

## ÍNDICE

<b>1. INTRODUCCIÓN HTML Y XHTML</b>	<b>6</b>
1.1 CARACTERÍSTICAS	6
1.2 VERSIONES	7
1.3 DIFERENCIAS HTML-XHTML	8
1.4 DOCTYPE	8
<b>2. DOCUMENTO HTML</b>	<b>9</b>
2.1 ESTRUCTURA DEL DOCUMENTO	9
2.2 SINTÁXIS BÁSICA	9
2.3 ETIQUETAS	10
2.4 CARÁCTERES ESPECIALES	10
ACTIVIDAD 1: CREAR ESTRUCTURA BÁSICA HTML	10
2.5 VALIDADOR HTML	11
2.6 DEFINICIÓN DE LA CODIFICACIÓN DEL TEXTO	12
2.7 ETIQUETAS BÁSICAS DE TEXTO	13
▪ <P>	13
▪   <WBR>	13
▪ <HR>	13
ACTIVIDAD 2: INTRODUCIR TEXTO	13
▪ <EM> <I>	13
▪ <STRONG> <B>	13
ACTIVIDAD 3: FORMATO DE TEXTO BÁSICO	14
2.8 ELEMENTOS EN LÍNEA Y EN BLOQUE	14
2.9 ANIDACIÓN DE ELEMENTOS	14
2.10 ATRIBUTOS	15
▪ FORMATO	15
▪ ATRIBUTO GLOBAL	15
▪ ATRIBUTO BOOLEANO	15
▪ COMILLAS EN EL VALOR DE LOS ATRIBUTOS	16
ACTIVIDAD 4: ELEMENTOS HTML	17
2.11 <HEAD> CABECERA DEL DOCUMENTO	17
▪ <CHARSET>	17
▪ <TITLE>	17
ACTIVIDAD 5: UTILIZAR <TITLE>	17
▪ <META>	17
ACTIVIDAD 6: INCLUIR METADATOS	18

▪ <LINK> .....	19
ACTIVIDAD 7: ASOCIAR UN ICONO .....	19
2.12 IDIOMA DEL DOCUMENTO .....	19
▪ IDIOMA GLOBAL .....	19
▪ IDIOMA PARCIAL .....	19
ACTIVIDAD 8: ASOCIAR UN LENGUAJE .....	20
2.13 ESTRUCTURA DE PÁGINA .....	20
▪ <HEADER> .....	21
▪ <NAV> .....	21
▪ <MAIN> .....	21
▪ <ASIDE> .....	21
▪ <ARTICLE> .....	21
▪ <SECTION> .....	21
▪ <FOOTER> .....	21
ACTIVIDAD 9: ESTRUCTURAR LA PÁGINA .....	22
<b>3. ENCABEZADOS .....</b>	<b>23</b>
▪ <H1> <H2> <H3> <H4> <H5> <H6> .....	23
ACTIVIDAD 10: ENCABEZADOS .....	23
<b>4. ETIQUETAS DE TEXTO .....</b>	<b>24</b>
4.1 ETIQUETAS DE FORMATO .....	24
▪ <I> .....	24
▪ <B> .....	24
▪ <U> .....	24
▪ <SUB> .....	24
▪ <SUP> .....	24
▪ <STRIKE> .....	24
▪ <PRE> .....	24
4.2 ETIQUETAS SEMÁNTICAS (DE SIGNIFICADO) .....	25
▪ <EM> .....	25
▪ <STRONG> .....	25
▪ <CODE> .....	25
▪ <ABBR> .....	26
▪ <MARK> .....	26
▪ <SMALL> .....	26
▪ <ADDRESS> .....	26
▪ <TIME> .....	26
4.3 ETIQUETAS PARA CITAS .....	26
▪ <Q> .....	26

▪ <BLOCKQUOTE>.....	26
▪ <CITE> .....	27
<b>5. ETIQUETAS DE DELIMITACIÓN .....</b>	<b>27</b>
▪ <DIV> .....	27
▪ <SPAN>.....	27
ACTIVIDAD 11: ETIQUETAS DE TEXTO .....	27
<b>6. LISTAS .....</b>	<b>28</b>
6.1 LISTAS ORDENADAS Y NO ORDENADAS .....	28
▪ <UL> <LI>.....	28
▪ <OL> <LI> .....	28
6.2 ATRIBUTOS DE LISTAS .....	28
▪ TYPE .....	28
▪ START .....	29
▪ VALUE .....	30
▪ REVERSED.....	30
6.3 LISTAS DE DESCRIPCIÓN DE TÉRMINOS .....	30
▪ <DL><DT><DD>.....	30
ACTIVIDAD 12: LISTAS ORDENADAS Y NO ORDENADAS .....	30
<b>7. DOM (DOCUMENT OBJECT MODEL) .....</b>	<b>31</b>
7.1 ÁRBOL DE NODOS.....	31
ACTIVIDAD 13: DOM Y VALIDADOR.....	33
<b>8. HIPERENLACES.....</b>	<b>34</b>
8.1 URL ( <i>UNIFORM RESOURCE LOCATOR</i> ).....	34
▪ RESTRICCIONES EN LAS URLs.....	35
▪ URLS EXTERNAS E INTERNAS .....	36
▪ RUTAS RELATIVAS Y ABSOLUTAS .....	36
8.2 ENLACES .....	39
▪ <A> .....	39
▪ ENLACE A UN PUNTO DEL MISMO DOCUMENTO: .....	39
▪ ENLACE A CORREO ELECTRÓNICO .....	39
ACTIVIDAD 14: ENLACES .....	40
<b>9. TABLAS.....</b>	<b>41</b>
▪ <TABLE>.....	41
▪ <TR> .....	41
▪ <TD> .....	41
▪ <TH> .....	41
ACTIVIDAD 15: CREAR UNA TABLA BÁSICA .....	42

▪ <CAPTION> .....	43
▪ ATRIBUTO COLSPAN .....	43
▪ ATRIBUTO ROWSPAN .....	44
▪ <THEAD><TBODY><TFOOT>.....	44
ACTIVIDAD 16: CREAR TABLAS.....	45
<b>10. IMÁGENES .....</b>	<b>46</b>
10.1 TIPOS DE IMÁGENES EN UN DOCUMENTO WEB .....	46
10.2 ETIQUETAS PARA IMÁGENES .....	47
▪ <IMG> .....	47
▪ TAMAÑO DE LA IMAGEN.....	47
▪ <FIGURE> <FIGCAPTION> .....	49
ACTIVIDAD 17: INSERTAR IMÁGENES.....	49
▪ <PICTURE> <SOURCE>.....	49
ACTIVIDAD 18: IMAGEN CON MEDIA QUERY .....	49
<b>11. VÍDEO Y AUDIO.....</b>	<b>50</b>
▪ <VIDEO>.....	50
▪ <SOURCE> .....	51
▪ <AUDIO>.....	51
ACTIVIDAD 19: INSERTAR VIDEO Y AUDIO.....	51
<b>12. MAPA DE IMAGEN.....</b>	<b>52</b>
▪ <IMG> <MAP> <AREA> .....	52
▪ COORDENADAS.....	52
▪ EJEMPLO. CREACIÓN DE MAPA DE IMÁGENES.....	54
ACTIVIDAD 20: MAPA DE IMÁGENES.....	54
<b>13. INSERTAR OBJETOS .....</b>	<b>55</b>
▪ <IFRAME>.....	55
▪ COPIAR CÓDIGO DE INSERCIÓN .....	56
ACTIVIDAD 21: INSERTAR OBJETOS CON IFRAMES .....	56
<b>14. FORMULARIOS .....</b>	<b>57</b>
14.1 INTRODUCCIÓN .....	57
14.2 DATOS DEL FORMULARIO .....	57
14.3 PROTOCOLOS DE TRANSFERENCIA DE LOS DATOS.....	58
14.4 MÉTODO DE TRANSFERENCIA DE LOS DATOS .....	59
▪ GET .....	59
▪ POST .....	59
14.5 ETIQUETAS DE FORMULARIO .....	59
▪ <FORM> .....	59

▪ <INPUT> .....	60
▪ ALGUNOS EJEMPLOS DEL USO DE <INPUT>: .....	62
ACTIVIDAD 22. FORMULARIOS BÁSICOS .....	63
▪ <BUTTON> .....	64
▪ <TEXTAREA> .....	64
▪ <SELECT><OPTION> .....	65
▪ <OPTGROUP> .....	67
ACTIVIDAD 23. FORMULARIOS AVANZADOS .....	67
▪ <FIELDSET><LEYEND> .....	68
▪ <LABEL> .....	68
ACTIVIDAD 24. ETIQUETAS COMPLEMENTARIAS DE FORMULARIO .....	68

### Enlaces de interés

[HTML Tutorial \(w3schools.com\)](https://www.w3schools.com/html/)

[HTML: HyperText Markup Language | MDN \(mozilla.org\)](https://developer.mozilla.org/en-US/docs/Web/HTML)

[HTML 5.1 2nd Edition \(w3.org\)](https://www.w3.org/html/)

## 1. Introducción HTML y XHTML

HTML es el lenguaje utilizado para crear la mayor parte de las páginas web. Es un estándar reconocido en todos los navegadores, por lo tanto, todos ellos visualizan una página HTML de forma muy similar independientemente del sistema operativo sobre el que se ejecutan.

**HTML** (Hypertext Markup Language) Lenguaje de Marcado de Hipertexto

**XHTML** (eXtensible Hyper Text Markup Language) XHTML o Lenguaje de Marcas de Hipertexto Extendido

### 1.1 Características

- ➔ **Texto plano:** compuestos por caracteres de texto que pueden interpretarse por un editor de texto.

Un documento escrito con lenguajes de marcado puede ser editado por un usuario con un sencillo **editor de textos**, aunque existen programas más sofisticados que faciliten el trabajo.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>IMÁGEN</title>
<style>

img{margin:20px;
border: 50px solid black;
border-image:url("fondo.jpeg") 100 repeat;
box-shadow:0px 0px 50px black;
}
</style>
</head>
<body>


</body>
</html>
```

- ➔ **Facilidad de procesamiento.**

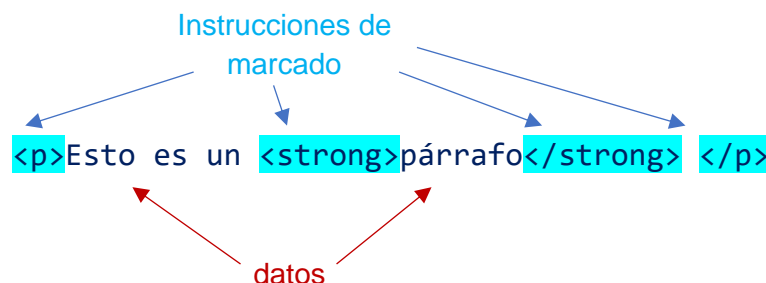
Una de las principales ventajas de este tipo de codificación es que **puede ser interpretada directamente**, dado que son archivos de texto plano. Esto es una ventaja evidente respecto a los sistemas de archivos binarios, que requieren siempre de un programa intermediario para trabajar con ellos.

Sólo es necesario un navegador.



- ➔ **Compacidad:** marcas + contenido

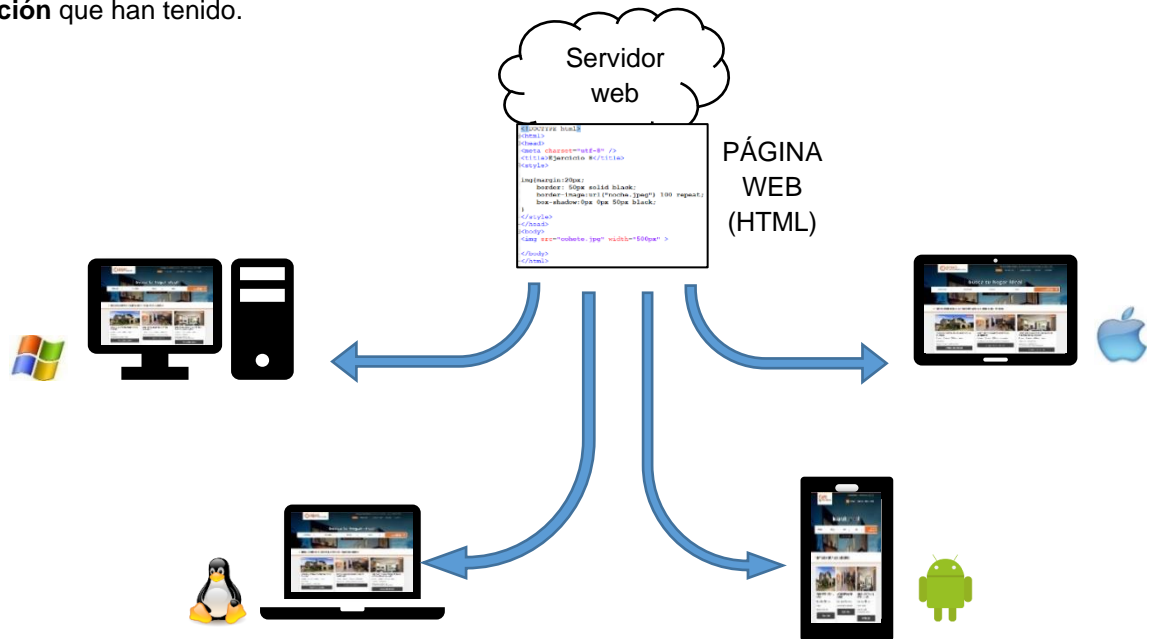
Las instrucciones de marcado se entremezclan con el propio contenido en un único archivo o flujo de datos.



- ➔ **Flexibilidad:** se puede compatibilizar con otros lenguajes, por ejemplo Javascript y PHP.

## → Independencia del dispositivo y sistema operativo

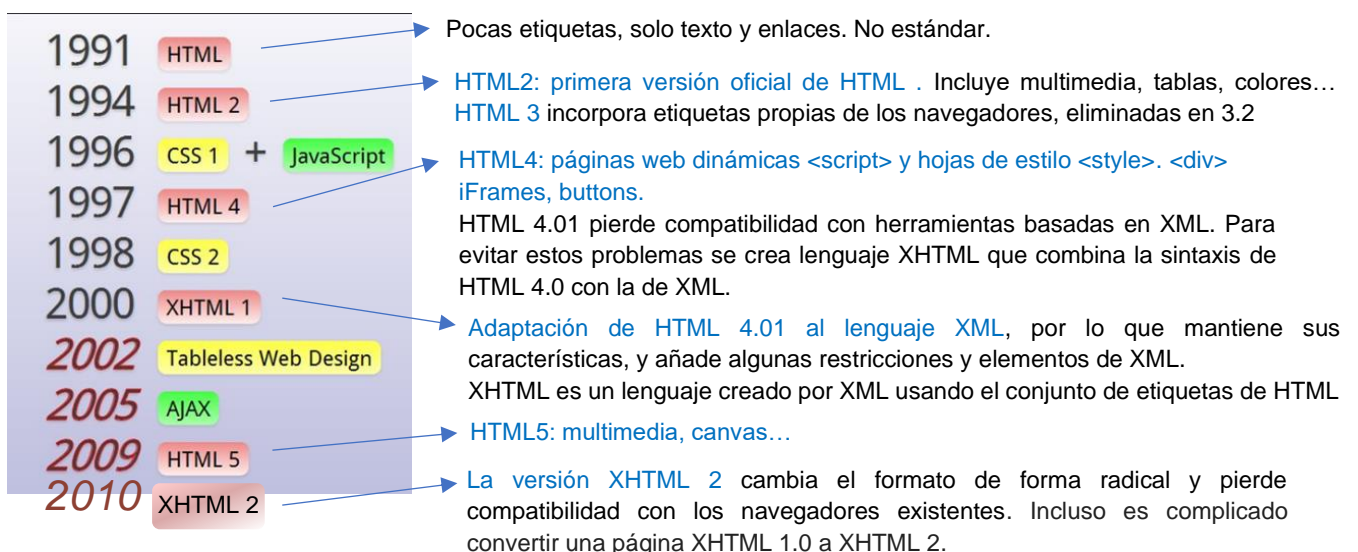
Al tratarse solamente de texto, los documentos son **independientes de la plataforma**, sistema operativo o programa con el que fueron creados. Es una de las razones fundamentales de la gran **aceptación** que han tenido.



## 1.2 Versiones

HTML está basado en SGML, resolviendo su problema de complejidad al utilizar solo un conjunto reducido de etiquetas apropiadas para la realización de documentos relativamente simples.

El origen de HTML fue un sistema de hipertexto: permite crear documentos interactivos que proporcionan información adicional mediante "**hiperenlaces**", también llamados "**enlaces**", que relacionan dos recursos. Dichos recursos pueden ser otras páginas web, imágenes, documentos o archivos.



La tendencia es HTML5 en detrimento de XHTML.

## ¿Dónde encaja CSS dentro de los lenguajes de marcas en entornos web?

CSS (Cascading Style Sheets) "Hojas de estilo en cascada", no es un lenguaje de marcas, es un [lenguaje de estilos](#). Encaminada a separar contenido y representación. También se podrá adaptar mejor cómo ver la información en función del dispositivo.

### 1.3 Diferencias HTML-XHTML

- Los documentos tienen que estar bien formados
- Los elementos vacíos llevan etiqueta de cierre. Ejemplo `<br></br>` o `<br/>`
- Los elementos no vacíos necesitan la etiqueta de cierre
- Los valores de los atributos tienen que llevar siempre comillas
- La minimización de atributos no es posible (`checked="checked"`)
- Manejo de espacios en blanco en los atributos
- Empleo de scripts (`<script>`) y estilo (`<style>`) en ficheros separados
- Exclusiones de SGML - Elementos que no pueden ser contenidos en otros elementos
- Solo el atributo `id` actúa como identificador
- Atributos con conjunto de valores predefinidos

### 1.4 DOCTYPE

Debido a que los navegadores son capaces de procesar diferentes tipos de archivos, lo primero que se debe hacer en la construcción de un documento HTML es indicar su tipo. Se debe agregar la declaración `<!DOCTYPE>` al comienzo del archivo, sin espacio ni código previo, para indicar al navegador cómo debe interpretar la página web.

Un documento HTML o XHTML puede pertenecer a una versión determinada, y a un estándar o variante dentro de esa versión. La etiqueta DOCTYPE hace referencia al estándar al que pertenece.

➡ **Estricto.** Variante con las normas más estrictas y las restricciones más severas. No pueden utilizar atributos relacionados con el aspecto de los contenidos, por lo que requiere una separación total de código HTML y estilos CSS. No permite el uso de elementos "deprecated".

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

➡ **Transicional.** Variante menos estricta que la anterior, ya que permite el uso de algunos atributos HTML relacionados con el aspecto de los elementos.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

➡ **Frameset.** Variante la utilizan las páginas que están formadas por *frames*, una práctica completamente desaconsejada y que hoy en día sólo utilizan los sitios web obsoletos.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
```

[W3C QA - Recommended list of Doctype declarations you can use in your Web document](#)

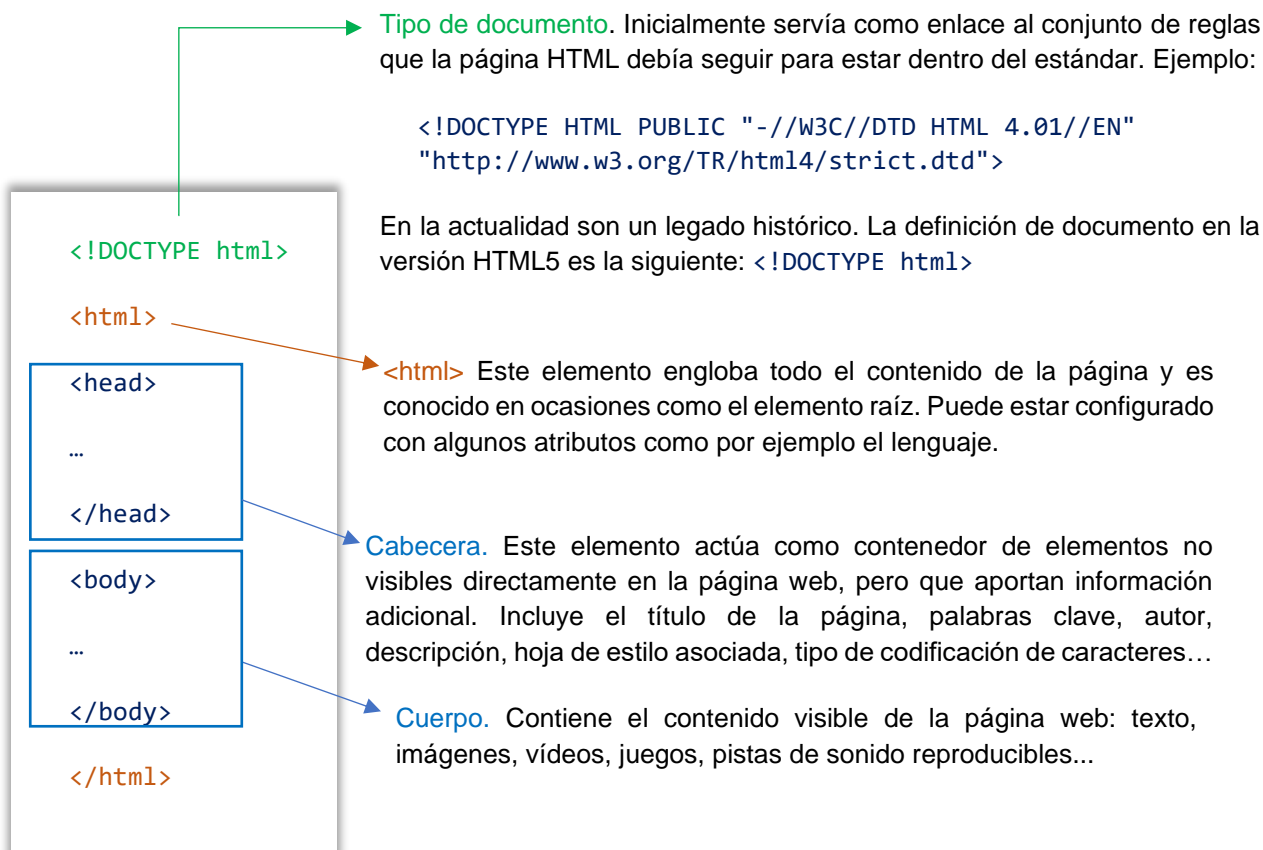
La declaración del tipo de documento en HTML5 es: `<DOCTYPE html>`



## 2. Documento HTML

### 2.1 Estructura del documento

La estructura básica de un documento HTML es la siguiente:



### 2.2 Sintaxis básica

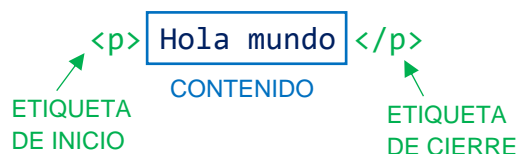
- ▶ No se distinguen mayúsculas o minúsculas. Sin embargo es más apropiado, por razón de estilo, emplear las minúsculas (excepto en DOCTYPE).
- ▶ No es necesario incorporar saltos de línea o tabulaciones, aunque es recomendable por legibilidad del documento.
- ▶ HTML permite introducir comentarios en el código, que no serán representados en el navegador. Estos comentarios se utilizan para aclarar aspectos del contenido, o dejar notas con la finalidad de una futura actualización.

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

- ▶ Los valores de los atributos se entrecomillan.

## 2.3 Etiquetas

Un documento HTML está formado por etiquetas. Esta es la estructura básica de un elemento html:



Una **etiqueta** es una marca establecida por el lenguaje concreto. La cantidad de etiquetas de HTML está limitada a aquellas que están definidas por el lenguaje.

Es el texto incluido entre los símbolos menor que < y mayor que >

Ejemplo, etiqueta h2:      <h2> Encabezado de nivel 2 </h2>

## 2.4 Caracteres especiales

En ocasiones queremos que se visualicen algunos caracteres especiales que el navegador interpreta como parte de HTML y no los representa. Para evitar este problema hay que utilizar un formato alternativo.

A continuación, se muestra una tabla con las equivalencias de representación de algunos caracteres.

Carácter	Representación	Carácter	Representación
<	&lt;	€	&euro;
>	&gt;	ç	&ccedil;
á	&aacute;	Ç	&Ccedil;
Á	&Aacute;	ü	&uuml;
é	&eacute;	Ü	&Uuml;
É	&Eacute;	&	&amp;
í	&iacute;	¿	&iquest;
Í	&Iacute;	¡	&iexcl;
ó	&oacute;	"	&quot;
Ó	&Oacute;	·	&middot;
ú	&uacute;	°	&ordm;
Ú	&Uacute;	ª	&ordf;
ñ	&ntilde;	¬	&not;
Ñ	&Ntilde;	©	&copy;
™	&#153;	®	&reg;

Ver el listado completo: <https://ascii.cl/es/codigos-html.htm>

### Actividad 1: crear estructura básica HTML

- 1.1 Crear el primer documento HTML
- 1.2 Introducir el título
- 1.3 Introducir texto
- 1.4 Caracteres especiales



## 2.5 Validador HTML

Existe un modo de validar que un documento HTML tiene un formato correcto, a través de la url:

<https://validator.w3.org/>

Hay que tener en cuenta que pueden aparecer warnings que sólo son avisos y errores. Entre estos últimos pueden haber errores de elementos obsoletos en HTML5.

Por ejemplo:

- El siguiente mensaje es un aviso, no un error. Aconseja que se incluya el lenguaje a través del atributo “lang”.



**Warning** Consider adding a `lang` attribute to the `html` start tag to declare the language of this document.

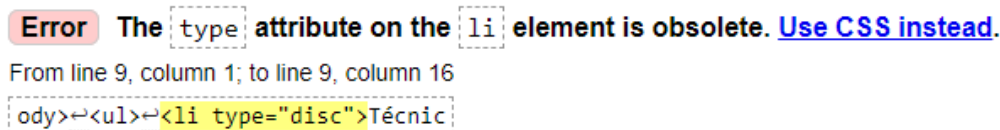
From line 1, column 16; to line 2, column 6

TYPE `html><html><head`

For further guidance, consult [Declaring the overall language of a page](#) and [Choosing language tags](#).

If the HTML checker has misidentified the language of this document, please [file an issue report](#) or [send e-mail to report the problem](#).

- El siguiente mensaje es un error. Indica que utilizar atributos de formato, como “type” dentro de una lista, está obsoleto en HTML5. Como ya hemos comentado anteriormente, el formato debe realizarse siempre con CSS (aunque hemos utilizado algún atributo de formato, con fines didácticos)



**Error** The `type` attribute on the `li` element is obsolete. [Use CSS instead](#).

From line 9, column 1; to line 9, column 16

`ody><ul><li type="disc">Técnic`

- A veces, cuando por ejemplo no ponemos una etiqueta de cierre, puede que el validador no muestre ningún mensaje de error, o que muestre una sucesión de mensajes relativos a problemas con las etiquetas siguientes en el código.

A partir de este momento vamos a validar todos los ejercicios realizados.

Valida el documento de la actividad 1

## 2.6 Definición de la codificación del texto

Dentro de la cabecera se define el **charset**: estándar de codificación alfanumérica que se va a emplear.

- ▶ **Código ASCII** (American Standard Code for Information Exchange) Representa caracteres alfanuméricos con palabras de 8 bits (bytes). Es una tabla que codifica un total de 256 símbolos diferentes, asociándoles a cada uno un número (del 0 al 255). Ejemplo: si escribimos “Hola” en un fichero:

H = 01001000

o = 01101111

l = 01101100

a = 01100001

Se almacenaría 01001000 01101111 01101100 01100001.

- ▶ **Código ANSI** (American National Standards Institute) Representa caracteres alfanuméricos con palabras de 8 bits (bytes).

- ▶ **Unicode**. Para solucionar la limitación de otros códigos con relación a alfabetos no latinos (chino, cirílico...) y otros símbolos, se define un código con mayor capacidad de representación. No usa un número fijo de bits. Define un símbolo como una combinación de caracteres con atributos asociados. Ejemplo:

$$\tilde{n}_{U+00F1} \equiv n_{U+006E} + \circ_{U+0303}$$

Existen varios estándares o códigos (UTF-8, UTF-16, UTF-32)

El más empleado es **UTF-8**, que se refiere a codificación UNICODE, e incluye una gran cantidad de caracteres, incluidas las vocales acentuadas.

Se define:

```
<head>
  <meta charset="UTF-8">
</head>
```

La definición anterior es válida para HTML5. Otras versiones anteriores requerían otro formato como:

```
<meta http-equiv="content-type" content="text/html; utf-8">
```

También existen otros charset frecuentes como por ejemplo el del alfabeto latino ISO-8859-1:

```
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">
<meta http-equiv="X-UA-Compatible" content="IE=edge" charset="iso-8859-1">
```

Listado completo de codificaciones: <https://www.iana.org/assignments/character-sets/character-sets.xhtml>.

## 2.7 Etiquetas básicas de texto

A modo de introducción, a continuación se listan una serie de etiquetas básicas para introducir texto en una página web. [Más adelante](#) se verán otras etiquetas de texto.

- **<p>**

Etiqueta de párrafo. Para distinguir párrafos se haría de la siguiente forma:

```
<p>Párrafo 1</p>
<p>Párrafo 2</p>
```

Los saltos de línea o espacios múltiples que se introducen en el código html no se ven reflejados en el navegador. El navegador sólo aplica formatos en función de las etiquetas.

- **<br> <wbr>**

**<br>** Break. Salto de línea. No tiene etiqueta de cerrado.

No debe emplearse para diferenciar párrafos (**<p>**), sino más bien dentro de los párrafos.

**<wbr>** Word Break. Salto de línea sugerido, en caso de que haya que realizar un corte. No tiene etiqueta de cerrado.

- **<hr>**

Introduce un salto de línea con una línea horizontal separadora.

### Actividad 2: introducir texto

[2.1 Definir el charset](#)

[2.2 Incluir comentarios](#)

[2.3 Párrafos](#)

[2.4 Código fuente sin saltos de línea](#)

[2.5 Introducir espacios](#)

[2.6 Saltos de línea](#)

[2.7 Líneas horizontales](#)

- **<em> <i>**

Ambos ponen el texto en cursiva.

**<em>** (emphasis). Indica que el texto es importante o diferente. Énfasis.

**<i>** (italic). Sólo hace referencia al formato cursivo, no indica nada sobre la importancia del texto.

- **<strong> <b>**

Ambos ponen el texto en negrita.

**<strong>** (strong). Indica que el texto es “fuerte”, es decir, se quiere resaltar su importancia.

**<b>** (bold). Sólo hace referencia al formato en negrita, no indica nada sobre la importancia del texto.

En HTML5 no está recomendado utilizar etiquetas que sólo hagan referencia al formato del texto, como es **<i>** o **<b>**. Lo recomendado es utilizar etiquetas con contenido semántico como **<em>** o **<strong>**, que hablan sobre cómo es el texto, en lugar de como hay que representarlo.

## Actividad 3: formato de texto básico

3.1 Resaltar el texto `<em>`, `<i>`

3.2 Resaltar el texto `<strong>`, `<b>`

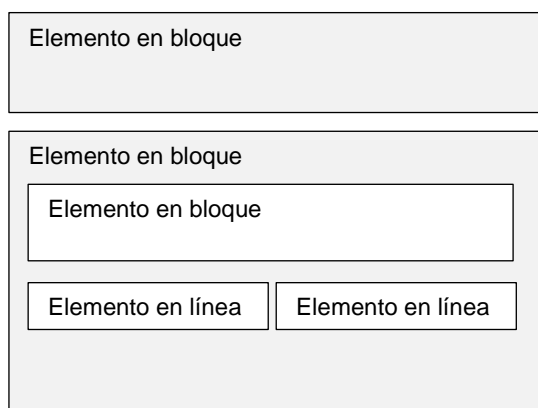
## 2.8 Elementos en línea y en bloque

Hay dos importantes categorías de elementos en HTML: elementos a nivel de bloque y elementos en línea.

→ Los **elementos a nivel de bloque** forman un bloque que ocupa una línea entera. Aparecerán en una nueva línea respecto al contenido anterior y cualquier contenido que venga después también aparecerá en una nueva línea. Los elementos a nivel de bloque tienden a ser elementos estructurales en la página que representan por ejemplo párrafos, listas, menús de navegación, pies de página, etc.

Un elemento a nivel de bloque no estaría anidado dentro de un elemento en línea, pero podría estar anidado dentro de otro elemento a nivel de bloque.

→ Los **elementos en línea** forman un bloque no ocupa una línea entera. Pueden estar contenidos en elementos de bloque. Un elemento en línea no causará que una nueva línea aparezca en el documento: normalmente aparecen dentro de un párrafo o texto, por ejemplo un elemento (hipervínculo) o elementos de énfasis como `<em>` o `<strong>`.



Un elemento en línea puede estar dentro de un elemento en bloque, pero no al revés.

## 2.9 Anidación de elementos

Unos elementos o etiquetas pueden encontrarse dentro de otros.

Es posible que la anidación no sea correcta y se visualice el contenido adecuadamente, gracias a la interpretación que realiza el navegador. Pero es importante realizar una correcta anidación de elementos.

ANIDACIÓN CORRECTA:

```
<em><strong>Texto en negrita y cursiva</strong></em>
```

ANIDACIÓN INCORRECTA:

```
<em><strong>Texto en negrita y cursiva</em></strong>
```

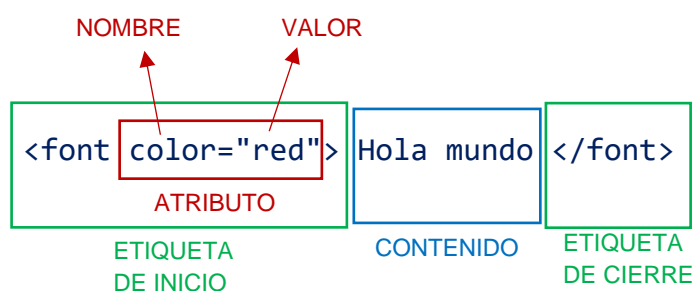
## 2.10 Atributos

Un **atributo** es un modificador de una etiqueta, que permite definir distintas posibilidades de la acción de la etiqueta.

Para cada uno de los atributos hay definido un conjunto de valores que se le puede asignar, si el valor de un atributo no es válido, el navegador le ignora.

### ▪ Formato

Un elemento puede contener atributos con la finalidad de parametrizarlo. Un atributo se identifica por un **nombre y un valor**, y se incluye dentro de la etiqueta de apertura.



Cada una de las etiquetas HTML define los atributos que puede utilizar, aunque algunos de ellos son comunes a muchas etiquetas.

Nota: `<font>` es una etiqueta que hace referencia a la fuente de la letra, es decir, al tipo de letra.

No está recomendado usar `font` como en el ejemplo, sin embargo se muestra a efecto didáctico. El formato de la fuente debe establecerse mediante estilo CSS, no mediante etiqueta HTML.

### ▪ Atributo global

Se refiere a un atributo que puede emplearse con cualquier etiqueta. Algunos ejemplos:

- `id` (asigna un identificador único a un elemento, no debe repetirse)
- `class` (asigna un identificador a un elemento, que puede repetirse)
- `lang` (idioma)
- `style` (estilo css)
- `title` (texto al situar el ratón por encima)
- Atributos de eventos (`onclick`, `onmouseover`, `onmouseout`, `onload`, `onfocus`...) que se combinarán con código javascript.

### ▪ Atributo booleano

En ocasiones es posible ver atributos escritos sin valores: esto está permitido.

Éstos son los llamados atributos booleanos, y pueden tener sólo un valor, que es, generalmente, el mismo que el nombre del atributo. La única alternativa sería no declarar el atributo.

Por ejemplo, el atributo `disabled` puede tener el valor `disabled`. Cuando el nombre del atributo es igual al valor del atributo, puede omitirse el valor.

Es equivalente:

```
<video src="movie.mp4" muted="muted">
<video src="movie.mp4" muted>
```

En ese ejemplo, se emplea una etiqueta video para reproducir un video (movie.mp4). El atributo muted se incorpora para indicar que, por defecto, el video comience a reproducirse sin sonido.

Si no se declara el atributo, no tendría el valor *disabled* sino el valor que defina por defecto, es este ejemplo *enabled* (lo opuesto).

En resumen, los atributos booleanos se utilizan así:

```
<video src="movie.mp4" muted>           Se reproduce sin sonido
<video src="movie.mp4">                 Se reproduce con sonido
```

#### ▪ Comillas en el valor de los atributos

El valor puede ponerse entre comillas simples o dobles, o incluso sin comillas. A veces un navegador concreto acepta unas u otras. Sin embargo, **se recomienda usar comillas siempre**.

```
nombre_atributo="valor"
nombre_atributo='valor'
```

Al observar diferentes páginas web, es posible encontrar todo tipo de estilos de etiquetado mal realizados, incluyendo valores de atributos sin comillas. Esto se permite en ciertas circunstancias, pero interrumpirá la edición en otras. Ejemplo:

```
<a></a> Etiqueta para insertar hipervínculos o enlaces en la página.
href    Atributo de <a> que indica a dónde debe llevar el enlace al pulsarlo.
title   Atributo de <a> que indica el texto que se va a ver si se sitúa el ratón encima, sin pulsar
```

El formato correcto es utilizar comillas en todos los atributos:

```
<a href='https://www.google.es/' title='buscar'>busca</a>
```

Si se quitan las comillas en algunos casos también funcionará:

```
<a href=https://www.google.es/ title=buscar>busca</a>
```

Sin embargo, el no utilizar comillas puede dar problemas. Dependerá de la interpretación del navegador. Por ejemplo si el título es “mi búsqueda” en lugar de “búsqueda”:

```
<a href=https://www.google.es/ title=mi búsqueda>busca</a>
```

El problema en el último caso es que el navegador va a considerar que el atributo title va a tener valor “mi”, y que “búsqueda” es un atributo booleano.

```
href= https://www.google.es/
title=mi
```

Diagrama de la etiqueta HTML: `<a href=https://www.google.es/ title=mi búsqueda>busca</a>`

- Atributo href** (azul): `href`
- valor** (azul): `= https://www.google.es/`
- Atributo title** (verde): `title`
- valor** (verde): `= mi`
- Atributo búsqueda (booleano)** (naranja): `búsqueda`



## Actividad 4: elementos htm

4.1. Elementos en bloque y en línea

4.2. Anidación de etiquetas

4.3. Integrar atributos

4.4. Uso de comillas en los atributos

Ejercicios: formato básico de texto. Ejercicios 1 y 2.

## 2.11 <head> Cabecera del documento

El contenido de la cabecera es lo que se encuentra entre las etiquetas <head></head>.

A continuación se describen algunos elementos que se utilizan en la cabecera del documento. Como se ha comentado anteriormente, no serán elementos con contenido visible directamente.

- **<charset>**

Ver [definición de la codificación del texto](#)

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

- **<title>**

Se emplea para especificar el título de la página. Se visualiza en la pestaña del navegador, no en el contenido.

No debe confundirse con <h1>, que es un elemento dentro de <body>

Ejemplo:

```
<title>Mi web</title>
```

Otra función de esta etiqueta es la de proponer un nombre a los marcadores, cuando el usuario desea guardar un marcador para la página.

## Actividad 5: utilizar <title>

5.1 Comparar el título <title> con <h1>

5.2 Añadir la página a marcadores

- **<meta>**

Se emplea para especificar metadatos, es decir, información sobre el documento en sí. No requiere cierre. Pueden insertarse varias etiquetas <meta> en un mismo documento.

Vamos a ver tres utilidades para <meta>:

1) **<meta> para especificar el charset**

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

Especifica la [codificación del texto](#).

## 2) <meta> para dar información sobre la página a usuarios y buscadores.

La información que se introduzca en <meta> puede ser utilizada por los **buscadores**.

Los buscadores consultan la información de esa de las páginas, buscando coincidencias con lo que el usuario pretende encontrar. Hay que comentar que en la actualidad muchos buscadores ya no tienen en cuenta esta información, debido a que hay mal uso de ella.

A través de esta etiqueta pueden especificarse los **atributos name y content**. El atributo name indica el tipo de información, y el atributo content indica el valor de dicha información.

Formato: `<meta name='...' content='... '>`

Para indicar el tipo de información (el valor de name y de content) podemos utilizar cualquier palabra, como puede ser "Autor", "Palabras clave", "Descripción", etc. Pero debido a que la mayoría de buscadores están en inglés, es preferible que el tipo de información se especifique en inglés.

Los tipos de información más utilizados son los siguientes:

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

Muchas aplicaciones y plataformas utilizan los metadatos para realizar acciones diseñadas por ellas mismas.

## 3) <meta> para indicar una acción que debe realizar el navegador

La etiqueta <meta> también se utiliza para indicarle al navegador alguna información o alguna acción que debe realizar. En este caso se utiliza el atributo http-equiv, en lugar del atributo name.

Formato: `<meta http-equiv='...' content='... '>`

Ejemplos:

```
<meta http-equiv="Refresh" content="30">
```

Indica que se debe refrescar la página cada 30 segundos automáticamente

```
<meta http-equiv="Refresh" content="5; URL=http://www.dam.com/index.htm">
```

Indica que pasados 5 segundos se debe recargar la página, pero cargando la nueva dirección indicada. Esto es útil cuando la dirección web ha cambiado y se quiere redirigir, de forma automática, a los usuarios que accedan a la página antigua.

## Actividad 6: incluir metadatos

### 6.1 Añadir charset y datos para buscadores

### 6.2 Recargar la página de forma automática

### 6.3 Redireccionar la página de forma automática

### ▪ <link>

Este elemento especifica la relación entre el documento y un recurso externo, generalmente usado para cargar archivos CSS, por ejemplo: `<link rel="stylesheet" href="miestilo.css">`

Cuando se asocia un icono a una web (favicon), éste aparecerá en múltiples sitios en los que se haga referencia a la página, por ejemplo se verá en la pestaña del navegador y en los marcadores.

Para ello, en la cabecera del documento hay que incluir un enlace con el formato:

<code>&lt;link</code>	<code>rel='shortcut icon'</code>	<code>href='...'</code>	<code>type='...'</code>	<code>&gt;</code>
	Indica que es un icono	Ruta de la imagen	Tipo de imagen	

Ejemplos:

```
<link rel="shortcut icon" href="mi_icono.png" type='image/png'>
<link rel="shortcut icon" href="mi_icono.jpg" type='image/jpg'>
```

## Actividad 7: asociar un icono

### 7.1 Asociar un icono en varios formatos

## 2.12 Idioma del documento

### ▪ Idioma global

El lenguaje de la página completa se puede especificar como en el atributo lang (language) del elemento <html>.

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

Utilidades:

- ▶ Hace más eficiente la búsqueda de los buscadores.
- ▶ Si se emplea una utilidad de lectura del texto, o una extensión del navegador, es necesario indicar el idioma para que la pronunciación sea correcta (por ejemplo la palabra “six” existe en inglés y francés, pero tiene distinta pronunciación).
- ▶ Se pueden aplicar diferentes formatos en función del idioma del texto (se verá con CSS)

Lista de idiomas disponibles: [https://www.w3schools.com/tags/ref\\_language\\_codes.asp](https://www.w3schools.com/tags/ref_language_codes.asp)

### ▪ Idioma parcial

Al definir el atributo “lang” en la etiqueta html, se está definiendo el lenguaje para todo el documento. Sin embargo, puede ser necesario que un elemento en concreto se asocie a otro lenguaje, por ejemplo, si se hace una cita a una frase en otro idioma. Por ello el atributo “lang” puede ser incorporado a otros elementos. Ejemplo:

```
<p lang="jp"> が熱い。 </p>
```

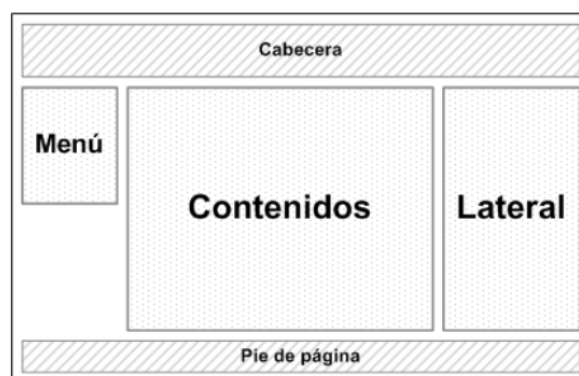
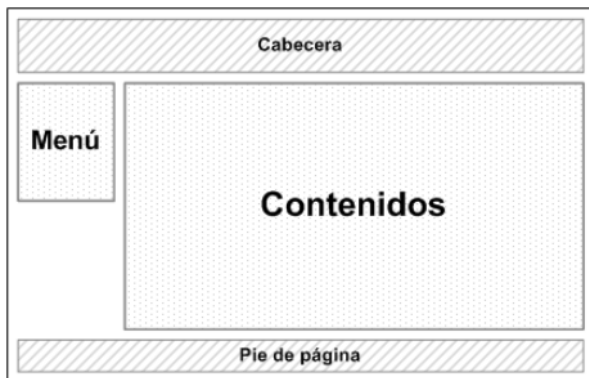
Aunque es un atributo global y se puede utilizar en cualquier elemento HTML, solo tiene efecto sobre elementos que contengan texto. Esto quiere decir que en elementos que no contienen texto el atributo lang es válido pero no tiene efecto.

## Actividad 8: asociar un lenguaje

### 8.1 Asociar un lenguaje a nivel global y parcial

## 2.13 Estructura de página

Se emplean distintos elementos para diseñar la estructura de la página: cabecera, menú, contenido principal, barra lateral, pie de página.



Ejemplo con página estructurada con cabecera, menú superior, barra lateral izquierda, barra lateral derecha, zona central y pie de página.

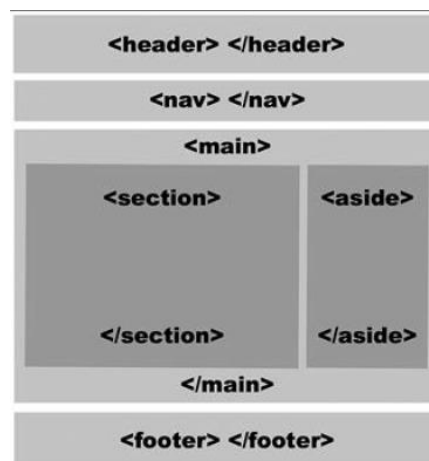


Ejemplo con página estructurada con cabecera, menú superior, zona central y pie de página.

Las siguientes etiquetas HTML se emplean para definir la estructura de la página.

Estos elementos no solo ayudan a los desarrolladores a crear el documento, sino que además informan al navegador sobre el propósito de cada sección.

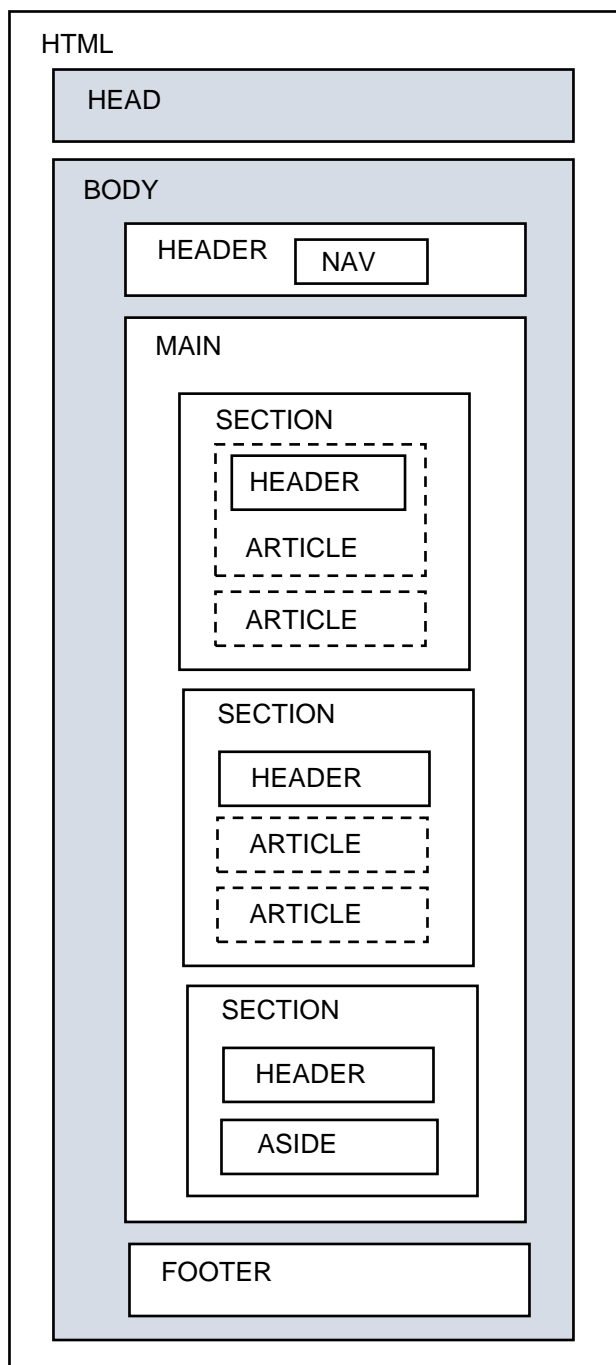
Son flexibles y se pueden implementar en diferentes partes del diseño.



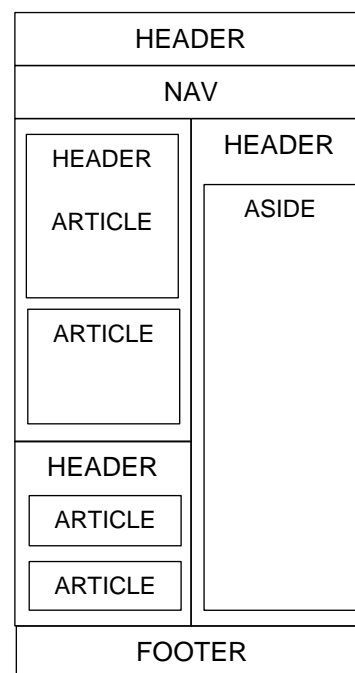
- **<header>**  
Cabecera. Se incluye el logo, el nombre del sitio, los subtítulos y una descripción breve del sitio o página web  
  
Puede ser la cabecera general (cuelga de <body>, pero no confundir con <head>), o bien puede ser la cabecera de otro elemento (cuelga de <article> o de <section>)
- **<nav>**  
Menú de navegación
- **<main>**  
Contenido principal
- **<aside>**  
Sidebar, menú lateral o contenido relacionado
- **<article>**  
Contenido con entidad propia. Puede representar una columna.  
  
Puede tener dentro un <header> y varias <section>
- **<section>**  
Contenido con entidad propia. Puede representar una columna.  
  
Puede tener dentro un <header> y varios <article>
- **<footer>**  
Pie de página. También denominado “barra institucional”, ya que es donde se muestra información general acerca del sitio web, el autor, la compañía, los enlaces relacionados con reglas de uso, términos y condiciones, el mapa del sitio, etc.

Estas etiquetas tienen justificación semántica, es decir, se eligen dependiendo del significado del texto. Es para dar sentido. **Pero en sí mismas no condicionan las secciones físicamente** (no crean barra lateral, pie de página, etc). **Es necesario darles formato con CSS para colocarlas en su posición.**

Ejemplo. Estructura:



Posible visualización en pantalla:



Es habitual encontrar páginas estructuradas con `<div>`, que es una etiqueta que no aporta información sobre su contenido.

## Actividad 9: estructurar la página

### 9.1 Uso de etiquetas de estructura

### 9.2 Bordes

### 9.3 Posicionamiento

### 3. Encabezados

En un documento existen diferentes encabezados, títulos y párrafos, que determinan la distinta importancia que se quiere dar a cada apartado.



- `<h1>` `<h2>` `<h3>` `<h4>` `<h5>` `<h6>`

Los distintos encabezados se consiguen con las siguientes etiquetas, en orden decreciente de importancia:

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

Se recomienda que haya un solo encabezado `<h1>`

No hay que confundir los encabezados con el título `<title>` de la página (no es visible en el contenido de la página)

#### Actividad 10: encabezados

10.1 Añadir encabezados

10.2 Resaltar texto

## 4. Etiquetas de texto

### 4.1 Etiquetas de formato

El formato de texto se debe realizar con CSS. De esta forma, se puede separar el contenido del documento de su formato de presentación. Sin embargo, vamos a repasar aquí algunas etiquetas de formato para poder identificarlas.

- `<i>`  
Italic. Letra en *cursiva*
- `<b>`  
Bold. Letra en **negrita**.
- `<u>`  
Underline. El texto aparece subrayado.
- `<sub>`  
Subíndice. Aparece como <sub>subíndice</sub>
- `<sup>`  
Superíndice. Aparece como <sup>superíndice</sup>
- `<strike>`  
Aparece el texto ~~tachado~~.
- `<pre>`  
Se respeta el formato original del texto. No hay que abusar de su uso. Muy útil para introducir códigos de programación.

Ejemplo:

```
<pre>
Texto con           tabulaciones

saltos
de línea y

espacios
</pre>
```

Visualización:

```
Texto con           tabulaciones

saltos
de línea y

espacios
```



Los elementos `<pre>` son especiales, ya que los navegadores les aplican las siguientes reglas:

- ▶ Mantienen todos los espacios en blanco (tabuladores, espacios y nuevas líneas)
- ▶ Muestra el texto con un tipo de letra especial, denominado "*de ancho fijo*", ya que todas sus letras son de la misma anchura
- ▶ No se ajusta la longitud de las líneas (las líneas largas producen un *scroll* horizontal en la ventana del navegador)

Esta última característica diferencia por completo a los párrafos de los elementos `<pre>`. Los navegadores ajustan la anchura de los párrafos de texto para que ocupen todo el tamaño de la ventana. Sin embargo, los elementos `<pre>` se muestran tal y como son originalmente, por lo que una línea muy larga dentro de un elemento `<pre>` provoca que la anchura de la página sea superior a la anchura de la ventana del navegador.

## 4.2 Etiquetas semánticas (de significado)

Estas etiquetas nos indican la utilidad que tiene el texto. Pueden modificar o no el formato.

- **`<em>`**

Emphasis. Indica que el texto es importante o diferente. Se ve en *cursiva*.

- **`<strong>`**

Strong. Indica que el texto es "fuerte", es decir, se quiere resaltar su importancia. Se ve en **negrita**.

- **`<code>`**

Código. Delimita el texto considerado un fragmento de código. Cambia el formato del texto de forma similar a `<pre>`, sin embargo no respeta espacios, tabulaciones ni saltos de línea. Suele ir dentro de `<pre>` (no al revés). Ejemplo:

```
<pre><code>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

struct datos_personales//define la estructura
{
    int posicion;
    char nombre[50];
    int edad;
    struct datos_personales *sig; //puntero a siguiente elemento de la lista
};
struct datos_personales *agenda = NULL; //puntero al primer elemento
</code></pre>
```

```
#include
#include
#include

struct datos_personales //define la estructura
{
    int posicion;
    char nombre[50];
    int edad;
    struct datos_personales *sig; //puntero a siguiente elemento de la lista
};
struct datos_personales *agenda = NULL; //puntero al primer elemento
```

- **<abbr>**

Abreviatura. En versiones anteriores de HTML se empleaba <acronym>.

La organización `<abbr title="International Organization for Standardization">ISO</abbr>` fue fundada el el 23 de febrero de 1947.

ISO fue fundada el el 23 de febrero de 1947.

International Organization for Standardization

- **<mark>**

Marca texto relevante.

- **<small>**

Este elemento representa letra pequeña, como declaraciones legales, descargos, etc.

`<small>&copy; Derechos Reservados</small>`

- **<address>**

Este elemento representa información de contacto. Se implementa con frecuencia dentro del pie de página para definir la dirección de la empresa o el sitio web.

`<address>Calle Suecia, 15, 3ºB</address>`

- **<time>**

Este elemento representa una fecha en formato legible para el usuario.

`<p>Abierto hasta las <time>22:00</time>.</p>`

Opcionalmente puede utilizarse el atributo `datetime` para especificar un valor en formato máquina.

`<p>Habrá una fiesta en <time datetime="2021-12-25 20:00">Navidad</time>.</p>`

## 4.3 Etiquetas para citas

- **<q>**

Quote. Especifica una sección de texto citado, tal y como es. Aparece entrecomillado (las comillas las agrega el navegador). Es un elemento en línea.

Ejemplo:

Dicen que `<q>Hacer ejercicio trae muchos beneficios</q>`

Dicen que "Hacer ejercicio trae muchos beneficios"

- **<blockquote>**

Similar a `<q>`, pero es un elemento en bloque. Especifica una sección de texto citado, tal y como es. No aparece entrecomillado.

Mediante el **atributo cite** se puede indicar la fuente del texto (esta fuente no es visible en la página, se incluye sólo para dar información). No crea un enlace a la fuente.

Ejemplo:

```
Dicen que <blockquote cite="http://www.bienysaludable.com/ejercicio/5-beneficios-inesperados-de-hacer-ejercicio">Hacer ejercicio trae muchos beneficios</blockquote>
```

Dicen que

Hacer ejercicio trae muchos beneficios

Se está citando una frase literal, que se ha tomado de una página web.

- **<cite>**

Se utiliza para citar una obra. Pone la letra en cursiva. No crea ningún enlace.

Ejemplo:

En su libro <cite>Planilandia</cite>, Edwin Abbott Abbott describe un mundo de múltiples dimensiones.

En su libro *Planilandia*, Edwin Abbott Abbott describe un mundo de múltiples dimensiones.

## 5. Etiquetas de delimitación

No tienen un significado especial, se utilizan solo para delimitar partes del texto y aplicar después formato mediante CSS. En HTML5 se aconseja limitar su uso y, en su lugar, emplear las etiquetas semánticas correspondientes al texto en concreto.

- **<div>**

Es un elemento en bloque.

- **<span>**

Es un elemento en línea.

Ejemplo:

```
<div>...<span>...</span><span>...</span><span>...</span></div>
<span>...</span><span>...</span>
<div>...</div>
```

### Actividad 11: Etiquetas de texto

11.1 Utilización de etiquetas de texto

11.2 Abreviaturas y citas

## 6. Listas

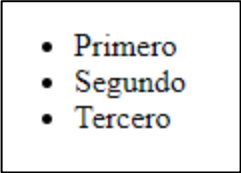
### 6.1 Listas ordenadas y no ordenadas

- `<ul>` `<li>`

`<ul>` unordered list      `<li>` list item

Se utilizan estas etiquetas para crear listas no numeradas. El formato es:

```
<ul>
  <li>Primero</li>
  <li>Segundo</li>
  <li>Tercero</li>
</ul>
```



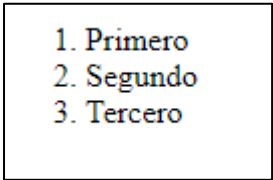
- Primero
- Segundo
- Tercero

- `<ol>` `<li>`

`<ol>` ordered list      `<li>` list item

Se utilizan estas etiquetas para crear listas numeradas. El formato es:

```
<ol>
  <li>Primero</li>
  <li>Segundo</li>
  <li>Tercero</li>
</ol>
```



1. Primero
2. Segundo
3. Tercero

### 6.2 Atributos de listas

- **Type**

En la lista **no ordenada** determina el tipo de punto que se empleará. A veces el símbolo en concreto que aparece depende del navegador, y no coincide en unos y otros.

ATRIBUTO	TIPO DE PUNTO	EJEMPLO	RESULTADO
type="circle"	Círculo	<pre>&lt;ul type="circle"&gt;   &lt;li&gt;Primera&lt;/li&gt;   &lt;li&gt;Segunda&lt;/li&gt;   &lt;li&gt;Tercera&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>○ Primera</li> <li>○ Segunda</li> <li>○ Tercera</li> </ul>
type="square"	cuadrado	<pre>&lt;ul type="square"&gt;   &lt;li&gt;Primera&lt;/li&gt;   &lt;li&gt;Segunda&lt;/li&gt;   &lt;li&gt;Tercera&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>▪ Primera</li> <li>▪ Segunda</li> <li>▪ Tercera</li> </ul>
type="disc"	disco	<pre>&lt;ul type="disc"&gt;   &lt;li&gt;Primera&lt;/li&gt;   &lt;li&gt;Segunda&lt;/li&gt;   &lt;li&gt;Tercera&lt;/li&gt; &lt;/ul&gt;</pre>	<ul style="list-style-type: none"> <li>• Primera</li> <li>• Segunda</li> <li>• Tercera</li> </ul>

Es posible incluir este atributo en <ul> o en <li>.

```
<ul>
<li type="circle">Primera</li>
<li type="square">Segunda</li>
<li type="disc">Tercera</li>
</ul>
```

- Primera
- Segunda
- Tercera

El atributo type es un atributo de formato, y como tal no está recomendado su uso. En su lugar se emplearía CSS (ejemplo list-style-type:disc)

En la **lista ordenada** determina el tipo de orden.

ATRIBUTO	TIPO DE ORDEN	EJEMPLO	RESULTADO
type="1"	Números	<pre>&lt;ol type="1"&gt; &lt;li&gt;Primera&lt;/li&gt; &lt;li&gt;Segunda&lt;/li&gt; &lt;li&gt;Tercera&lt;/li&gt; &lt;/ol&gt;</pre>	1. Primera 2. Segunda 3. Tercera
type="A"	Letras mayúsculas	<pre>&lt;ol type="A"&gt; &lt;li&gt;Primera&lt;/li&gt; &lt;li&gt;Segunda&lt;/li&gt; &lt;li&gt;Tercera&lt;/li&gt; &lt;/ol&gt;</pre>	A. Primera B. Segunda C. Tercera
type="a"	Letras minúsculas	<pre>&lt;ol type="a"&gt; &lt;li&gt;Primera&lt;/li&gt; &lt;li&gt;Segunda&lt;/li&gt; &lt;li&gt;Tercera&lt;/li&gt; &lt;/ol&gt;</pre>	a. Primera b. Segunda c. Tercera
type="I"	Números romanos	<pre>&lt;ol type="I"&gt; &lt;li&gt;Primera&lt;/li&gt; &lt;li&gt;Segunda&lt;/li&gt; &lt;li&gt;Tercera&lt;/li&gt; &lt;/ol&gt;</pre>	I. Primera II. Segunda III. Tercera
type="i"	Números romanos en minúscula	<pre>&lt;ol type="i"&gt; &lt;li&gt;Primera&lt;/li&gt; &lt;li&gt;Segunda&lt;/li&gt; &lt;li&gt;Tercera&lt;/li&gt; &lt;/ol&gt;</pre>	i. Primera ii. Segunda iii. Tercera

#### ▪ Start

Indica el valor inicial de orden en una lista ordenada

```
<ol start="5">
<li>Primera</li>
<li>Segunda</li>
<li>Tercera</li>
</ol>
```

- 5. Primera
- 6. Segunda
- 7. Tercera

### ▪ Value

Indica el número de orden en una lista ordenada.

```
<ol>
  <li value="20">Primera</li>
  <li>Segunda</li>
  <li>Tercera</li>
</ol>
```

20. Primera  
21. Segunda  
22. Tercera

```
<ol>
  <li value="20">Primera</li>
  <li value="3">Segunda</li>
  <li>Tercera</li>
</ol>
```

20. Primera  
3. Segunda  
4. Tercera

### ▪ Reversed

Indica orden inverso en una lista ordenada

```
<ol reversed>
  <li>Primera</li>
  <li>Segunda</li>
  <li>Tercera</li>
</ol>
```

3. Primera  
2. Segunda  
1. Tercera

## 6.3 Listas de descripción de términos

### ▪ <dl><dt><dd>

dl (descripción list)

dt (descripción term) title

dd (description definition)

```
<dl>
  <dt>Primera</dt>
  <dd>Explicación 1</dd>
  <dd>Explicación 2</dd>
  <dt>Segunda</dt>
  <dd>Explicación 1</dd>
  <dd>Explicación 2</dd>
</dl>
```

Primera  
Explicación 1  
Explicación 2  
Segunda  
Explicación 1  
Explicación 2

## Actividad 12: Listas ordenadas y no ordenadas

12.1 Estructurar una receta

12.2 Uso de atributos en una lista ordenada

12.3 Lista de descripción

Ejercicio: listas. Ejercicio 3.

## 7. DOM (Document Object Model)

### 7.1 Árbol de nodos

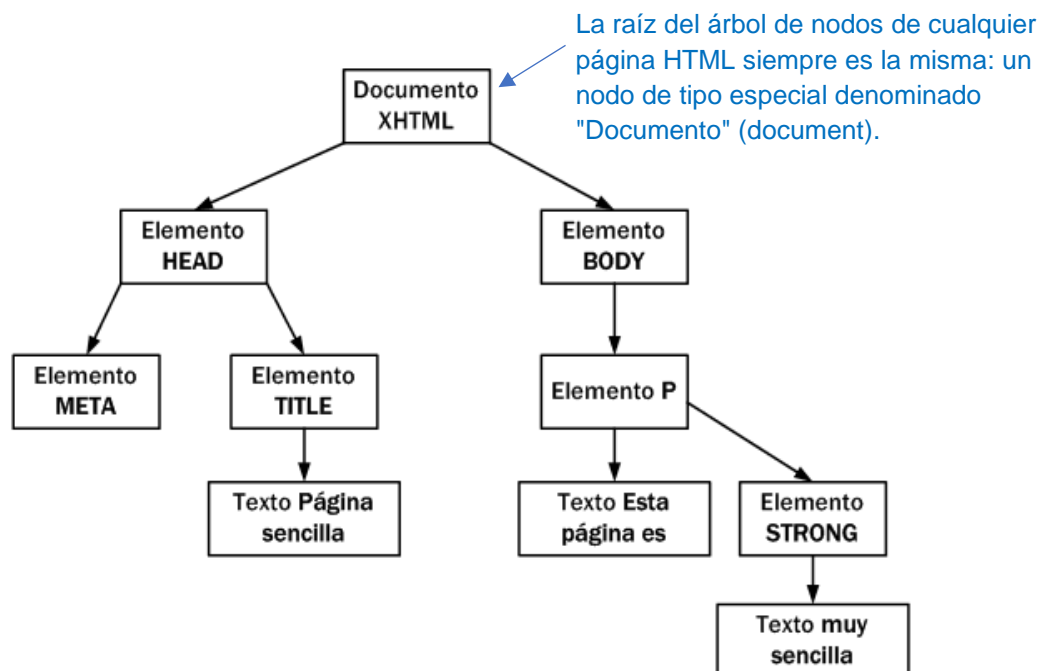
**DOM** es el Modelo de Objetos del Documento. Se crea cuando se interpreta el documento HTML, es decir en el navegador web. El DOM tiene una estructura modular en la que se incluyen elementos HTML, eventos y hojas de estilo, entre otros.

Representa una estructura jerárquica de objetos (los elementos HTML).

DOM transforma los documentos HTML en un conjunto de **nodos**, que están interconectados y que representan los contenidos de las páginas web y las relaciones entre ellos. Por su aspecto, la unión de todos los nodos se llama "**árbol de nodos**".

Por ejemplo, la siguiente página web sencilla, se transforma en el siguiente árbol de nodos:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Página sencilla</title>
</head>
<body>
<p>Esta página es <strong>muy sencilla</strong></p>
</body>
</html>
```


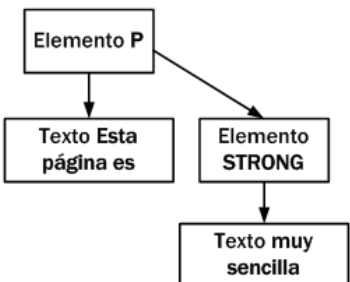


A partir del nodo raíz "Documento", cada elemento o etiqueta HTML se transforma en un nodo de tipo "Elemento". La conversión de etiquetas en nodos se realiza de forma **jerárquica**.

De esta forma, del nodo raíz solamente pueden derivar los nodos head y body. A partir de esta derivación inicial, **cada etiqueta HTML se transforma en un nodo** que deriva del nodo correspondiente a su "etiqueta padre".

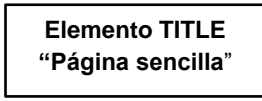
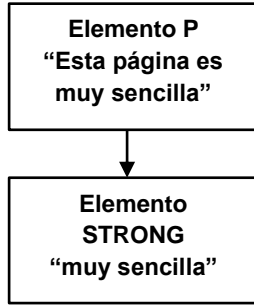
La transformación de las etiquetas HTML habituales genera dos nodos: el primero es el nodo de tipo “Elemento” (correspondiente a la propia etiqueta HTML) y el segundo es un nodo de tipo “Texto” que contiene el texto encerrado por esa etiqueta HTML.

Ejemplos:

ELEMENTO HTML	NODOS GENERADOS
<code>&lt;title&gt;Página sencilla&lt;/title&gt;</code>	 <pre> graph TD     A[Elemento TITLE] --&gt; B[Texto Página sencilla]           </pre>
<code>&lt;p&gt;Esta página es &lt;strong&gt;muy sencilla&lt;/strong&gt;&lt;/p&gt;</code>	 <pre> graph TD     A[Elemento P] --&gt; B[Texto Esta página es]     A --&gt; C[Elemento STRONG]     C --&gt; D[Texto muy sencilla]           </pre>

Como se puede suponer, las páginas HTML habituales producen árboles con miles de nodos. Aun así, el proceso de transformación es rápido y automático.

Como simplificación, se puede considerar que cada elemento HTML genera un único nodo que es la etiqueta HTML con su contenido.

ELEMENTO HTML	NODOS GENERADOS
<code>&lt;title&gt;Página sencilla&lt;/title&gt;</code>	 <pre> graph TD     A["Elemento TITLE \"Página sencilla\""]           </pre>
<code>&lt;p&gt;Esta página es &lt;strong&gt;muy sencilla&lt;/strong&gt;&lt;/p&gt;</code>	 <pre> graph TD     A["Elemento P \"Esta página es muy sencilla\""] --&gt; B["Elemento STRONG \"muy sencilla\""]           </pre>



Desde un navegador puede visualizarse el DOM como una estructura jerárquica



```
...<!doctype html> == $0
<html>
  <head>...</head>
  <body>
    <ul>
      <li type="disc">Técnicas</li>
        <ol start="10" type="a" reversed>
          <li>Óleo</li>
          <li>Acrílico</li>
          <li>Tinta</li>
          <li>Carboncillo</li>
          <li>Grafito</li>
          <li>Pastel</li>
          <li>Acuarela</li>
        </ol>
      <li type="circle">Material</li>
        <ol type="A">...</ol>
          <li type="square">Motivo</li>
          <ol start="2" type="I">...</ol>
        </ol>
    </ul>
  </body>
</html>
```

El DOM es accesible desde javascript y permite crear páginas web dinámicas, al interactuar con el usuario.  
(Ver ejemplos)

### Actividad 13: DOM y validador

- 13.1. Revisar el DOM y validar un fichero html
- 13.2. Otras etiquetas sin cierre
- 13.3 Corregir errores

## 8. Hiperenlaces

Los **hipervínculos**, **hiperenlaces**, **enlaces**, **vínculos** o **hipertexto** son elementos fundamentales en HTML.

De hecho, las letras "HT" de la sigla HTML significan "hipertexto" (hypertext en inglés), por lo que el significado completo de HTML podría traducirse como "lenguaje de marcado para hipertexto". El elemento principal del hipertexto es el "hiperenlace", también llamado "enlace web" o simplemente "enlace".

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque muchos enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos.

Una característica que no se suele tener en cuenta en los enlaces es que están formados por dos extremos y un sentido. En otras palabras, el enlace comienza en un recurso y apunta hacia otro recurso. Cada uno de los dos extremos se llaman "**anchors**" en inglés, que se puede traducir literalmente como "anclas".



Los enlaces se identifican por la URL, es decir la "ruta" hacia el recurso vinculado.

Cuando se habla de que un enlace está vinculado a un recurso, significa que al pulsarlo se obtendrá dicho recurso. Normalmente un recurso es una página web, es decir, un documento html, pero hay otro tipo de recursos como: imágenes, videos, documentos... (por ejemplo doc.html, doc.png, doc.mp4, doc.pdf...)

### 8.1 URL (*Uniform Resource Locator*)

URL hace referencia a la localización única de cada recurso disponible en Internet (página web, imagen, video, documento...).

De esta forma es posible crear enlaces que apunten de forma inequívoca a un determinado recurso.

No sólo identifican de forma única a cada recurso de Internet, sino que también proporcionan a los navegadores la información necesaria para poder llegar hasta ese recurso.

Ejemplo de URL:



- ▶ **Protocolo** (https://) el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan el protocolo HTTP (Hypertext Transport Protocol) o el HTTPS (Hypertext Transport Protocol Secure).
- ▶ **Servidor** (www.w3.org): simplificando mucho su explicación, se trata del servidor en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre a través del protocolo DNS (Domain Name System)

- ▶ **Ruta** (WAI/design-develop/): *camino* que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder.
- ▶ **Recurso** ([index.html](#)): elemento. Cuando no se indica el documento dentro de la ruta, por defecto se buscará el documento “index.html” o “index.php”)

Las URLs pueden además contener otro tipo de información, por ejemplo:

`https://www.mgh.com:80/documentos/index.html#art`

↙ Puerto TCP/UDP      ↘ Sección (#art1): indica una parte de la página web

### ■ Restricciones en las URLs

Existen caracteres reservados que no pueden estar dentro de las URLs, como por ejemplo :, =, & y /. Por otra parte, aunque desde hace tiempo ya es posible incluir en las URL caracteres de otros idiomas que no sean el inglés, aún no es completamente seguro utilizar estos caracteres en las URL. Si se utilizan letras como ñ, á, é o ç, es posible que algunos navegadores no las interpreten de forma correcta.

Si es necesario incluir estos caracteres reservados y especiales en una URL, se sustituyen por combinaciones de caracteres seguros.

Carácter original	Carácter codificado	Carácter original	Carácter codificado
/	%2F	?	%3F
:	%3A	@	%40
=	%3D	&	%26
"	%22	\	%5C
'	%27	~	%7E
(espacio en blanco)	%20	#	%23
ñ	%F1	Ñ	%D1
á	%E1	Á	%C1
é	%E9	É	%C9
í	%ED	Í	%CD
ó	%F3	Ó	%D3
ú	%FA	Ú	%DA
ç	%E7	Ç	%C7

Ejemplos de URLs problemáticas:

`http://www.ejemplo.com/otoño.html`

URL CORRECTAS:

`http://www.ejemplo.com/oto%F1o.html`

`http://www.ej.com/ruta/nombre página.html`

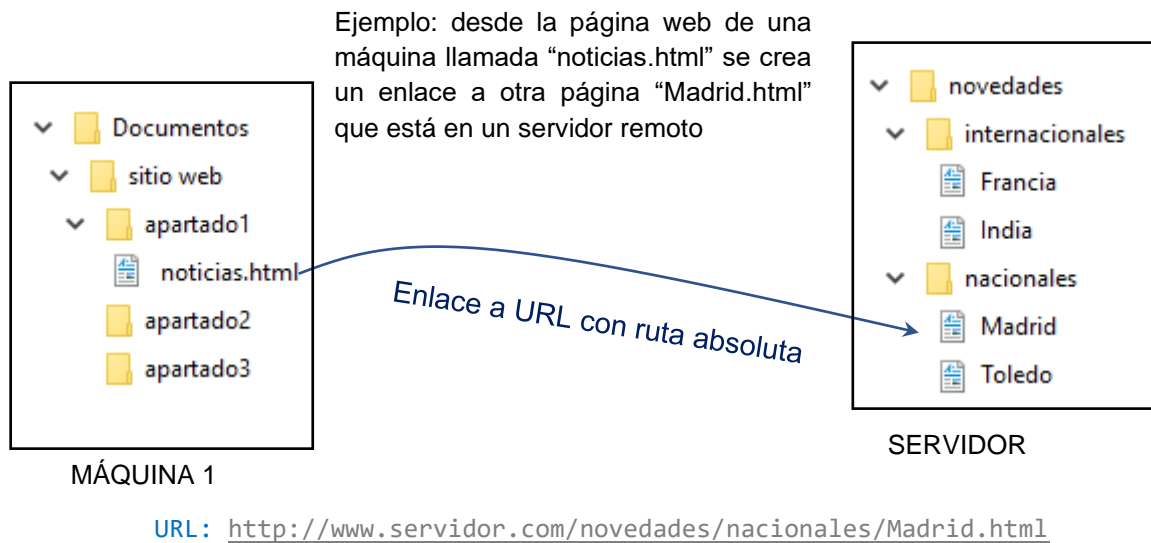
`http://www.ej.com/ruta/nombre%20p%E1gina.html`

## ■ URLs externas e internas

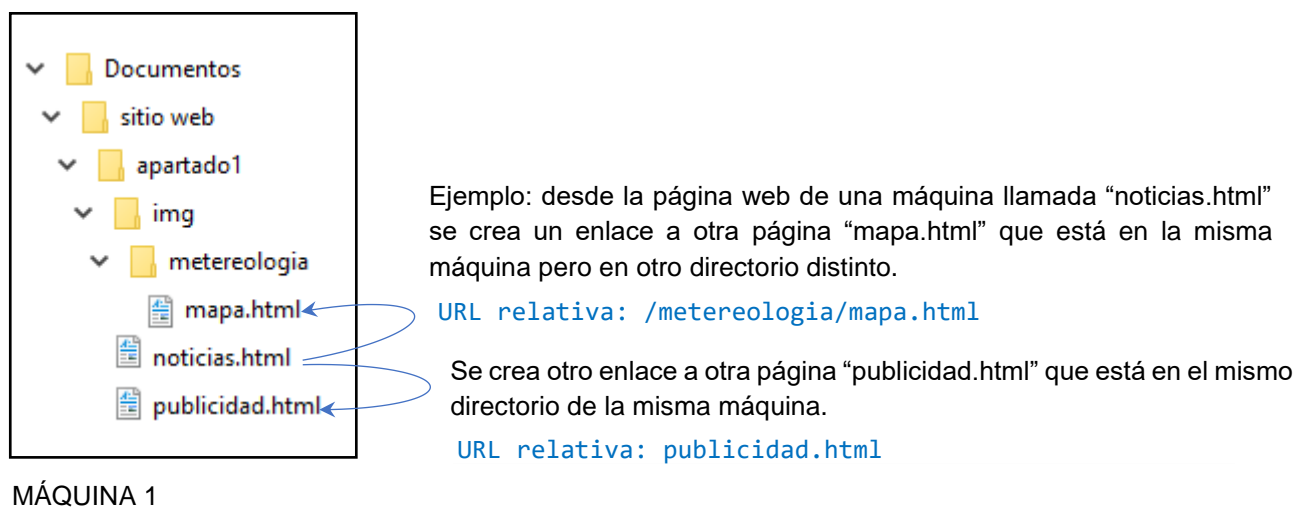
- ➔ **Enlace externo:** apunta a páginas que están en otros sitios web (están en otras máquinas a las que debe acceder a través de una red de comunicación).
- ➔ **Enlace interno:** apunta a páginas que están en el mismo sitio web (en la misma máquina). Estos enlaces funcionarán aunque no exista una red de comunicación.

## ■ Rutas relativas y absolutas

- ➔ **Ruta absoluta:** incluye todas las partes de la URL (protocolo, servidor, ruta y recurso) por lo que no se necesita más información para obtener el recurso al que apunta.



- ➔ **Ruta relativa:** no incluye todas las partes de la URL. Se parte de la ubicación del recurso actual para acceder al recurso al que se apunta.



**Ejemplos de rutas relativas y absolutas:**

A continuación se exponen algunos ejemplos de rutas absolutas y relativas dentro de una misma máquina.

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el mismo directorio
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html</code>
URL relativa	<code>pagina2.html</code>

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el directorio superior llamado <code>ruta2</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/pagina2.html</code>
URL relativa	<code>../pagina2.html</code>

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en el directorio superior llamado <code>ruta1</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/pagina2.html</code>
URL relativa	<code>../../pagina2.html</code>

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio inferior llamado <code>ruta4</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html</code>
URL relativa	<code>ruta4/pagina2.html</code>

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio inferior llamado <code>ruta6</code> que está dentro del directorio <code>ruta5</code> y que a su vez está dentro del directorio <code>ruta4</code>
URL absoluta	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/ruta5/ruta6/pagina2.html</code>
URL relativa	<code>ruta4/ruta5/ruta6/pagina2.html</code>

Elemento	Valor
Página origen	<code>http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</code>
Página enlazada	Página web llamada <code>pagina2.html</code> y que se encuentra en un directorio llamado <code>ruta4</code> que se encuentra en la raíz del servidor
URL absoluta	<code>http://www.ejemplo.com/ruta4/pagina2.html</code>
URL relativa	<code>../../../../ruta4/pagina2.html</code>

**Nunca se debe indicar en un enlace la ruta absoluta local**, es decir, la ruta absoluta del ordenador o dispositivo en el que se encuentra.

En páginas web, el directorio raíz para las rutas absolutas no es el directorio raíz del sistema de ficheros, sino el directorio raíz del servidor web.


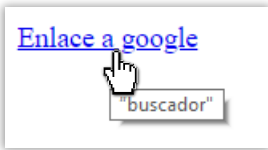
`C:/usuarios/Documentos/index.html`

- Problemas de seguridad
- No es accesible desde otros dispositivos

## 8.2 Enlaces

### ▪ `<a>`

“Anchor” o ancla. Define un enlace a un recurso (imagen, página web...). Atributos básicos:

ATRIBUTO	SIGNIFICADO	EJEMPLO
href	URL del recurso que se quiere alcanzar	<pre>&lt;a href="http://www.google.es"&gt;Enlace a google&lt;/a&gt;</pre> 
title	Descripción que aparece al pasar el ratón por encima.	<pre>&lt;a href="http://www.google.es" title="buscador"&gt;Enlace a google&lt;/a&gt;</pre> 
target	Especifica cómo se va a abrir el enlace. En HTML se emplea sólo con la opción “_blank” para crear una nueva pestaña.	<pre>&lt;a href="http://www.google.es" title="buscador" target="_blank"&gt;Enlace a google&lt;/a&gt;</pre>
download	Atributo booleano, indica que el recurso que se va a abrir debe descargarse	<pre>&lt;a href="img/foto.jpg" download&gt;Descarga&lt;/a&gt;</pre>

Incluso se pueden indicar protocolos como FTP:

```
href="ftp://ftp.srv.com/ruta/archivo.zip"
```

### ▪ Enlace a un punto del mismo documento:

Si un elemento cualquiera del documento se identifica con un valor unívoco a través del atributo “id”, es posible referenciarlo desde otro punto del documento de la siguiente manera:

```
<p id="inicio">Texto</p>
...
<a href="#inicio">Ir a la sección 1</a>
```

Esto es muy útil para crear referencias dentro de páginas largas, o para acceder a partes concretas de una página.

### ▪ Enlace a correo electrónico

También es posible vincular con un enlace el correo electrónico. Al hacerlo se consigue que, al pulsar sobre el enlace, se abra la aplicación de correo electrónico que esté instalada por defecto en el ordenador, y aparezca en ella una ventana para crear un nuevo mensaje de correo. La ventaja es que aparecerán ya cargados los datos que nos interesen.

Por ejemplo:

```
href="mailto:nombre@direccion.com"
```

Al pulsar sobre el enlace que tenga ese recurso asociado, se abrirá la aplicación de correo electrónico, y aparecerá la ventana correspondiente a crear un nuevo correo, con la dirección de destino nombre@direccion.com.

Es posible además indicar más parámetros del correo electrónico con el siguiente formato:

```
href="mailto:....?campo1=valor&campo2=valor... "
```


Si se quieren introducir espacios en el valor de algún campo, hay que utilizar el carácter codificado %20. Los saltos de línea se codifican con %0D%0A (Ver [Restricciones en las URLs](#))

Ejemplo de enlace a correo electrónico:

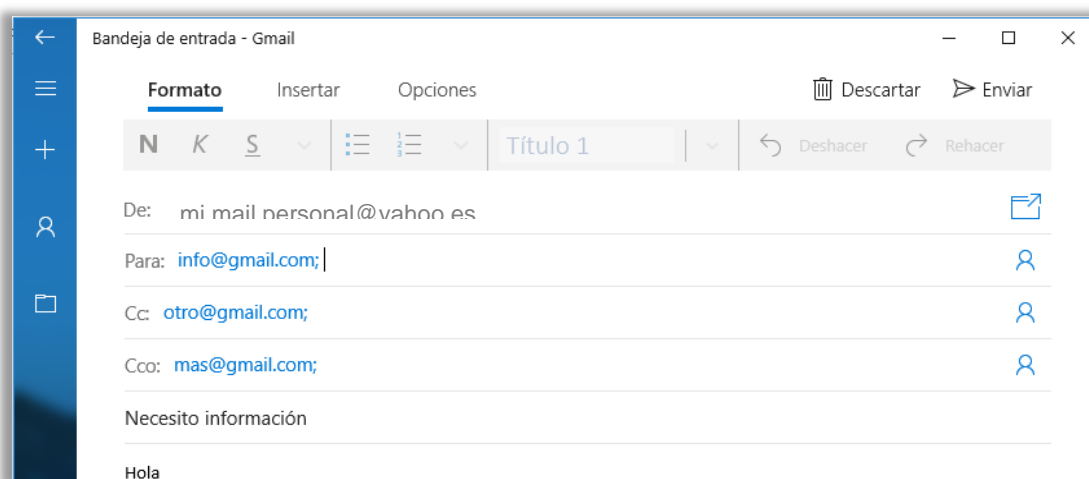
El siguiente enlace:

```
<a  
href="mailto:info@gmail.com?cc=otro@gmail.com&bcc=mas@gmail.com&subject=Necesito%  
20información&body=Hola">Envia mail</a>
```

Aparece así en el navegador:



Al pulsar sobre el enlace se abre la ventana del correo electrónico instalado en el ordenador, con los campos ya precargados:



## Actividad 14: Enlaces

- 14.1. Añadir enlaces básicos
- 14.2. Crear pestañas nuevas
- 14.3. Referencias dentro de una página
- 14.4. Enlazar distintos tipos de recursos con un enlace
- 14.5. Enlace a correo electrónico
- 14.6. Uso de URLs en enlaces



Ejercicios: enlaces. Ejercicios 4 y 5.

## 9. Tablas

Las tablas son recursos muy utilizados en HTML. **Debe evitarse su uso como elemento estructurador del contenido de la página.** Las tablas son recursos para mostrar datos de forma estructurada.

Etiquetas básicas de tabla:

- **<table>**

Etiqueta de tabla

- **<tr>**

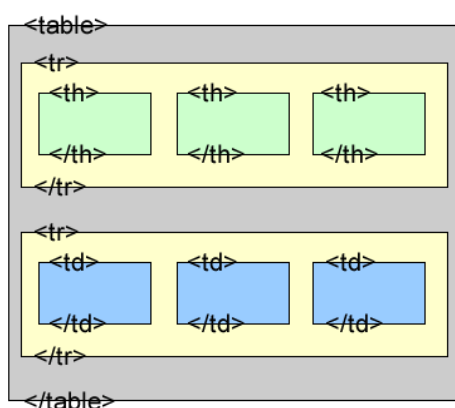
Table Row. Etiqueta de fila dentro de una tabla

- **<td>**

Table Data Cell. Etiqueta de campo dentro de una fila de una tabla

- **<th>**

Table Header. Etiqueta de campo cabecera dentro de una fila de una tabla. Por defecto aparecen los datos resaltados en negrita.



En las cabeceras <th> se puede incluir el atributo “[scope](#)” que indica de qué es cabecera (de una fila, de una columna, de un conjunto de filas o de un conjunto de columnas).

Las posibilidades son: row, col, rowgroup y colgroup.

El formato de las tablas, como en el resto de elementos HTML, se realiza mediante CSS. Sin embargo, en este guion no vamos a emplear CSS (se verá más adelante). En su lugar, vamos a utilizar un atributo de tabla a modo didáctico, **sólo hasta incorporar el código CSS.**

FORMATO DE TABLA SIN CSS	EJEMPLO
Border. Define el borde de la tabla en píxeles.	<code>&lt;table border="1"&gt;</code>

Ejemplos de tablas sencillas:

DEFINICIÓN DE TABLA	RESULTADO
<pre> &lt;table&gt;   &lt;tr&gt;     &lt;td&gt;uno&lt;/td&gt;     &lt;td&gt;dos&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;tres&lt;/td&gt;     &lt;td&gt;cuatro&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;cinco&lt;/td&gt;     &lt;td&gt;seis&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt; </pre>	<pre>   uno  dos tres  cuatro cinco seis </pre>
<pre> &lt;table&gt;   &lt;tr&gt;     &lt;th scope="col"&gt;cabecera1&lt;/th&gt;     &lt;th scope="col"&gt;cabecera2&lt;/th&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;uno&lt;/td&gt;     &lt;td&gt;dos&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;tres&lt;/td&gt;     &lt;td&gt;cuatro&lt;/td&gt;   &lt;/tr&gt;   &lt;tr&gt;     &lt;td&gt;cinco&lt;/td&gt;     &lt;td&gt;seis&lt;/td&gt;   &lt;/tr&gt; &lt;/table&gt; </pre>	<pre> <b>cabecera1 cabecera2</b> uno      dos tres     cuatro cinco    seis </pre>

## Actividad 15: Crear una tabla básica

### 15.1 Crear una tabla básica

### 15.2. Añadir una cabecera

Otras etiquetas de tabla:

### ▪ **<caption>**

Crea una descripción de la tabla

```
<table border="1">
  <caption>Tabla de números</caption>
  <tr >
    <th scope="col">cabecera1</th>
    <th scope="col">cabecera2</th>
  </tr>
  <tr >
    <td >uno</td>
    <td>dos</td>
  </tr>
  <tr>
    <td>tres</td>
    <td>cuatro</td>
  </tr>
  <tr>
    <td>cinco</td>
    <td>seis</td>
  </tr>
</table>
```

Tabla de números

<b>cabecera1</b>	<b>cabecera2</b>
uno	dos
tres	cuatro
cinco	seis

### ▪ **Atributo colspan**

Es un atributo de tabla que “expande” columnas. Se aplica a <td>.

Ejemplo:

```
<table border="1" >
  <tr >
    <th scope="col">cabecera1</th>
    <th scope="col">cabecera2</th>
    <th scope="col">cabecera3</th>
  </tr>
  <tr >
    <td colspan="2">uno</td>
    <td>tres</td>
  </tr>
  <tr>
    <td>cuatro</td>
    <td>cinco</td>
    <td>seis</td>
  </tr>
  <tr>
    <td>siete</td>
    <td>ocho</td>
    <td>nueve</td>
  </tr>
</table>
```

<b>cabecera1</b>	<b>cabecera2</b>	<b>cabecera3</b>
uno		tres
cuatro	cinco	seis
siete	ocho	nueve

```
<th colspan="2" scope="colgroup">Cabecera1</th>
```

Cabecera1	
Dato1	Dato2
Dato3	Dato4

En este caso <th> sería cabecera de un conjunto de columnas

### ▪ Atributo rowspan

Es un atributo de tabla que “expande” filas. Se aplica a <td>.

Ejemplo:

```
<table border="1" >
  <tr >
    <th scope="col">cabecera1</th>
    <th scope="col">cabecera2</th>
    <th scope="col">cabecera3</th>
  </tr>
  <tr >
    <td>uno</td>
    <td>dos</td>
    <td>tres</td>
  </tr>
  <tr >
    <td rowspan="2">cuatro</td>
    <td>cinco</td>
    <td>seis</td>
  </tr>
  <tr>
    <td>ocho</td>
    <td>nueve</td>
  </tr>
</table>
```

<b>cabecera1</b>	<b>cabecera2</b>	<b>cabecera3</b>
uno	dos	tres
cuatro	cinco	seis
	ocho	nueve

```
<th rowspan="2" scope="rowgroup">Cabecera1 </th>
```

<b>Cabecera1</b>	Dato1	Dato2
	Dato3	Dato4

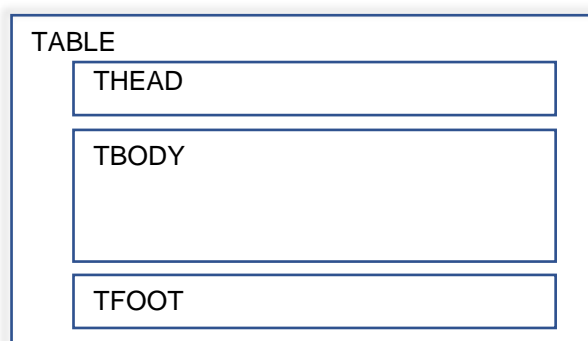
En este caso <th> sería cabecera de un conjunto de filas

### ▪ <thead><tbody><tfoot>

Estas etiquetas semánticas estructuran el contenido de la tabla. Normalmente <thead> se emplea para la cabecera de la tabla, es decir, los títulos, y <tfoot> para el pie de tabla (totales o resúmenes).

Tienen dos funciones:

- ▶ Poder aplicar distinto formato a cada una. Por defecto no implican ningún formato en concreto, pero puede aplicarse un estilo CSS específico.
- ▶ A la hora de imprimir, la cabecera y el pie se repiten en cada página, si la tabla ocupa varias páginas.



Ejemplo:

```

<table border="1">
  <caption>Lista de la compra</caption>
  <thead>
    <tr>
      <th scope="col">Producto</th>
      <th scope="col">Cantidad</th>
      <th scope="col">Precio</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Acelgas</td>
      <td>1</td>
      <td>1,99 &euro;</td>
    </tr>
    <tr>
      <td>Tomates</td>
      <td>2</td>
      <td>5,20 &euro;</td>
    </tr>
    <tr>
      <td>Espárragos</td>
      <td>1</td>
      <td>2,99 &euro;</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="2">Total</td>
      <td>15,38 &euro;</td>
    </tr>
  </tfoot>
</table>

```

**THEAD**

**TBODY**

**TFOOT**

Lista de la compra

Producto	Cantidad	Precio
Acelgas	1	1,99 €
Tomates	2	5,20 €
Espárragos	1	2,99 €
Total		15,38 €

## Actividad 16: Crear tablas

- 16.1 Añadir una descripción de tabla
- 16.2. Añadir filas expandidas
- 16.3 Filas expandidas completamente
- 16.4 Dar formato a la tabla

Ejercicios: tablas. Ejercicios 6, 7 y 8.

Práctica tablas

Práctica horario

## 10. Imágenes

### 10.1 Tipos de imágenes en un documento web

En un documento HTML pueden insertarse imágenes a partir de un fichero con formato de imagen. Se distingue dos tipos de imágenes:

➔ **Imágenes de contenido.** Proporcionan información y complementan la información textual. No se puede prescindir de ellas ya que forman parte del contenido junto con el texto.

**Se incluyen directamente en el código HTML mediante la etiqueta <img>.**

*Por ejemplo, en la siguiente página web de la wikipedia se incluyen imágenes de contenido que representan los distintos huesos del cráneo.*

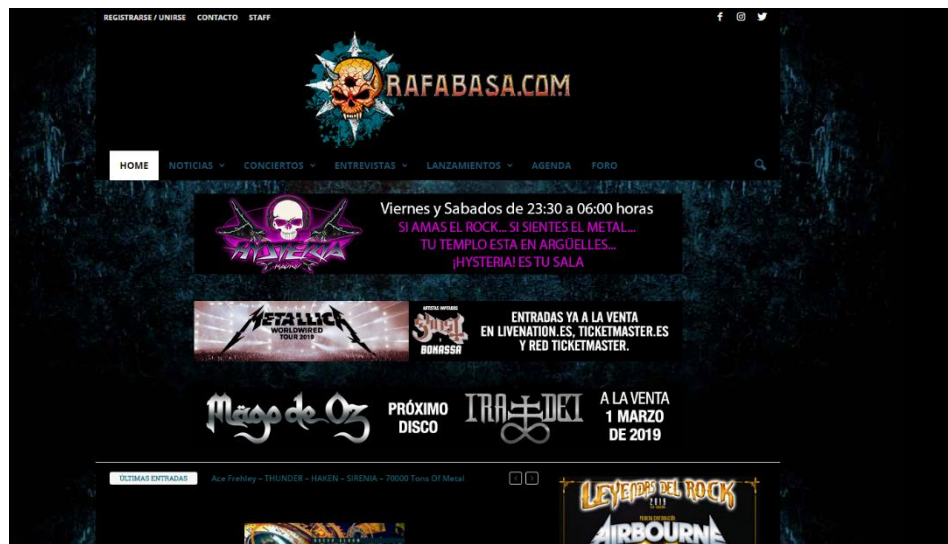


<https://es.wikipedia.org/wiki/Cr%C3%A1neo>

➔ **Imágenes de formato o adorno.** Se utilizan para mejorar el aspecto de la página web. Por ejemplo mostrando pequeños iconos en las listas de elementos, fondos de página, etc. Se podría prescindir de ellas sin perder información.

**No se deben incluir en el código HTML, sino que se deben emplear hojas de estilos CSS para mostrarlas.**

*En la siguiente página web también aparecen imágenes de cráneos, pero en este caso son decorativas.*



## 10.2 Etiquetas para imágenes

### ▪ `<img>`


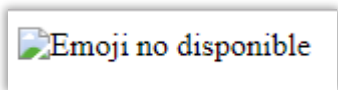
Etiqueta para insertar una imagen de contenido en una página web.

El formato básico es el siguiente (**no tiene etiqueta de cierre**):

```

```

Atributos básicos de `<img>`:

ATRIBUTO	SIGNIFICADO	EJEMPLO
src	Fichero con la imagen (ruta y nombre de fichero)	<pre>&lt;img src="una_imagen.jpg"&gt;</pre> <pre>&lt;img src="img/una_imagen.jpg"&gt;</pre> <pre>&lt;img src="../una_imagen.gif"&gt;</pre>
title	<p>Descripción que aparece al pasar el ratón por encima.</p> <p>Si la imagen no está disponible, y no hay atributo alt definido, se mostrará el título en lugar de la imagen.</p>	<pre>&lt;img src="gafas1.jpg" title="Emoji"&gt;</pre> 
alt	<p>Inserta un texto "alternativo", que se visualizará en lugar de la imagen, en caso de que ésta no pueda ser insertada.</p> <p>Las razones por las que no se puede insertar una imagen pueden ser que no se encuentra el archivo, que el formato de imagen es desconocido o que no está soportado.</p>	<pre>&lt;img src="gafas1.jpg" alt="Emoji no disponible"&gt;</pre> <p>Si la imagen no existe, o no está en el directorio adecuado, el navegador presentará:</p> 
width	<p>Ancho.</p> <p><u>No está recomendado</u> definir el ancho como un atributo de <code>&lt;img&gt;</code>.</p>	<pre>&lt;img src="gafas1.jpg" width="200"&gt;</pre> <pre>&lt;img src="gafas1.jpg" width="35%"&gt;</pre>
height	<p>Alto.</p> <p><u>No está recomendado</u> definir el ancho como un atributo de <code>&lt;img&gt;</code>.</p>	<pre>&lt;img src="gafas1.jpg" width="200"&gt;</pre> <p>No se emplea con porcentajes.</p>

### ▪ Tamaño de la imagen

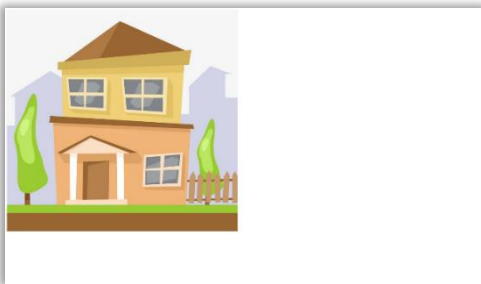
Como ya se ha comentado, los atributos `width` y `height` se emplean en `<img>` para indicar el tamaño en el que se desea representar la imagen. Hay que recordar que no está recomendado definir el tamaño de esta forma, sino de debería definirse con un estilo CSS. Sin embargo lo introducimos aquí de forma didáctica.

Observa el comportamiento de los atributos de tamaño en los siguientes ejemplos:

```

```

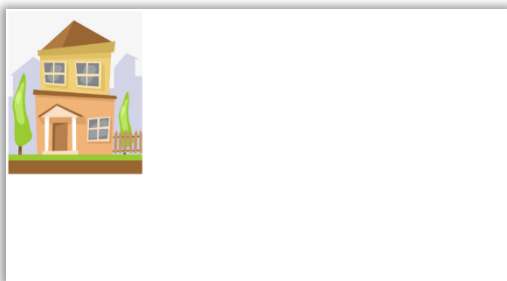
La imagen aparece en su tamaño original



```

```

Se limita el alto de la imagen a 300 píxeles. No cambia la proporción de la imagen, es decir, el navegador ajusta el ancho para que la imagen siga viéndose igual.



```

```

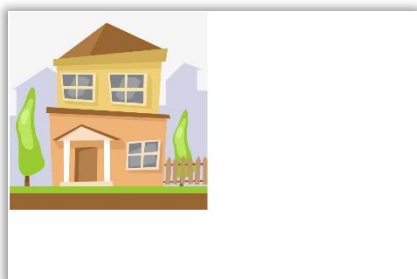
La imagen se amplía hasta ocupar el 100% del ancho, es decir, todo el ancho disponible de la página. Si cambiamos el tamaño de la ventana, la imagen se ajustará para ocupar siempre el 100% del ancho



```

```

Al poner el atributo height con un porcentaje, el navegador lo ignora y añadirá la imagen en su tamaño original



```

```

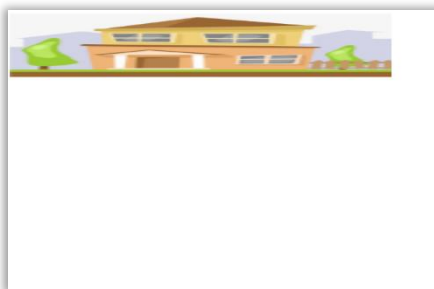
Se limita el ancho de la imagen a 200 píxeles. No cambia la proporción de la imagen, es decir, el navegador ajusta el alto para que la imagen siga viéndose igual.



```

```

Se limita el alto y el ancho de la imagen. En este caso el navegador no respeta la proporción original de la imagen



```
< img src="casa.jpg" width="10%">
```

La imagen se amplía hasta ocupar el 100% del ancho, es decir, todo el ancho disponible de la página. Si cambiamos el tamaño de la ventana, la imagen se ajustará para ocupar siempre el 100% del ancho





- **<figure> <figcaption>**

<figure> marca una imagen en un documento. <figcaption> asocia una descripción a esa imagen.

```
<figure>
  
  <figcaption>Fig.1.Construcción blanda con judías hervidas, Dalí, 1929</figcaption>
</figure>
```

## Actividad 17: Insertar imágenes

17.1. Insertar la primera imagen

17.2. Imagen inaccesible

17.3. Ajustar el tamaño de la imagen

- **<picture> <source>**

<picture> aumenta la flexibilidad para indicar el archivo origen al utilizar <img>.

Cuando se utiliza solamente <img>, se especifica el fichero con la imagen, que puede ser dimensionado con CSS o con los atributos width y height. Sin embargo, <picture> permite utilizar distintos ficheros en función de la pantalla de visualización.

<source> especifica ficheros alternativos que el navegador puede utilizar. En el atributo **media** se indica una propiedad del dispositivo de visualización y en el atributo **srcset** se establece el fichero.

```
<picture>
  <source media="all and (min-width:300px)" srcset="imagen_1.jpg">
  <source media="tv and (width:650px)" srcset="imagen_2.jpg">
  
</picture>
```

El atributo “media” hace referencia a [media query](#), una técnica, empleada sobre todo en css, que permite modificar el formato de presentación en función de las propiedades de la pantalla del dispositivo. Puede indicarse tanto el tipo de dispositivo (screen, tv, braille, projection, handheld...) como sus propiedades (tamaño, orientación, resolución, color...)

## Actividad 18: Imagen con media query

18.1. Insertar <picture>

18.2 Tamaño de la pantalla de visualización

## 11. Vídeo y audio

En HTML5 es posible insertar directamente video y audio en la página web de una forma similar a cómo se insertan imágenes.

### ▪ <video>

Etiqueta para insertar un video en una página web.

El formato básico es el siguiente. Observa que **sí tiene etiqueta de cierre**:

```
<video src="un_video.mp4"></video>
```

Se puede añadir un texto entre las etiquetas <video> y </video> que sólo será visible si el navegador no es compatible con HTML5:

```
<video src="un_video.mp4">Navegador no compatible</video>
```

Ese texto sólo se verá en caso de que el navegador no reconozca la etiqueta <video>

#### ACLARACIÓN

Observa que el atributo alt en las imágenes es un texto alternativo que aparece cuando **la imagen no puede añadirse** (formato desconocido, ruta incorrecta o imagen inexistente). Pero el navegador sí es compatible con la etiqueta <img>

En cambio, el texto que se pone entre las etiquetas <video> y </video> aparecerá sólo si **el navegador no es compatible** con la etiqueta <video>. No depende de si el video existe o no, ni de su formato.

Atributos básicos de <video>:

ATRIBUTO	SIGNIFICADO	EJEMPLO
src	Fichero con el video (ruta y nombre de fichero)	<pre>&lt;video src="un_video.mp4"&gt;&lt;/video&gt; &lt;video src="videos/un_video.mp4"&gt;&lt;/video&gt; &lt;video src="../un_video.webM"&gt;&lt;/video&gt;</pre>
controls	Indica si deben aparecer los controles del video para que el usuario pueda utilizar las opciones de pausa, adelantar, retrasar... Es un atributo booleano.	<pre>&lt;video src="un_video.mp4"controls&gt; &lt;/video&gt;</pre>
poster	Asocia una imagen de forma que aparezca como portada del video	<pre>&lt;video src="un_video.mp4" poster="una_imagen.jpg"&gt;&lt;/video&gt;</pre>
autoplay	Indica que el video debe empezar a reproducirse automáticamente. Es un atributo booleano. (En Chrome no funciona correctamente).	<pre>&lt;video src="un_video.mp4" autoplay controls&gt;&lt;/video&gt;</pre>

loop	Indica que el video debe reproducirse continuamente, es decir, cuando finalice una reproducción, comenzará de nuevo desde el principio. Es un atributo booleano.	<code>&lt;video src="un_video.mp4" loop controls&gt; &lt;/video&gt;</code>
muted	Indica que el video debe reproducirse por defecto en silencio. Es un atributo booleano	<code>&lt;video src="un_video.mp4" muted controls&gt; &lt;/video&gt;</code>
width	Ancho <u>No está recomendado</u> definir el ancho como un atributo de <video>.	<code>&lt;video src="un_video.mp4" width ="200" controls&gt;&lt;/video&gt;</code>
height	Alto <u>No está recomendado</u> definir el alto como un atributo de <video>.	<code>&lt;video src="un_video.mp4" height ="200" controls&gt;&lt;/video&gt;</code>

#### ▪ **<source>**

Establece formatos alternativos, por si el navegador no es compatible con algún formato de video concreto.

```
<video controls>
<source src="un_video.mp4" type="video/mp4">
<source src="un_video.webM" type="video/webM">
</video>
```

#### ▪ **<audio>**

Etiqueta para insertar un audio en una página web. El formato básico es el siguiente. Observa que **sí tiene etiqueta de cierre**:

```
<audio src="un_audio.mp3"></audio>
```

Se puede añadir un texto entre las etiquetas <audio> y </audio> que sólo será visible si el navegador no es compatible con HTML5:

```
<audio src="un_audio.mp4">Navegador no compatible</audio>
```

Ese texto sólo se verá en caso de que el navegador no reconozca la etiqueta <audio>

En general, el comportamiento de la etiqueta <audio> es como la etiqueta <video>

Como es lógico, algunos atributos de <video> no pueden aplicarse a <audio>: poster, width, height.

## Actividad 19: Insertar video y audio

### 19.1. Insertar el primer video

### 19.2. Añadir una portada al video

### 19.3. Insertar el primer audio

## 12. Mapa de imagen

Un mapa de imagen permite definir diferentes zonas dentro de una imagen. El usuario puede pulsar con el ratón sobre cada una de las zonas definidas y cada una de ellas puede apuntar a una URL diferente.

Actualmente su uso no está muy extendido, se emplea sólo para casos muy específicos

### ▪ `<img>` `<map>` `<area>`

Las zonas o regiones que se pueden definir en una imagen se crean mediante rectángulos, círculos y polígonos. Para crear un mapa de imagen, en primer lugar se inserta la imagen original mediante la etiqueta `<img>`. A continuación, se utiliza la etiqueta `<map>` para definir las zonas o regiones de la imagen. Cada zona se define mediante la etiqueta `<area>`.

`<img>` aparte del atributo `src` donde se indica el fichero de la imagen, debe añadirse el atributo `usemap` que lo asocie con el mapa.

`<map>` debe crearse asignándole un nombre. Tiene etiqueta de apertura y cierre.

```

<map name="nombre_mapa">
</map>
```

`<area>` Se debe crear un área para cada zona de la imagen que se quiera diferenciar. No tiene etiqueta de cierre.

Tiene una serie de atributos:

➔ **href**: indica el enlace al que se irá en el caso de pulsar sobre el área

➔ **shape**: indica la forma del área. Posibles valores:

shape="rect"	área rectangular
shape="circle"	área circular
shape="poly"	área poligonal

➔ **coords**: indica las coordenadas del área. Dependiendo del tipo indicado en `shape`, habrá que poner más o menos coordenadas.

### ▪ Coordenadas

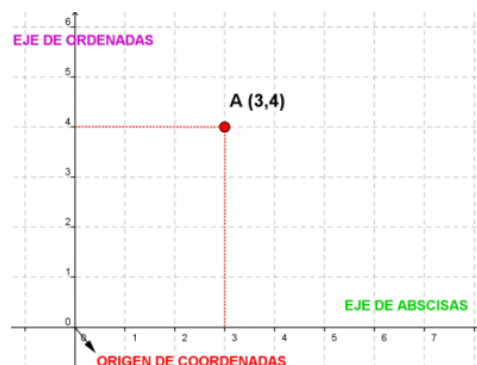
Cada punto tiene asociadas dos coordenadas: (X, Y).

X: eje horizontal

Y: eje vertical

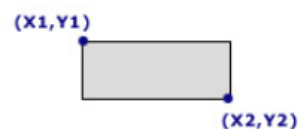
El punto A tiene las coordenadas (3,4), es decir, x=3, y=4

Estas coordenadas son absolutas (píxeles) por lo que si se redimensiona la imagen con `width` o `height` puede haber problemas con los mapas.



**Área rectangular (shape="rect").** Es necesario conocer dos puntos para delimitar un rectángulo. Hay que indicar 4 coordenadas, dos por cada punto.

```
<area shape="RECT" coords="X1,Y1,X2,Y2" href="#">
```



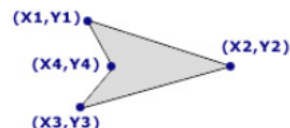
**Área circular (shape="circle").** Es necesario conocer un punto y el radio. Hay que indicar 3 coordenadas, dos por el punto y una por el radio.

```
<area shape="CIRCLE" coords="X1,Y1,R" href="#">
```



**Área poligonal (shape="poly").** Es un área compleja en el que se crea una forma uniendo sus puntos. Puede tener un número variable de coordenadas. Serviría para formar triángulos, hexágonos,... o cualquier otra forma regular o irregular. Hay que dar dos coordenadas por cada punto de la imagen.

```
<area shape="POLY" coords="X1,Y1,X2,Y2,X3,Y3,X4,Y4" href="#">
```



Las coordenadas de un mapa pueden hacerse "a mano", aunque si el mapa es complejo se puede recurrir a alguna ayuda que genere las coordenadas de forma automática. Algunos generadores:

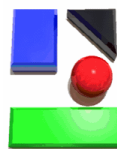
<https://www.image-map.net/>

<http://imagemap-generator.dariodomi.de/>

<https://www.image-maps.com/>

### ▪ Ejemplo. Creación de mapa de imágenes

Se tiene la siguiente imagen, cuyo nombre es "formas.gif":



Queremos que al pulsar sobre cada una de las formas geométricas, nos vayamos a una página html distinta, a través de un enlace.

**Paso 1:** insertar la imagen ``

**Paso 2:** insertar el mapa, dándole el nombre que se quiera, por ejemplo "objetos". Observa que `<map>` tiene etiqueta de cierre `</map>`

```

<map name="objetos">
</map>
```

**Paso 3:** asociar el mapa con la imagen a través del atributo **usemap**

```

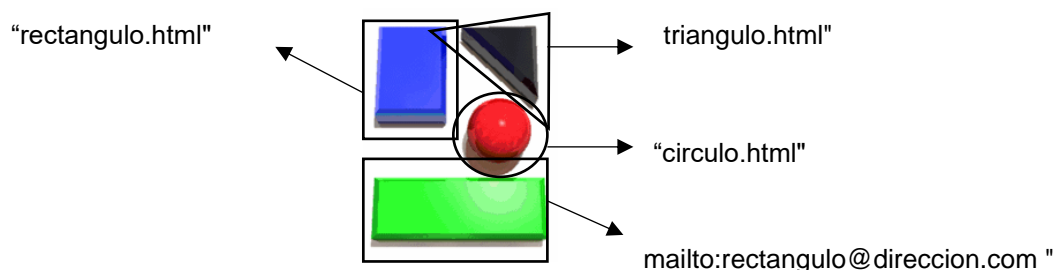
<map name="objetos">
</map>
```

**Paso 4:** insertar tantos `<area>` como zonas distintas queramos delimitar en el mapa de imagen:

```

<map name="objetos">
  <area shape="rect" coords="20,25,84,113" href="rectangulo.html">
  <area shape="polygon" coords="90,25,162,26,163,96,89,25,90,24"
    href="triangulo.html">
  <area shape="circle" coords="130,114,29" href="circulo.html">
  <area shape="rect" coords="19,156,170,211"
    href="mailto:rectangulo@direccion.com">
  <area shape="default" nohref="nohref">
</map>
```

**Resultado:** Al pasar pulsar con el ratón por cada una de las zonas delimitadas, nos conduce a la página correspondiente. Por ejemplo, al pulsar sobre el círculo el enlace nos lleva a `circulo.html`.



## Actividad 20: Mapa de imágenes

### 20.1 Crear mapa de imágenes

## 13. Insertar objetos

La forma de insertar objetos en una página web ha cambiado en las distintas versiones de HTML.

- ▶ Al principio se usaron **frameset** que dividían la página en frames (una especie de subventanas). Daban rapidez a la carga de la página pero causaban problemas de dimensionamiento. Actualmente el concepto de frame está obsoleto.
- ▶ Después se emplearon **Applets de Java y Flash**, pero tenían problemas de seguridad entre otros.
- ▶ Hoy se usa **iframe**, aunque pueden generar problemas de formato y seguridad de la página.

### ▪ <iframe>

Un iframe es una subventana en la que pueden insertarse distintos objetos (videos, documentos, otras páginas web...)

Presenta problemas de seguridad, ya que esa subventana puede ejecutar código malicioso no controlado. Para controlarlo se emplea el atributo `sandbox`, quien establece los permisos del iframe. Es recomendable usar iframes solo con HTTPS.

El formato básico es el siguiente. Observa que **sí tiene etiqueta de cierre**:

```
<iframe src="pag.html"></iframe>
```

Se puede añadir un texto entre las etiquetas `<iframe>` y `</iframe>` que sólo será visible si el navegador no es compatible con iframes:

```
<iframe src="pag.html">Navegador no compatible</iframe>
```

Ese texto sólo se verá en caso de que el navegador no reconozca la etiqueta `<iframe>`

Atributos básicos de `<iframe>`:

ATRIBUTO	SIGNIFICADO	EJEMPLO
src	Fichero con el objeto (ruta y nombre de fichero)	<pre>&lt;iframe   src="https://es.wikipedia.org"&gt; &lt;/iframe&gt; &lt;iframe   src="img/cisne.jpg"&gt;&lt;/iframe&gt; &lt;iframe   src="libro_cm.pdf"&gt;&lt;/iframe&gt; &lt;iframe src="video/lago-   cisnes.mp4"&gt; &lt;/iframe&gt;</pre>
sandbox	<p>Proporciona seguridad al iframe, evitando que se ejecute algún software en él que quede fuera del control del navegador.</p> <p>En su opción por defecto "sandbox se dan los permisos mínimos para el iframe.</p>	<pre>&lt;iframe   src=https://es.wikipedia.org   sandbox&gt;&lt;/iframe&gt;  &lt;iframe   src=https://es.wikipedia.org   sandbox="allow-scripts"&gt; &lt;/iframe&gt;</pre>

	Otros parámetros serían: "allow-scripts", "allow-forms", "allow-popups"... (por ejemplo sandbox="allow-scripts")	
width	Ancho <u>No está recomendado</u> definir el ancho como un atributo de <object>.	<code>&lt;iframe src="https://es.wikipedia.org" width="200"&gt; &lt;/iframe&gt;</code>
height	Alto <u>No está recomendado</u> definir el alto como un atributo de <object>.	<code>&lt;iframe src="https://es.wikipedia.org" height="200"&gt; &lt;/iframe&gt;</code>

### ▪ Copiar código de inserción

Hay ciertas aplicaciones o páginas que facilitan para su contenido el crear iframes con objetos. Es decir, facilitan el que podamos incorporar un iframe en nuestro sitio web, que sea un objeto de su contenido.

Un ejemplo será **youtube**.

Si durante la visualización de un video desde youtube pulsamos con el botón derecho y seleccionamos la opción "copiar código de inserción", directamente en el portapapeles se copiará la etiqueta <iframe> con los campos necesarios para incluirla en nuestra web.

Un iframe extraído de youtube:

```
<iframe width="751" height="563" src="https://www.youtube.com/embed/CYFUiYoYPbg"
frameborder="0" allow="autoplay; encrypted-media" allowfullscreen></iframe>
```

## Actividad 21: Insertar objetos con iframes

21.1 Insertar distintos objetos en iframes

21.2 Insertar un video de youtube con el código de inserción proporcionado

Ejercicios: multimedia. Ejercicios 9, 10 y 11.



## 14. Formularios

### 14.1 Introducción

Los formularios son elementos, que como los enlaces, permiten una **interacción del usuario** con la página web.

Su tarea principal es **recoger información**. El usuario, por tanto, debe introducir esa información en los campos del formulario.

Una vez se ha recogido esa información el formulario **la enviará al servidor** o donde decida el programador para ser tratada, mostrada y/o almacenada.



Formulario

Tipo documento  
Cédula

Nombres:

Apellidos:

Genero:  
☒ Masculino  
☐ Femenino

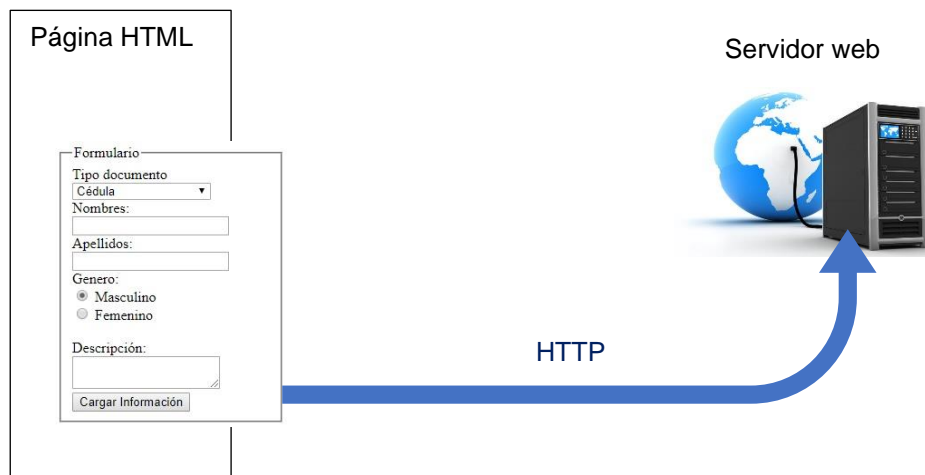
Descripción:

Cargar Información

### 14.2 Datos del formulario

El estándar HTML permite crear formularios para que los usuarios interactúen con las aplicaciones web.

Las etiquetas de formulario permiten que el usuario introduzca datos en la página web, que serán enviados al servidor para su procesamiento. **Estas etiquetas no incluyen el tratamiento de los datos, sólo se utilizan para recogerlos y transferirlos** mediante el protocolo HTTP. Será el servidor el encargado de analizar y utilizar esos datos introducidos por el usuario.



En la propia página web sí puede hacerse una primera **validación** de los datos introducidos, en dos niveles:

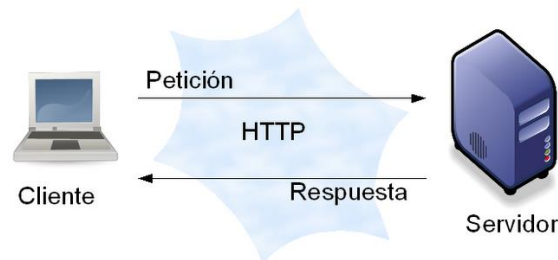
- ▶ En un primer nivel se realiza una validación sencilla a través de las propias **etiquetas HTML** (tipo de campo y expresiones regulares)
- ▶ En un segundo nivel se puede emplear código **JavaScript** para la validación de la información.

Cada dato de un formulario se recoge a través de un campo de formulario concreto, y se envía mediante un par: **nombre valor**.

### 14.3 Protocolos de transferencia de los datos

#### HTTP Hypertext Transfer Protocol

Es el protocolo de comunicación que permite las transferencias de información en la World Wide Web. Definido por W3C y IETF



Está basado en el modelo cliente-servidor. Se define por defecto el puerto 80 del servidor para HTTP.

Es un protocolo sin estado, es decir no guarda información de conexiones anteriores. Por ello, algunas aplicaciones web emplean algunas técnicas destinadas a agrupar varias conexiones (cookies, sesiones)

Se basa en la transmisión de texto plano, dividido en:

- Cabecera
- Cuerpo (opcional)

#### Ejemplo de comunicación HTTP:

PETICIÓN HTTP  
(enviada por el  
cliente)

```
GET /index.html HTTP/1.1
Host: www.example.com
Referer: www.google.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0)
Gecko/20100101 Firefox/45.0
Connection: keep-alive
[Línea en blanco]
```

RESPUESTA HTTP  
(enviada por el  
servidor)

```
HTTP/1.1 200 OK
Date: Fri, 31 Dec 2020 23:59:59 GMT
Content-Type: text/html
Content-Length: 1221
<html lang="eo">
<head>
<meta charset="utf-8">
<title>Título del sitio</title>
</head>
<body>
<h1>Página principal de tuHost</h1>
(Contenido)
.
.
.
</body>
</html>
```

La respuesta incluye, dentro del cuerpo, el texto plano con el código HTML.

El navegador web al recibirlo interpreta el lenguaje de marcas y presenta al usuario el contenido de la página web.

## HTTPS Hypertext Transfer Protocol Secure

Similar a HTTP con algunas diferencias:



- Emplea por defecto el puerto 443 del servidor
- La información se transmite cifrada por la red.

### 14.4 Método de transferencia de los datos

Los datos del formulario pueden enviarse al servidor por diversos métodos. Los más utilizados son:

#### ▪ **get**

- Los datos se envían en la cabecera HTTP.
- Los datos se envían como variables URL. Se añaden pares nombre=valor a la URL
- Los datos serán visibles en la barra de direcciones del navegador.  
`Dirección_servidor?nombre1=valor1&nombre2=valor2`
- Hay que respetar la limitación de longitud de URL (aprox. 3000 caracteres)
- No permite anexar archivos.

#### ▪ **post**

- Los datos se envían en el cuerpo del mensaje HTTP.
- Los datos se envían como pares nombre=valor, pero no serán visibles en la barra de direcciones del navegador. Esto no significa que los datos se envíen de forma segura.
- No hay limitaciones de tamaño.
- Permite anexar archivos.

### 14.5 Etiquetas de formulario

#### ▪ **<form>**

Etiqueta principal en un formulario. Define el método de envío y el receptor de los datos.

```
<form>
  Etiquetas de formulario. Incluye todos los elementos botones, cuadros de texto,
  listas desplegables...
</form>
```

Atributos básicos de <form>:

ATRIBUTO	SIGNIFICADO	EJEMPLO
action	Establece dónde se enviarán los datos recopilados en el formulario. Indica la URL de la aplicación del servidor que se encarga de procesar los datos y genera la respuesta.	<code>&lt;form   action="http://www.servidor   .es/maneja_formulario.php"&gt;</code>

method	Establece el método de envío de los datos del formulario (los más habituales son get y post)	<code>&lt;form method="get"&gt;</code> <code>&lt;form method="post"&gt;</code>
enctype	<p>Establece el tipo de codificación de los datos enviados al servidor.</p> <p>Posibles valores:  <a href="https://www.w3schools.com/tags/att_form_enctype.asp">https://www.w3schools.com/tags/att_form_enctype.asp</a></p> <p>Cuando se envían ficheros el valor es:  <code>multipart/form-data</code></p>	<code>&lt;form</code> <code>enctype="multipart/form-</code> <code>data"&gt;</code>

### ▪ **<input>**

Etiqueta básica de formularios que permite integrar una gran diversidad de campos de entrada de datos. Debe estar dentro de un elemento `<form>`. No tiene etiqueta de cierre.

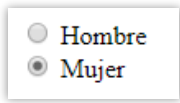
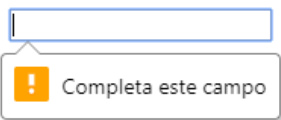


Algunos tipos:

- ▶ **text**: campo de tipo texto
- ▶ **color**: campo de tipo color
- ▶ **email**: campo de tipo email
- ▶ **number**: campo de tipo número
- ▶ **date**: campo de tipo fecha
- ▶ **tel**: campo de tipo teléfono
- ▶ **time**: campo de tipo hora
- ▶ **password**: campo de tipo contraseña
- ▶ **image**: campo de tipo imagen
- ▶ **range**: campo de tipo rango
- ▶ **button**: campo de tipo botón
- ▶ **reset**: campo de tipo botón “reset” (borra los datos)
- ▶ **submit**: campo de tipo botón “enviar”
- ▶ **checkbox**: campo de tipo checkbox (selección o no de una opción)
- ▶ **radio**: campo de tipo “radiobutton” (selección de una opción entre varias)
- ▶ **file**: campo de tipo fichero
- ▶ **hidden**: campo de fichero oculto (el usuario no lo ve ni puede actualizarlo. SE emplea para enviar datos al servidor)

Puedes consultar una lista completa de los atributos de `<input>`, y de todas las posibilidades que admite cada tipo, en el enlace:

[https://www.w3schools.com/tags/tag\\_input.asp](https://www.w3schools.com/tags/tag_input.asp)

Atributos básicos de <input>:

ATRIBUTO	SIGNIFICADO	EJEMPLO
name	<p>Especifica el nombre que se le da a ese dato. Es imprescindible para que el servidor identifique de qué dato se trata.</p> <p>El nombre puede ser cualquiera que se quiera dar.</p>	<pre>&lt;input type="text" name="apellido"&gt;</pre>
value	<p>Especifica el valor inicial que tendrá un dato. Aparecerá en pantalla como valor por defecto y el usuario podrá modificarlo. Es opcional.</p>	<pre>&lt;input type="text" name="apellido" value="Martínez"&gt;</pre>
type	<p>Establece el tipo de elemento input (text, color, email...)</p>	<pre>&lt;input type="text" name="apellido"&gt;</pre>
checked	<p>Especifica que el campo de formulario aparecerá marcado por defecto.</p> <p>Sólo es aplicable a campos que permiten ser marcados: <b>checkbox</b> o <b>radio</b>.</p> <p>Es un atributo booleano.</p>	<pre>&lt;input type="radio" name="genero" value="H"&gt; Hombre &lt;br&gt; &lt;input type="radio" name="genero" value="M" checked&gt; Mujer</pre> 
maxlength	<p>Especifica que la longitud máxima en caracteres que puede introducirse en un campo de tipo texto.</p>	<pre>&lt;input type="text" name="nombre" maxlength="10"&gt;</pre>
size	<p>Especifica que la longitud en caracteres que se visualiza en un campo de tipo texto.</p>	<pre>&lt;input type="text" name="nombre" size="10"&gt;</pre>
required	<p>Atributo de &lt;input&gt; que especifica que el campo de formulario es obligatorio. Es un atributo booleano.</p>	<pre>&lt;input type="text" name="apellido" required&gt;</pre> 
readonly	<p>Especifica que el campo de formulario es de sólo lectura, no puede modificarse. Es un atributo booleano.</p>	<pre>&lt;input type="text" name="apellido" value="Martinez" readonly&gt;</pre> 
disabled	<p>Especifica que el campo de formulario es visible, pero está deshabilitado (aparece en gris). Es un atributo booleano.</p>	<pre>&lt;input type="text" name="apellido" value="Martinez" disabled&gt;</pre> 

autofocus	Especifica que el campo de formulario aparecerá con el foco al cargar la página (el cursor estará situado en él, y está resaltado de alguna forma). Lógicamente sólo debe haber un campo con este atributo. Es un atributo booleano.	<code>&lt;input type="text" name="apellido" value="Martinez" autofocus&gt;</code>
placeholder	Muestra un texto al usuario como ayuda para que entienda el dato que debe rellenar.	<code>&lt;input type="text" name="apellido" placeholder="Martinez" &gt;</code>

### ▪ Algunos ejemplos del uso de `<input>`:

INPUT TYPE = **NUMBER**

Permite seleccionar un número indicando un mínimo, máximo, e incluso los saltos posibles entre uno y otro. Ejemplo:

Medida anillo: `<input type="number" name="medida" min="10" max="20" step="2">`

Medida anillo:

Sólo es posible seleccionar los valores: 10, 12, 14, 16, 18 y 20

INPUT TYPE = **CHECKBOX**

Permite seleccionar o no una opción. Es necesario indicar el nombre (atributo name), sin embargo el valor (atributo value) es opcional, ya que si no se indica, tomará el valor "on" por defecto, al marcar la opción.

Cada opción es un campo distinto, no ocurre como en el caso de los radio (opciones múltiples para un mismo campo), por lo que **cada opción tiene su propio nombre** (atributo name) Ejemplo:

`<input type="checkbox" name="conforme" checked>Estoy de acuerdo<br>`  
`<input type="checkbox" name="suscribir">Deseo suscribirme`

☒ Estoy de acuerdo  
☐ Deseo suscribirme

Al enviar los datos del formulario, se enviará el par `conforme=on`, ya que `name=conforme` y `value=on` (valor por defecto, al no estar indicado). No enviará el campo "suscribir" ya que no está marcado.

INPUT TYPE = **RADIO**

Permite seleccionar una única opción entre varias. Es necesario indicar el mismo nombre (atributo name) en todas las opciones, ya que se trata de un mismo dato con varias posibilidades. Cada opción tendrá un valor diferente (atributo value). Cuando se envíen los datos del formulario, se enviará el dato con el valor de la opción elegida, es decir, se enviará el par name=value. Ejemplo:

```
<input type="radio" name="genero" value="H" > Hombre <br>
<input type="radio" name="genero" value="M" > Mujer
```

☐ Hombre
 ☒ Mujer

Tienen el mismo name
 Tienen distinto value

Al enviar los datos del formulario, se enviará el par **genero=M**, ya que name="genero" y value="M"

INPUT TYPE = **RESET, SUBMIT, BUTTON**

Son botones que realizan una acción, no definen ningún dato. Si no se indica el atributo "value", el navegador mostrará un texto de botón por defecto.

- ➔ **RESET: Botón para borrar los datos del formulario.** Al pulsarlo se borran todos los datos que el usuario haya introducido previamente.
- ➔ **SUBMIT: Botón para enviar los datos del formulario.** Al pulsarlo se envían todos los datos que el usuario ha introducido previamente en el resto de campos.
- ➔ **BUTTON: Botón de uso libre.** Se combina con el uso de JavaScript, de modo que al pulsarlo se ejecute algún código definido específicamente para ello.

```
<input type="reset"> Restablecer
<input type="reset" value="borrar datos"> borrar datos
```

## Actividad 22. Formularios básicos

- 22.1 Crear un formulario con nombre y password
- 22.2. Crear un formulario básico
- 22.3. Enviar datos del formulario con métodos get y post
- 22.4 Incluir un campo oculto
- 22.5 Incluir archivos en un formulario

### ▪ **<button>**

Etiqueta para insertar botones en un formulario HTML.

```
<button type="reset">
<button type="submit">
<button type="button">
```

Es similar a utilizar:

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

Pero la diferencia es que permite algunas opciones adicionales, como por ejemplo incluir texto con formato en su interior, o una imagen.

Por ejemplo:

```
<button type="submit">emoji </button>
```



```
<button type="submit">Pulsa <strong> aquí</strong> para avanzar</button>
```

Pulsa aquí para avanzar

### ▪ **<textarea>**

Etiqueta de formulario que permite introducir un campo de texto extenso.

```
<textarea> Escribe aquí tu texto</textarea>
```

Muchos de sus atributos son comunes a los de input (name, maxlength, required, readonly, disabled, autofocus...) Otros atributos básicos de <textarea>:

ATRIBUTO	SIGNIFICADO	EJEMPLO
cols	Número de columnas de la región del texto	<textarea cols="10" rows="20" name="miArea"> Escribe aquí tu texto</textarea>
rows	Número de filas de la región del texto.	



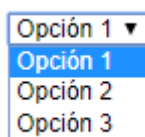
## ▪ <select><option>

Etiquetas de formulario que permiten hacer campos con [listas desplegables](#). La etiqueta <select> define la lista y encierra todas las opciones que muestra la lista. Cada una de las opciones de la lista se define mediante una etiqueta <option>.

El formato es el siguiente:

```
<select name="lista">
  <option value="op1">Opción 1</option>
  <option value="op2">Opción 2</option>
  <option value="op3">Opción 3</option>
</select>
```

Este es el resultado:



Los atributos name y value también se utilizan en las listas desplegable:

Name : atributo de <select> que indica el nombre del dato. Como en todos los campos de formulario, es obligatorio si se quiere identificar el dato que se está mandando.

Value: atributo de <option> que especifica el valor que tendrá un dato. Si el usuario elige esa opción de la lista desplegable, el valor asignado a esa opción será el que se mande asociado al nombre del dato correspondiente

Nombre del dato

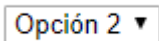
```
<select name="lista">
  <option value="op1">Opción 1</option>
  <option value="op2">Opción 2</option>
  <option value="op3">Opción 3</option>
</select>
```

Posibles valores

El dato se se enviaría es el par `lista=op1`, ya que:

name= lista  
value= op1 (suponiendo que el usuario ha elegido esa opción)

Muchos de sus atributos son comunes a los de input (required, readonly, disabled, autofocus...). Otros atributos básicos de <select> y <option>:

ATRIBUTO	SIGNIFICADO	EJEMPLO
selected	Atributo de <opcion> que indica que es la opción seleccionada por defecto. Es un atributo booleano.	<pre>&lt;select name="lista"&gt;   &lt;option value="op1"&gt;Opción 1&lt;/option&gt;   &lt;option value="op2" selected&gt;Opción 2&lt;/option&gt;   &lt;option value="op3"&gt;Opción 3&lt;/option&gt; &lt;/select&gt;</pre> 

size	Atributo de <select> que indica cuántas opciones son visibles, sin desplegar la lista. Cuando no se pone este atributo, por defecto se ve sólo la primera opción y es necesario desplegar la lista para ver el resto.	<a href="#">Ver ejemplo abajo</a>
multiple	Atributo de <select> que permite escoger más de una opción de la lista desplegable. Cuando se envíe el dato har hacer "submit" del formulario, se enviarán tantos pares nombre=valor como opciones se hayan seleccionado. Es un atributo booleano.	<a href="#">Ver ejemplo abajo</a>

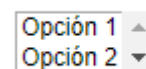
### Ejemplo size:

```
<select name="lista">
  <option value="op1">Opción 1</option>
  <option value="op2">Opción 2</option>
  <option value="op3">Opción 3</option>
</select>
```



Si se indica el atributo size, aparecerán visibles varias opciones:

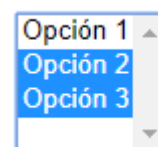
```
<select name="lista" size="2">
  <option value="op1">Opción 1</option>
  <option value="op2">Opción 2</option>
  <option value="op3">Opción 3</option>
</select>
```



### Ejemplo multiple:

```
<select name="lista" multiple>
  <option value="op1">Opción 1</option>
  <option value="op2">Opción 2</option>
  <option value="op3">Opción 3</option>
</select>
```

Si se seleccionan dos opciones:



Se enviarán los pares `lista=op2` y `lista=op3`.

Si se está usando el método get así es como se verá la barra de direcciones:

```
pag.html?lista=op2&lista=op3
```

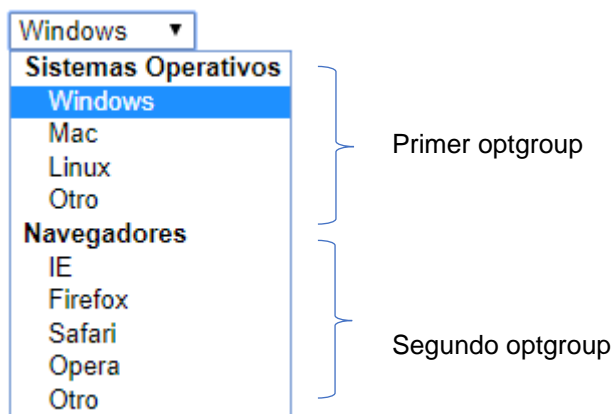
### ▪ <optgroup>

Etiqueta que puede utilizarse dentro de <select> para agrupar varias opciones. Indicando un valor al atributo label se consigue poner un título a cada agrupación.

Ejemplo:

```
<select name="programa">
  <optgroup label="Sistemas Operativos">
    <option value="Windows">Windows</option>
    <option value="Mac">Mac</option>
    <option value="Linux">Linux</option>
    <option value="Other">Otro</option>
  </optgroup>
  <optgroup label="Navegadores">
    <option value="Internet Explorer">IE</option>
    <option value="Firefox">Firefox</option>
    <option value="Safari">Safari</option>
    <option value="Opera">Opera</option>
    <option value="Other">Otro</option>
  </optgroup>
</select>
```

Aparecería así:



## Actividad 23. Formularios avanzados

23.1 Crear formulario con botones con imagen

23.2 Añadir área de texto

23.3 Añadir lista desplegable

23.4 Categorías de lista desplegable

23.5 Opciones de lista desplegable

- **<fieldset><legend>**

Etiquetas que permiten dar formato a un formulario, agrupando varios campos y poniéndoles un título.

Formato:

```
<fieldset>
  <legend>Título</legend>
  Campos de formulario
</fieldset>
```

Ejemplo:

```
<fieldset>
  <legend>Agrupación de varios controles:</legend>
  Campo de texto1:
  <br><input type="text" name="texto1"><br>
  Campo de texto2:
  <br><input type="text" name="texto2"><br>
</fieldset>
<fieldset>
  <legend>Agrupación de botones:</legend>
  <input type="reset">
  <input type="submit">
</fieldset>
```

Tendría el aspecto:

- **<label>**

Etiqueta que permite agrupar el texto descriptivo y el campo de formulario asociado.

Proporciona más coherencia y accesibilidad al formulario.

Al pulsar sobre el texto descriptivo, el foco se sitúa sobre el campo de formulario asociado.

```
<label for="nom">Nombre y apellidos:</label>
  <input type="text" id="nom" name="texto">
```

*Asociación mediante los atributos for e id*

## Actividad 24. Etiquetas complementarias de formulario

### 24.1 Categorías dentro del formulario

### 24.2 Incluir la etiqueta <label>