)

cellpadding	Espacio entre los bordes de celdas y su contenido
cellspacing	Espacio entre bordes
height	Altura de la tabla
width	Anchura de la tabla
bgcolor	Color de fondo

Varios de los atributos pueden configurarse en la tabla entera <table>, una fila <tr> o una celda <td> o <th>, según dónde lo coloquemos.

5.9 - Tablas en HTML - Bordes

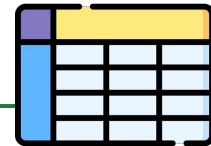


Para que la tabla tenga bordes, hace falta definir el atributo **border** en la etiqueta **<table>**. P.ej. `<table border="5">`.

Podemos cambiar su color (**bordercolor**) y espaciado (**cellspacing**).

```
<table border="3" bordercolor="Blue" cellspacing="5">
  <tr><th> Columna 1 </th><th> Columna 2 </th></tr>
  <tr><td> Celda 1-1 </td><td> Celda 1-2 </td> </tr>
  <tr><td> Celda 2-1 </td><td> Celda 2-2 </td> </tr>
</table>
```

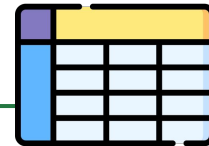
5.9 - Tablas en HTML - Tamaño



Se puede ajustar el tamaño de la tabla con **width** y **height**. Podemos darle un valor en píxeles (100) o en porcentaje de ventana (50%)

```
<table border="1" width="400" height="100">
  <tr><td> Celda 1-1 </td><td> Celda 1-2 </td> </tr>
  <tr><td> Celda 2-1 </td><td> Celda 2-2 </td> </tr>
</table>
<hr>
<table border="3" width="70%" height="300">
  <tr><td> Celda 1-1 </td><td> Celda 1-2 </td> </tr>
  <tr><td> Celda 2-1 </td><td> Celda 2-2 </td> </tr>
</table>
```

5.9 - Tablas en HTML - Alineación

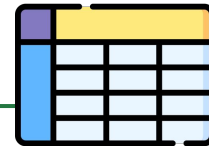


Se puede cambiar la alineación vertical y horizontal de la tabla y celdas con **align** (left, center, right) y **valign** (top, middle, bottom).

Si los ponemos en la tabla, cambia donde se coloca la tabla entera en la página. Colocado en filas o celdas, cambia la alineación de su contenido.

```
<table align="center" border="1" width="500" height="200">
  <tr align="center" valign="top">
    <td> Celda 1-1 </td><td align="right"> Celda 1-2 </td> <td> Celda 1-3 </td>
  </tr>
  <tr align="right" valign="bottom">
    <td> Celda 2-1 </td><td align="center"> Celda 2-2 </td> <td> Celda 2-3 </td>
  </tr>
</table>
```

5.9 - Tablas en HTML - Fondo

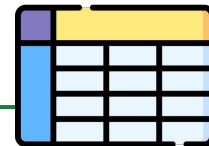


Podemos cambiar el color de fondo con **bgcolor** o poner una imagen con **background**. El color podemos cambiarlo para la tabla, una fila o celda.

```
<table border="4" bgcolor="Lime">
  <tr bgcolor="Cyan"><th> Columna 1 </th><th> Columna 2 </th></tr>
  <tr><td> Celda 1-1 </td><td> Celda 1-2 </td> </tr>
  <tr><td> Celda 2-1 </td><td bgcolor="Yellow"> Celda 2-2 </td> </tr>
</table>
```

```
<table border="4" background="fondo.jpg">
  <tr background="fondo2.png"><td> Celda 1-1 </td><td> Celda 1-2 </td> </tr>
  <tr><td> Celda 2-1 </td><td background="fondo3.png"> Celda 2-2 </td> </tr>
</table>
```

5.9 - Tablas en HTML - Espaciado de celdas



Con **cellpadding** podemos modificar el espacio vacío entre los bordes de cada celda y su contenido.

En HTML, este atributo solo puede modificarse para la tabla entera.

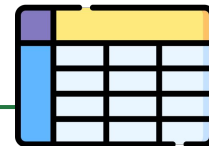
Celda 1-1	Celda 1-2
Celda 2-1	Celda 2-2

`<table>`

Celda 1-1	Celda 1-2
Celda 2-1	Celda 2-2

`<table cellpadding="40">`

5.9 - Tablas en HTML - Combinar celdas



Con **colspan** y **rowspan** podemos combinar celdas. Se le asigna el nº de celdas que queremos que ocupe la celda, pero estas celdas deben existir en la tabla. P.ej. no podemos crear una celda 1x4 en una tabla 2x2.

```
<table border=1>
  <tr><td>Dato1</td> <td> Dato2 </td></tr>
  <tr><td colspan="2"> Dato3 </td></tr>
</table>
```

Dato1	Dato2
Dato3	

```
<table border=1>
  <tr><td> Dato1</td><td rowspan="2"> Dato3 </td></tr>
  <tr><td> Dato2 </td></tr>
</table>
```

Dato1	Dato3
Dato2	

5.9 - Tablas en HTML - Combinar celdas



Prueba a crear estas tablas utilizando colspan y rowspan:

Dato1	
Dato2	Dato3

dato1	Dato2
	Dato3

dato1	Dato2
	Dato3
Dato4	Dato5



Etiquetas HTML

Formularios

5.10 - Formularios en HTML

Los **formularios** son formas de **introducir datos** en las páginas web, son muy habituales para que los usuarios rellenen sus datos (usuario, contraseña), marquen opciones en casillas, desplegables...



Registration Form

First Name :

Last Name :

Email :

Password :

Gender:

Male ☐

Female ☐

Other ☐

Age :

Se crean dentro de las etiquetas **<form>** **</form>**, pero como vamos a ver, los hay de muchos tipos, con distintos atributos.

Sirven para **recopilar información** y adjuntarla por correo, a otras páginas o scripts.

5.10 - Formularios - `<form>` y `<input>`

La estructura general será:

- Un formulario `<form>` `</form>` que puede contener distintas entradas.
- Las entradas o `<input>`, de un tipo concreto y con un identificador.

```
<form>  
  <input type="text" name="user"> Usuario <br>  
  <input type="password" name="password"> Contraseña <br>  
  <input type="submit" name="botonLogin" value="Login">  
</form>
```

Tanto `<form>` como `<input>` tienen muchos atributos para configurarlos, como vamos a ir viendo.

5.10 - Formularios - <form> y <input>

```
<form action="home.html">
  First Name:<br>
  <input type="text" name="first_name">
</br>
  Last Name:<br>
  <input type="text" name="last_name">
</br>
  Email:<br>
  <input type="text" name="email">
</br>
  <input type="submit" name="Submit">

</form>
```

First Name:

Last Name:

Email:

Submit Query

Ejemplo de un formulario

5.10 - Formularios - Etiqueta <form>

En la etiqueta **form** se configuran varias etiquetas, aunque en este módulo apenas trabajaremos con ellas:

- **action** Es la acción que se va a realizar con la información, a dónde se envía, normalmente la ruta de otra página. P.ej. **action="login.php"**
- **method** Forma en que se envía la información, puede ser "post" o "get". Get es menos seguro porque se ven los valores del formulario en la URL, en nuestro nivel, son indiferentes. P.ej. **method="POST"**
- **enctype** Cómo se muestra la información (encoding). Lo ignoraremos en la mayoría de casos. P.ej. **enctype="text/plain"**.

5.10 - Formularios - Etiqueta <form>

Añadiendo estos atributos al ejemplo anterior, la información generada se enviaría a un archivo login.php, que tendría que recibirla y procesarla.

```
<form action="login.php" method="POST">  
  <input type="text" name="user"> Usuario <br>  
  <input type="password" name="password"> Contraseña <br>  
  <input type="submit" name="botonLogin" value="Login">  
</form>
```

No vamos a trabajar esto en detalle, ya que hacen falta scripts de código (PHP, JavaScript...) para procesar la información de los formularios

5.10 - Formularios - Etiqueta <input>

Con **<input>** se crea un elemento del formulario. Es una etiqueta en línea, sin contenido. Se añaden textos, párrafos y demás para identificarlos.

```
<input type="text" name="user"> Nombre de usuario
```

Sus atributos principales son **type** y **name**, independientemente del tipo:

- **type** indica de qué tipo es la entrada (texto, número, botón, imagen...) Cada tipo tendrá otros atributos propios para configurarlo.

- **name** indica el nombre de la entrada, en HTML no vemos su efecto, pero al procesar los formularios permite identificar cada parte del formulario y su respuesta con un nombre.

5.10 - Formularios - <input type>

Hay muchos **tipos** de **entradas** <input type="...">, que iremos viendo:

text	Caja de texto
password	Texto oculto con ***
radio	Botones donde se elige sólo una opción
checkbox	Botones donde se pueden elegir varias opciones
image	Utiliza una imagen como botón de submit

submit	Botón para enviar los datos
reset	Botón para limpiar los datos
button	Botón genérico (se usa con código)
file	Para subir archivos
hidden	Datos ocultos (se usan con código)

Algunos los utilizaremos más a menudo que otros, p.ej. button y hidden, se utilizan al combinar formularios con scripts de código.

5.10 - Formularios - <input type>

Con **HTML5** se añadieron varios tipos de entrada nuevos y atributos para configurar los formularios de forma más sencilla.

time	Seleccionar una hora.
date	Fechas (con calendario)
datetime-local	Fecha y hora (sin zona horaria)
month	Mes y año.
week	Semana y año.
range	Una barra con un rango de nºs.

color	Seleccionar un color.
email	Correo electrónico
search	Similar a text, para búsquedas.
tel	Nº de teléfono
number	Sólo acepta números
url	Para direcciones web

P. ej.: **<input type="email" name="correo_login">**

5.10 - Formularios - Atributos de <input>

También hay atributos que podemos añadir en **<input>** para modificar su comportamiento. P.ej. **<input type="text" *required*>**

required	Es obligatorio responder
placeholder	Muestra un valor de ejemplo
autofocus	Aparece seleccionado
autocomplete	Si se autocompleta o no
value	Valor inicial del input

disabled	Input desactivado
readonly	No puede modificarse

Hay atributos muy específicos, p.ej. **multiple** permite adjuntar varios archivos en un **file**.

P. ej.: **<input type="email" name="correo_login">**

5.10 - Formularios - `<input type="text">`

Con ***type="text"***, añadimos entradas de texto genéricas.

Se pueden modificar con muchos **atributos específicos**:

size (tamaño de la caja)

maxlength (longitud máxima aceptada)

value (valor por defecto)

title (texto que aparece al poner el ratón)

```
<form>
```

```
  Escribe tu nombre:
```

```
    <input type="text" name="nombre" placeholder="Tu nombre">
```

```
</form>
```

5.10 - Formularios - `<input type="password">`

Con ***type="password"***, se crea una entrada de texto con el símbolo ****, para ocultar **contraseñas**. Comparte muchos atributos con *text*.

```
<form>  
  Escribe tu contraseña:  
  <input type="password" name="contrasena">  
</form>
```

¡OJO!

Aunque utilicemos *password*, la contraseña seguirá viéndose en la URL si utilizamos *method="GET"* en la etiqueta *<form>*.

5.10 - Formularios - `<input type="radio">`

Con **`type="radio"`**, se crean casillas pulsables, en las que sólo se puede pulsar una de las opciones:

Para cada opción, añadimos una nueva etiqueta radio.

```
<form>
```

```
  Selecciona tu curso:
```

```
  <input type="radio" name="curso" value="SMR" /> SMR
```

```
  <input type="radio" name="curso" value="ASIR" /> ASIR
```

```
  <input type="radio" name="curso" value="DAM" /> DAM
```

```
</form>
```

Para que los radio estén asociados al mismo nombre y sólo se pueda pulsar uno, hay que ponerles el mismo atributo **`name`**.

5.10 - Formularios - `<input type="checkbox">`

Con **`type="checkbox"`**, se crean casillas para pulsar, en las que podemos marcar una, varias o ninguna opción.

Igual que con radio, añadimos una nueva etiqueta para cada opción.

```
<form>
  Selecciona tu preferencia:
  <input type="checkbox" name="m" value="m" /> Mañana
  <input type="checkbox" name="t" value="t" /> Tarde
</form>
```

En checkbox, se les da diferentes nombres, al poder pulsar varios a la vez.

5.10 - Formularios - Botones - `<input="submit">`

Hay varios tipos de botones, que actúan sobre el formulario en el que estén (dentro de su etiqueta **`<form>`**).

- `type="submit"`, es necesario para enviar los resultados del formulario según hayamos configurado el `<form>` con su atributo `action`.

Por esto, es habitual encontrar este atributo en todos los formularios.

```
<form>
```

```
  Indica tu nombre:
```

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

```
  <input type="submit">
```

```
</form>
```



Enviar

5.10 - Formularios - Botones - `<input="reset">`

- `type="reset"`, reinicia todos las entradas a su valor por defecto (vacío).
- `type="button"`, permite añadir comportamientos adicionales si añadimos scripts de código a nuestras páginas (no lo utilizamos en este módulo).

```
<form>  
  Indica tu nombre:  
  <input type="text" name="nombre">  
  <input type="reset">  
</form>
```

Borrar datos del formulario

5.10 - Formularios - Botones - <input="range">

Con **<range>** creamos una barra que permite elegir entre un rango de números (por defecto 0-100). Podemos modificar el rango con **min y max**.

```
<form action method="get">  
  Volumen (entre 0 y 10):  
  <input type="range" name="vol" min="0" max="10">  
  <input type="submit">  
</form>
```

Volumen (entre 0 y 10): 

Con el atributo **step** podemos hacer que acepte números en un intervalo, p.ej. con **step="5"** aceptará valores como 0, 5, 10, 15...

5.10 - Formularios - <input> Time, date, month...

Hay varios tipos de input referidos a **tiempo y fechas**. Se comportan de forma parecida y utilizan varios **atributos en común**.

```
<form>
  <input type="time">
  <input type="date">
  <input type="datetime-local">
  <input type="week">
  <input type="month">
</form>
```

- **min y max** da valores mínimos y máximos a la entrada.
- **step** permite definir separación entre valores (0,2,4), (5,10,15...)
- **pattern** define un patrón para la entrada (p.ej. xx-xx-xx)

Estos atributos son compatibles con otros types, según el caso.

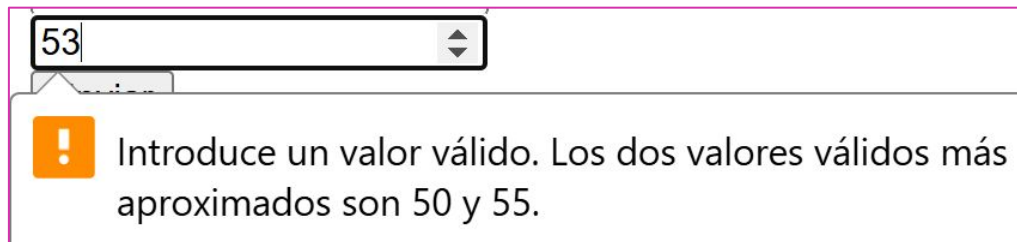
5.10 - Formularios - <input> Time, date, month...

Ejemplos de min, max y step:

```
<form>  
  <input type="date" min="2020-01-01" max="2024-12-31"> <br>  
  <input type="number" min="0" step="5"> <br>  
</form>
```

El primer input sólo permite fechas entre 2020 y 2024.

El segundo sólo acepta valores de 5 en 5 mayores a partir de 0.



The image shows a web browser interface with a number input field. The field contains the number '53'. Below the input field, there is a validation error message. The message consists of an orange square icon with a white exclamation mark, followed by the text: 'Introduce un valor válido. Los dos valores válidos más aproximados son 50 y 55.'

5.10 - Formularios - <input> Time, date, month...

Para definir **patrones**, hay que utilizar expresiones regulares siguiendo la sintaxis de JavaScript, que no vemos en este módulo.

Aquí vemos un ejemplo dando formato a un DNI (8 números y 1 letra):

```
<form action method="get">  
  <p> Introduce tu DNI: </p>  
  <input type="text" placeholder="00000000X"  
    pattern="[0-9]{8}[A-Z]{1}" required>  
  <input type="submit">  
</form>
```

Introduce tu DNI:

5.10 - Formularios - <input> Time, date, month...

Otro ejemplo con un teléfono:

Tiene 3 números, guión, dos números, guión, tres números.

```
<form>
  Introduce un teléfono:
  <input type="tel" name="phone" placeholder="123-45-678"
    pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}" required>
  <p> Formato: 123-45-678 </p>
  <input type="submit" value="Submit">
</form>
```

5.10 - Formularios - Otros elementos (no <input>)

Además del atributo input, hay otras etiquetas para definir elementos de formulario como ***select***, ***fieldset***, ***textarea***... Veremos los más comunes.

<label>	Crea identificadores visibles asociados a los elementos del formulario.
<select>	Entrada en forma de desplegable en la que se da a elegir entre opciones.
<option>	Cada una de las opciones en un <select> o un <datalist>
<textarea>	Caja de texto que puede ser de varias líneas (tamaño ajustable)
<fieldset>	Permite agrupar partes de un formulario, representada con una caja
<legend>	Permite mostrar un nombre identificador para un <fieldset>
<datalist>	Opciones predefinidas para completar una entrada.

5.10 - Formularios - Identificadores <label>



Hasta ahora hemos identificado cada parte del formulario con un atributo **name** de cara al navegador y de cara al usuario con un texto o párrafo.

Lo hemos hecho así para no complicar el texto, pero en un entorno profesional, los elementos del formulario se identifican con **name** y **label**.

<label> es una etiqueta contenedora, que se asocia a un elemento con **for**.

P.ej. **<label for="apellido"> Introduce tu apellido </label>**

Se asocia a un elemento con un atributo **id** con el mismo valor que el **for**.

P.ej. **<input type="text" id="apellido" name="apellido">**

5.10 - Formularios - Identificadores <label>



Utilizado en un formulario sencillo, el resultado sería este:

Introduce tu apellido:

```
<form>
  <label for="apellido"> Introduce tu apellido: </label><br>
  <input type="text" id="apellido" name="apellido">
</form>
```

Estéticamente no hay diferencia, al ser una etiqueta contenedora podemos dar el formato que queramos al texto, ponerlo en un <p>... Como ventajas:

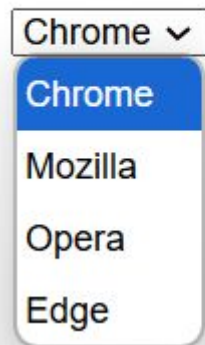
- El label es un **área clickable** que lleva al elemento asociado.
- Permite a los **lectores de pantalla** para personas ciegas leer las labels.

5.10 - Formularios - Desplegables <select>

<select> permite crear una entrada en la que elegimos entre opciones en un desplegable. Cada opción se define con **<option>** y debe tener un value.

El texto que aparece en cada opción se escribe entre <option> y </option>

Elige un navegador:



Chrome ▾

Chrome

Mozilla

Opera

Edge

```
<form>
  <label for="navegador"> Elige navegador: </label><br>
  <select name="navegador" id="navegador" size="1">
    <option value="Chrome">Chrome</option>
    <option value="Mozilla">Mozilla</option>
    <option value="Opera">Opera</option>
  </select>
</form>
```

size indica cuantas opciones aparecen en el desplegable a la vez.

5.10 - Formularios - Texto multi-línea <textarea>

Con **<textarea>** podemos definir una caja de texto con varias líneas.

Tiene varios atributos para definir su tamaño:

- **cols** indica el ancho de la caja en caracteres
- **rows** indica el alto de la caja en líneas de texto

Área de texto

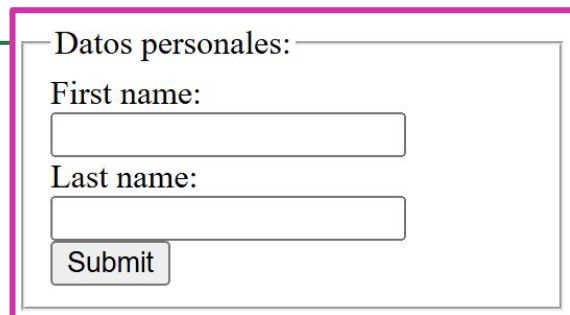
Texto en el área

```
<form>
  <label for="texto"> Área de texto </label><br>
  <textarea rows="5" cols="20" name="texto" id="texto">
    Texto por defecto en el área de texto
  </textarea>
</form>
```

5.10 - Formularios - Grupos <fieldset> y <legend>

<fieldset> es una etiqueta contenedora en la que podemos incluir otros elementos del formulario. Esto permite agruparlos y mostrarlos juntos en una “caja”. Con **<legend>** podemos añadir un identificador a la caja.

```
<form>
  <fieldset>
    <legend> Datos personales: </legend>
    <label for="nombre">First name:</label><br>
    <input type="text" id="nombre" name="nombre"><br>
    <label for="apellidos">Last name:</label><br>
    <input type="text" id="apellidos" name="apellidos"><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```



Datos personales:

First name:

Last name:

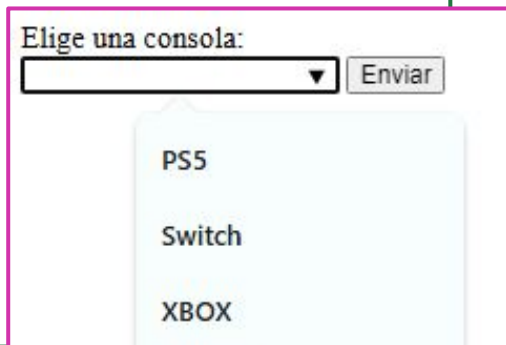
Submit

5.10 - Formularios - Valores predefinidos <datalist>

<datalist> permite definir una serie de valores por defecto para un input. Cada valor se representa con un **<option>** dentro de **<datalist>**.

Para asociar el datalist a un input, ponemos el atributo **id** en **<datalist>** y el atributo **list** en **<input>**, ambos con el mismo valor.

```
<form>
  <p> Elige una consola: <p>
  <input type="text" list="consolas" name="consola">
    <datalist id="consolas">
      <option value="PS5">
      <option value="Switch">
      <option value="XBOX">
    </datalist>
  <input type="submit">
</form>
```



Elige una consola:

▼

Enviar

PS5

Switch

XBOX

6. HTML - Conclusión

- En este tema hemos visto las **etiquetas y atributos más habituales** en HTML, pero ten en cuenta que **existen muchas más**.
- Para comprobar los atributos de una etiqueta, formas de uso o buscar información sobre elementos que no hemos visto, es recomendable consultar la **documentación oficial**. [W3C tiene una página muy completa.](#)
- La mayoría de páginas web se maquetan con **hojas de estilo CSS**, además de con HTML, esto lo veremos más adelante.
- Además, también utilizan **scripts en lenguajes servidor o cliente** para elementos dinámicos o interactivos.