

## TEMA 2. INTRODUCCIÓN AL HTML

### 1. ¿Qué es HTML?

Definiéndolo de forma sencilla, **HTML** es el lenguaje que se utiliza para crear páginas web. Los navegadores que utilizamos los usuarios muestran las páginas web después de leer su contenido **HTML**. **HTML** son las siglas de *HyperText Markup Language*.

El lenguaje **HTML** es un estándar reconocido en todo el mundo y cuyas normas define un organismo sin ánimo de lucro llamado [World Wide Web Consortium](#), más conocido como **W3C**. Como se trata de un estándar reconocido por todas las empresas relacionadas con el mundo de Internet, una misma página HTML se visualiza de forma muy similar en cualquier navegador de cualquier sistema operativo.

El propio **W3C** define el lenguaje **HTML** como "un lenguaje reconocido universalmente y que permite publicar información de forma global".

#### 1.1. Breve historia de HTML

El origen de **HTML** se remonta a 1980, cuando el físico **Tim Berners-Lee**, trabajador del [CERN](#) (*Organización Europea para la Investigación Nuclear*) propuso un nuevo sistema de "hipertexto" para compartir documentos.

El primer documento formal con la descripción de **HTML** se publicó en 1991 bajo el nombre *HTML Tags* (*Etiquetas HTML*) y todavía hoy puede ser consultado online a modo de *reliquia informática*.

A partir de ese momento van apareciendo diferentes versiones de HTML, siendo el [W3C](#) (*World Wide Web Consortium*) el encargado de publicar estos estándares.

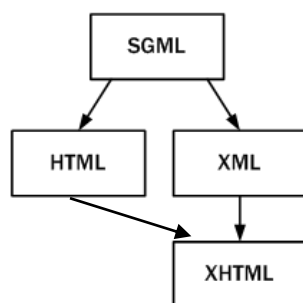
HTML 4.0 se publicó el 24 de Abril de 1998 y supone un gran salto desde las versiones anteriores. Entre sus novedades más destacadas se encuentran las hojas de estilos CSS, la posibilidad de incluir pequeños programas o *scripts* en las páginas web, tablas complejas y mejoras en los formularios.

Poco después, apareció HTML 4.01 con pocas novedades significativas.

Desde la publicación de HTML 4.01, no apareció nada nuevo hasta 2009 que aparece HTML 5.

#### 1.2. HTML y XHTML

El lenguaje **XHTML** es muy similar al lenguaje **HTML**. De hecho, **XHTML** no es más que una adaptación de **HTML** al lenguaje **XML**. Técnicamente, **HTML** es descendiente directo del lenguaje **SGML**, mientras que **XHTML** lo es del **XML** (que, a su vez, también es descendiente de **SGML**).



Las páginas y documentos creados con XHTML son muy similares a las páginas y documentos HTML.

## 2. Edición básica en html

Un documento HTML contiene texto. Ahora bien, hay texto que sirve para indicar **elementos** y propiedades de la página. Es decir, el texto contiene también lo que se denomina **etiquetas**.

Las etiquetas sirven para delimitar elementos de la página. Por ejemplo, un párrafo es un elemento de la página, una tabla también. Incluso hay elementos que contienen otros elementos (las tablas constan de filas y las filas de celdas...).

Las etiquetas son textos encerrados entre los signos de mayor y menor (< >). Cuando un navegador se encuentra un texto así encerrado, entenderá que dentro de los símbolos menor y mayor lo que se indica es el inicio de un elemento. El inicio del elemento se marca con el nombre del elemento entre los símbolos < >.

**Ejemplo:**

<strong>

La etiqueta **strong** sirve para indicar texto importante (se suele mostrar en negrita).

La mayoría de las etiquetas afectan a un determinado texto, el cual estará *encerrado* por las etiquetas. Por lo tanto, casi siempre, existe una etiqueta de apertura y otra de cierre, que se interpretará como el inicio y fin de un determinado elemento de la página.

El cierre de una etiqueta se marca poniendo el símbolo /, seguido del nombre del elemento que estamos cerrando.

**Ejemplo:**

Texto normal <strong>texto negrita</strong>

Ya se ha comentado que un elemento puede contener dentro más elementos, por lo que entre la apertura y el cierre de una etiqueta puede haber más etiquetas. Una norma importante es que siempre debemos de cerrar primero las etiquetas abiertas más tarde.

Es decir, este código **no es correcto**.

Texto normal <strong><em>texto negrita y cursiva</strong></em>

Se está cerrando la etiqueta *em* después de *strong* y eso no es correcto porque *em* es la última que se abrió. **Lo correcto es:**

Texto normal <strong><em>texto negrita y cursiva</em></strong>

Es decir, se cierra primero el elemento **em** que fue el último en abrirse.



## Atributos

Algunas etiquetas tienen atributos. Los atributos son propiedades de cada elemento a las que podemos asignar un valor, de modo que dicho valor varía el comportamiento del elemento. La forma de utilizar un atributos es:

```
<elemento nombreDeAtributo1="valor1" nombreDeAtributo2="valor1" ... >
```

A los atributos se les asignan valores que deben ir entre comillas. El símbolo de igualdad (=) es obligatorio.

**Ejemplo:**

```
<p align="center">
```

Indica que el elemento está alineado al centro.

La sintaxis de atributos es:

```
<etiqueta atributo1="valor1" atributo2="valor2" ...>
....
</etiqueta>
```

En el cierre sólo se cierra la etiqueta, sin indicar los atributos. Ejemplo:

## 3. Estructura de un documento HTML

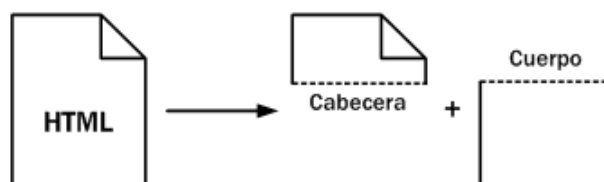
**HTML**, como ya hemos comentado, es un **lenguaje de etiquetas** (también llamado **lenguaje de marcado**) y las páginas web habituales están formadas por cientos o miles de pares de etiquetas. Además de HTML, existen muchos otros lenguajes de etiquetas como XML y SGML.

La principal ventaja de los lenguajes de etiquetas es que son muy sencillos de leer y escribir por parte de las personas. La principal desventaja es que pueden aumentar mucho el tamaño del documento, por lo que en general se utilizan etiquetas con nombres muy cortos.

Un documento **HTML** se divide en dos partes principales: **la cabecera y el cuerpo**.

La **cabecera** incluye información sobre la propia página, como por ejemplo su título y su idioma.

El **cuerpo** de la página incluye todos sus contenidos, como párrafos de texto e imágenes.

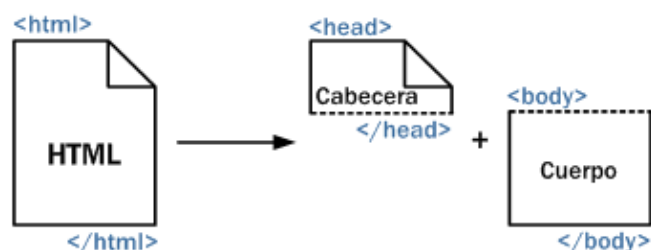


El cuerpo (llamado *body* en inglés) contiene todo lo que el usuario ve en su pantalla y la cabecera (llamada *head* en inglés) contiene todo lo que no se ve (con la única excepción del título de la página, que los navegadores muestran como título de sus ventanas).

### Estructura básica de una página en HTML5

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Primer Ejercicio HTML 5</title>
</head>
<body>
  <p>Esperamos que <strong> que el curso</strong> vaya bien y estéis cómodo
  </p>
</body>
</html>
```

- La primera etiqueta es **<!DOCTYPE>** Esta etiqueta indica que versión de HTML se utiliza en el documento, para que los navegadores puedan aplicar la gramática y sintaxis correspondiente al documento. En nuestro ejemplo le estamos indicando que vamos a usar **HTML 5**.
- Seguidamente se encuentra la etiqueta **<html lang="es">**, **html** indica el comienzo y el final de un documento (**</html>**). Ninguna etiqueta o contenido puede colocarse antes de la etiqueta **<html>** ni después de la etiqueta **</html>**(con la excepción del tipo). En el interior de la etiqueta **<html>** se definen la cabecera y el cuerpo del documento.
- **<head>**: delimita la parte de la cabecera del documento. La cabecera contiene información sobre el propio documento HTML, como por ejemplo su título y el idioma de la página. Los contenidos indicados en la cabecera no son visibles para el usuario, con la excepción de la etiqueta **<title>**, que se utiliza para indicar el título del documento y que los navegadores lo visualizan en la parte superior de la ventana del navegador. También tenemos la etiqueta **<meta charset="utf-8">** esta etiqueta se utiliza para especificar el juego de codificación de caracteres que se ha de utilizar para realizar la visualización de la página.
- **<body>**: delimita el cuerpo del documento HTML. El cuerpo encierra todos los contenidos que se muestran al usuario (párrafos de texto, imágenes, tablas). En general, el **<body>** de un documento contiene cientos de etiquetas HTML, mientras que el **<head>** no contiene más que unas pocas.



## Etiquetas obsoletas de html

De todas las etiquetas disponibles en html, las siguientes se consideran **obsoletas** y no se deben utilizar: **applet, basefont, center, dir, isindex, menu, strike, acronym**.

## 4. Elementos de texto y párrafo

Hoy en día en una página web, todo el texto está dentro de alguna etiqueta, en realidad hay tres contextos principales a tener en cuenta sobre el texto:

- **Elementos capaces de contener varios párrafos.** Cada elemento de esta parte es lo que hoy en día se conoce como **capa** o **sección**. Si no indicamos capa alguna, entonces todos los párrafos están contenidos en la etiqueta **body**, lo que significa que el contenedor de los párrafos será el área que el navegador deja para el texto y que puede ser casi toda la pantalla.
- **Elementos que contienen párrafos individuales.** Aunque en HTML 5 se permite escribir texto que no esté dentro de un párrafo, no es una buena práctica, ya que no podríamos indicar más adelante el formato del párrafo (alineaciones, márgenes...).  
Lógicamente dentro de un párrafo, no podemos indicar otro párrafo; es decir no podemos meter una etiqueta **p**, dentro de una etiqueta **h1**.
- **Elementos que contienen texto simple o conjuntos de caracteres** Son elementos pensados para ser interiores a un párrafo. Un mismo párrafo puede contener muchos elementos de este tipo. El mismo texto puede estar contenido en varios elementos de este tipo a la vez. Por ejemplo, podemos indicar que un texto es **strong** y **em** a la vez.

### 4.1. Etiquetas básicas de párrafo y texto

Desde hace años todo texto en un documento HTML debe ir dentro de una etiqueta que sirva para decir qué tipo de texto es. Y eso significa, que el texto debe de ir dentro de una etiqueta de párrafo. Se comentan las etiquetas disponibles para marcar párrafos.

#### Párrafo simple

La norma HTML estándar establece que el texto de la página debe estar contenido dentro de un elemento. Es decir, no se puede poner texto directamente dentro de la etiqueta *body*; ha de haber un contenedor para ese texto.

El elemento más sencillo lo marca la etiqueta **p**, que indica un elemento de párrafo normal.

Normalmente los navegadores utilizan fuentes tipo **Times de tamaño 11pt** para la letra de párrafo normal.

**Ejemplo de uso:**

```
<body>
```

```
<p>
```

Párrafo con un poco de texto

```
</p>
```

```
<p>
```

Este es un párrafo completo metido dentro de una etiqueta **<em>p</em>**, aunque

el texto se escriba

con todos estos saltos de línea y espacios

se mostrará seguido,

sin saltos de

línea interiores

y entre cada palabra dejando un

solo espacio

```
</p>
```

```
</body>
```

En el navegador saldría:

Párrafo con un poco de texto

Este es un párrafo completo englosado dentro de una etiqueta *p*, aunque el texto se escriba con todos estos saltos de línea y espacios se mostrará seguido, sin saltos de línea interiores y entre cada palabra dejando un solo espacio

En la imagen se observa el modo en el que el navegador muestra el párrafo. Por defecto los párrafos tipo **p** dejan espacio arriba y abajo (normalmente media línea en cada dirección).

### Salto de línea

A veces es necesario dentro del texto de un determinado párrafo hacer un salto de línea. El elemento que lo realiza no tiene cierre y se llama **br**.

**Ejemplo:**

```
<p>Primera línea <br>Segunda línea</p>
```

Aparecerá cada texto en una línea distinta.

Primera línea

Segunda línea

## Línea horizontal

Otra posibilidad es hacer un salto, pero dejando una línea horizontal en el hueco de las palabras. Esto lo hace la etiqueta **hr** (que tampoco tiene cierre):

```
<p>Primera línea <hr>Segunda línea</p>
```

Aunque los navegadores entienden este código. En realidad, **hr** tiene que estar fuera de las etiquetas de párrafo, es decir lo correcto es:

```
<p>Primer párrafo</p>
<hr>
<p>Segundo párrafo</p>
```

Resultado:

Primer párrafo

---

Segundo párrafo

## Atributos de hr

- **size:** permite variar el grosor de la línea.
- **width:** permite variar la longitud de la línea.

Ejemplo de uso:

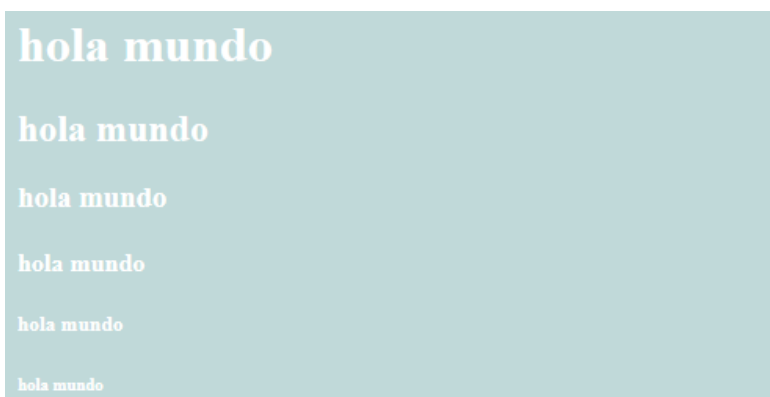
```
<body>
<hr width="50%" size="5">
```

## Títulos

Hay una serie de siete etiquetas que comienzan con la letra *h* a la que le sigue un número del 1 al 6. Sirven para marcar párrafos de forma que se considerarán títulos del texto. De modo que el elemento **h1** marcará títulos de primer nivel, **h2** de segundo nivel, ... y así hasta **h6** (sexto nivel).

```
Ejemplo:
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Estrellas</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>
    <h2>hola mundo</h2>
    <h3>hola mundo</h3>
    <h4>hola mundo</h4>
    <h5>hola mundo</h5>
    <h6>hola mundo</h6></body>
</html>
```

El resultado sería:



### Elementos de marcado de texto simple

En este caso, detallaremos los elementos que se emplean para señalar texto interior a un párrafo.

#### Texto importante

En HTML 5 el texto importante se marca con el elemento **strong**. En general todos los navegadores marcan el texto *strong* en negrita (letra más gruesa).

Sin duda, **strong** es uno de los elementos HTML fundamentales e imprescindible en los documentos HTML.

#### Texto enfatizado

El elemento **em** marca texto como enfatizado. El resultado visual con que le dan formato por defecto los navegadores es texto en cursiva.

La idea semántica es marcar con **em** texto con énfasis de segundo nivel (dejando a **strong** el primer nivel) que normalmente se dedica a palabras no aceptadas aun, títulos de trabajos, apodos, texto subjetivo, etc.



### texto resaltado

Se trata del tercer elemento de HTML5 para remarcar texto. Se hace a través el elemento **mark**. En este caso, su efecto visual, suele ser aplicar en el texto así marcado un fondo de color vivo que simula el uso de un subrayador.

#### Ejemplo:

```
<p> <strong>Este texto esta con negrillas</strong></p>  
<p><u>Este texto esta con subrayado</u></p>  
<p> <em>Este texto esta con inclinación itálica</em></p>  
<p>Yo soy texto normal, <mark>y yo estoy remarcado</mark></p>
```

Obtendría el resultado:

Este texto esta con negrillas

Este texto esta con subrayado

*Este texto esta con inclinación itálica*

Yo soy texto normal, y yo estoy remarcado

Este elemento no es tan antiguo como los anteriores porque apareció con HTML5 (los navegadores antiguos no lo reconocen). La idea semántica es marcar texto que queremos realzar de forma muy llamativa indicando texto a recordar o sobre el que profundizar.

### Elemento s

En la versión de HTML 4 quedó obsoleto, no se recomendaba su uso. Con HTML 5 se ha redefinido su uso para marcar texto obsoleto que, visualmente, se muestra con efecto de tachado.

#### Ejemplo:

```
<p>La capital de Alemania es <s>Bonn</s> Berlín</p>
```

Resultado:

La capital de Alemania es ~~Bonn~~ Berlín

### Elemento b

Se trata del elemento histórico (de las primeras versiones de HTML) utilizado para marcar texto en negrita. En HTML 4 quedó como obsoleto. En HTML 5 se volvió a permitir. En definitiva, elemento de uso muy poco aconsejable que se mantienen solo por la gran cantidad de creadores que lo siguen utilizando.

### Elemento *i*

Tiene el mismo problema que el anterior. Es el elemento histórico (de las primeras versiones de HTML) utilizado para marcar texto en cursiva. En HTML 4 quedó como obsoleto. En HTML 5 se puede utilizar para marcar texto tipo alternativo si no encaja con otros elementos de marcado de caracteres.

Al igual que ocurre con el elemento **b**, su uso no es aconsejable. Se puede usar **em**.

#### Ejemplo:

```
<p><b>Este texto está en negrita (etiqueta b).</b> <i> Este texto está en cursiva(etiqueta i)</i></p>
```

### Elemento *u*

Los navegadores muestran este texto con un subrayado.

#### Ejemplo:

```
<p><u>Este texto esta con subrayado</u></p>
```

## Elementos de marcado de caracteres

### Subíndices

Permite que el texto aparezca por debajo de la línea base y en un tamaño más pequeño. Lo hace el elemento **sub**.

#### Ejemplo:

```
<p>La fórmula del agua es H<sub>2</sub>O</p>
```

Mostraría: H<sub>2</sub>O

### Superíndices

Parecida al anterior, pero ahora el texto marcado con el elemento **sup** aparecerá por encima y en pequeño.

#### Ejemplo:

```
<p>La fórmula del agua es H<sup>2</sup>O</p>
```

Obtendría: H<sup>2</sup>O

### Elemento *small*

Marcado de **letra pequeña**. Se usa, por ejemplo, en textos que marquen copyright, derechos de uso, comentarios, notas anexas y letra pequeña (tipo la de los contratos) en general. El efecto visual es que el texto sale con un tamaño más pequeño. Es posible incluso anidar un elemento **small** dentro de otro, de modo que cada vez sale más pequeño el texto.

**Ejemplo:**

```
<p>Texto normal <small>Pequeño <small>Aún más pequeño</small></small></p>
```

Resultado:

Texto normal Pequeño Aún más pequeño

### Abreviaturas con *abbr*

Marca texto que indica una **abreviatura o un acrónimo**. Su uso requiere utilizar el atributo **title** para indicar el significado de la abreviatura. En general, los navegadores no muestran ningún resultado, pero sí un cartel con el contenido de **title** cuando se arrima el ratón.

**Ejemplo:**

```
<p>Se ha coronado a <abbr title=" Su Alteza Real">SAR</abbr> el Rey Pepe Juan</p>
```

Se ha coronado a SAR el Rey Pepe Juan

### Definiciones

El elemento **dfn** se usa de forma similar al anterior. Sirve para indicar definiciones para el término o palabras que encuadra.

**Ejemplo:**

```
<dfn title="usar dos conceptos de significado opuesto en una sola expresión">oxímoron</dfn>
```

Los navegadores suelen mostrar el contenido del elemento en cursiva y, como siempre, en un cartelito muestran el contenido del atributo **title**:

*oxímoron*

usar dos conceptos de significado opuesto en una sola expresión

## Fuentes

La etiqueta **font** nos indica el tamaño, color, o fuente del texto que contiene.

Sus etiquetas son: `<font>` y `</font>` (ambas obligatorias). Los atributos son:

- **Size:** tamaño de la fuente (valores del 1 al 7)
- **Color:** color del texto.
- **Face:** fuente para el texto.

### Ejemplo:

```
<p><font color="green"> Este texto es verde </font></p>
```

```
<p><font color="green" size="5" face="verdana"> Este texto es verde, de tamaño 5 y  
verdana </font></p>
```

## Elemento body

Sólo puede haber un documento **body** en un documento html.

Atributos:

- **Background:** URL de una imagen para usar como fondo.
- **Bgcolor:** color de fondo del documento.
- **Text:** color del texto

### Ejemplo:

```
<body bgcolor="#C0D9D9" text="white">
```

## Escritura de código o texto informático

### Escritura de código

HTML proporciona un elemento llamado **code** que se utiliza cuando en un documento deseamos escribir ejemplos de código escritos en un lenguaje de programación.

### Ejemplo:

```
<p>Este es un ejemplo de código en JavaScript:</p>
```

```
<code>  
    window.onload=function(){  
        alert("Hola");  
    }  
</code>
```

Lo cual obtendría este resultado:

Este es un ejemplo de código en JavaScript:

```
window.onload=function(){ alert("Hola"); }
```

Se puede observar cómo los navegadores modifican el tipo de letra (utilizan un tipo de letra monoespaciada) pero no respetan los espacios y tabuladores utilizados para indicar el código. Como es muy normal querer respetar los espacios y tabuladores cuando se escriben ejemplos de código, lo habitual es combinar con la etiqueta **pre**.

La etiqueta **pre** representa un bloque de texto preformateado donde, los espacios continuos y los saltos de línea son respetados.

**Ejemplo.**

```
<p>Este es un ejemplo de código en JavaScript:</p>
<pre><code>
    window.onload=function(){
        alert("Hola");
    }
</code></pre>
```

Y así obtenemos este otro resultado:

Este es un ejemplo de código en JavaScript:

```
    window.onload=function(){
        alert("Hola");
    }
```

### Resultados de operaciones en un ordenador

Con intenciones similares al elemento anterior, tenemos el elemento **samp**. Se utiliza cuando queremos en un documento indicar ejemplos de resultados de operaciones en un ordenador.

**Ejemplo:**

```
<p>Al ejecutar el comando obtuve el mensaje
<samp>Comando o nombre de archivo incorrecto </samp>
</p>
```

Los navegadores suelen mostrar este elemento en letra monoespaciada. La salida del código anterior suele ser:

```
Al ejecutar el comando obtuve el mensaje Comando o nombre
de archivo incorrecto
```

## Citas

Formas de marcar citas en html5:

- **q**. Usada para indicar **citas cortas** incluidas en un párrafo con otro tipo de texto. Indica un texto que procede de otra fuente. Remarca una cita, con la connotación de “texto entrecomillado”. De hecho, los navegadores, por defecto, entrecomillan su contenido. Es un elemento de línea.

### Ejemplo:

```
<p>Como dijo Julio César, <q  
cite="http://es.wikipedia.org/wiki/Alea_iacta_est">  
Alea Jacta es</q></p>
```

No es obligatorio utilizar el atributo **cite**. Este atributo permite indicar una fuente para consultar la cita original (los navegadores, sin embargo, no hacen nada con el contenido de este enlace).

- **blockquote**. Crea citas en bloque, marca las citas a otros autores o documentos. Es decir, indica párrafos completos que son citas procedente de fuentes externas al documento. Se le puede indicar (al igual que se hace con el elemento **q**) el atributo **cite** para indicar un enlace a la fuente de la cita.
  - `<blockquote>` no genera automáticamente comillas alrededor del contenido.
  - Por defecto, `<blockquote>` incluye un margen izquierdo cuando se visualiza.

### Ejemplo:

```
<blockquote  
cite="http://noticias.juridicas.com/base_datos/Admin/constitucion.tp.html#a1">  
<strong>Artículo 1 de la constitución</strong><br>  
España se constituye en un Estado social y democrático de  
Derecho...</blockquote>
```

Puede utilizar el atributo **cite** (como se observa en el ejemplo), para indicar la dirección URL que se ha utilizado como fuente de la cita.

Visualmente los navegadores muestran el texto **blockquote** añadiendo una sangría a la derecha y a la izquierda. Ignoran, visualmente, el uso del atributo **cite**.

## span

Sirve para marcar un trozo de texto de forma inespecífica, sin indicar semántica alguna. En definitiva, selecciona un trozo de texto por necesidades técnicas (por ejemplo, para indicar un formato CSS diferente a ese texto).

### Elemento address

El elemento **address** permite indicar una dirección, especialmente indicada para mostrar información sobre el autor o los créditos del documento (con información de los autores, por ejemplo) al estilo de los pies de artículos periodísticos.

Aunque, de forma clásica, se le considera un elemento de marcado de párrafos, actualmente se le ha denotado como un elemento de marcado de secciones.

**Ejemplo:**  
<address>Texto escrito por IES Castelar<br>  
Avda. Ramón y Cajal, 2<br>  
Badajoz<br>  
España  
</address>

Los navegadores suelen mostrar este tipo de párrafos en cursiva y centrados.

## 5. Listas

Las listas permiten crear párrafos agrupados y alineados mediante símbolos como viñetas o números para facilitar la lectura y organización de las ideas del documento.

### Listas con viñetas (no ordenadas)

Las listas con viñetas se deben englobar dentro de un elemento **ul** (acrónimo de *unordered list*, lista no ordenada), después cada párrafo de la lista estará dentro de elementos de tipo **li** (de *list item*, elemento de lista).

**Ejemplo:**  
<p>Lista de la compra</p>  
<ul>  
<li>Agua</li>  
<li>Vino</li>  
<li>Cerveza</li>  
</ul>

**Resultado:**

Lista de la compra

- Agua
- Vino
- Cerveza

## Listas numéricas (ordenadas)

Las listas numéricas aparecen dentro del elemento **ol** (de *ordered list*, lista ordenada), después cada párrafo de la lista estará dentro de elementos de tipo **li**, al igual que las anteriores. La diferencia ahora es que cada párrafo con **li**, aparece con un número y no con una viñeta.

### Ejemplo:

```
<p>Lista de la compra</p>
<ol>
  <li>Agua</li>
  <li>Vino</li>
  <li>Cerveza</li>
</ol>
```

Resultado:

Lista de la compra

1. Agua
2. Vino
3. Cerveza

## Listas anidadas

Es posible meter una lista dentro de otra, por ejemplo:

```
<p>Lista de la compra</p>
<ul>
  <li>
    No alcohólicas
    <ul>
      <li>Agua</li>
    </ul>
  </li>
  <li>
    Alcohólicas
    <ul>
      <li>Vino</li>
      <li>Cerveza</li>
    </ul>
  </li>
</ul>
```



Con el resultado:

- Lista de la compra
- No alcohólicas
    - Agua
  - Alcohólicas
    - Vino
    - Cerveza

También es posible anidar mezclando tipos de listas:

```
<p>Lista de la compra</p>
<ol>
  <li>
    No alcohólicas
    <ul>
      <li>Agua</li>
    </ul>
  </li>
  <li>
    Alcohólicas
    <ul type="square">
      <li>Vino</li>
      <li>Cerveza</li>
    </ul>
  </li>
</ol>
```

Con el resultado:

- Lista de la compra
1. No alcohólicas
    - Agua
  2. Alcohólicas
    - Vino
    - Cerveza

## Listas de términos

Permite crear una lista de definiciones de términos. En ellas se indica el término a definir y su significado.

### Ejemplo:

```
<dl>
  <dt>Windows</dt>
  <dd>
    Sistema operativo de <strong>Microsoft</strong> disponible para PC
    disponible en versiones de 32 y 64 bits y para servidores,
    ordenadores e incluso tabletas y móviles.<br>
    La última versión es la versión 10 y la 2016 para servidores.
  </dd>
  <dt>Linux</dt>
  <dd>
    Sistema operativo de código abierto disponible
    en numerosas distribuciones gratuitas y de pago.
    Es la base del sistema <strong>Android</strong>.
  </dd>
  <dt>Mac OS</dt>
  <dd>
    Sistema operativo de los ordenadores de la empresa <strong>Apple</strong>
    <br>
    La última versión es <strong>Mojave</strong>
  </dd>
</dl>
```

Resultado:

|         |  |
|---------|--|
| Windows | Sistema operativo de <b>Microsoft</b> disponible para PC disponible en versiones de 32 y 64 bits y para servidores, ordenadores e incluso tabletas y móviles.<br>La última versión es la versión 10 y la 2016 para servidores. |
| Linux   | Sistema operativo de código abierto disponible en numerosas distribuciones gratuitas y de pago. Es la base del sistema <b>Android</b> .  |
| Mac OS  | Sistema operativo de los ordenadores de la empresa <b>Apple</b><br>La última versión es <b>Mojave</b>  |

## 6. Imágenes

Las imágenes son fundamentales para que una página web sea más atractiva. Son el primer contenido no textual que se planteó poder incorporar al estándar HTML. Prácticamente no hay páginas sin imágenes.

Los navegadores tienen capacidad de mostrar imágenes, pero sólo las que pertenezcan a tipos concretos. Los tipos de imágenes más usados y reconocidos por la mayoría de los navegadores son:

- **Formato jpg.** Son imágenes que ocupan muy poco espacio gracias a su alta compresión. No admiten animaciones ni zonas marcadas con transparencia. Son el formato habitual de la imagen digital en una página web.

- **Formato gif.** Imágenes con hasta 256 colores, por lo que son buenas para dibujos y logotipos, pero no para fotografías. Pueden incluso contener animaciones (uno de sus usos habituales).
- **Formato png.** Imágenes fotográficas comprimidas al estilo de jpg pero con más calidad (ocupan más, normalmente)
- **Formato webp.** Formato de imagen desarrollado por **Google** para conseguir mejorar a PNG y JPG haciendo que ocupen menos sin pérdida de calidad respecto a estos formatos. Sin embargo, no todos los navegadores son capaces de interpretarlas.

Las imágenes son fundamentales en las páginas web, su elección resulta vital para la estética de esta, nunca se considera un mero acompañamiento ya que el impacto visual y las sensaciones sobre la profesionalidad de la página siempre le consiguen las imágenes con ayuda de la tipografía (pero ahí ya entra el lenguaje CSS) y la disposición o maquetación (para lo que también necesitamos CSS).

Las imágenes se colocan mediante la etiqueta **img**. El atributo **src** indica la URL (relativa o absoluta) a la imagen.

#### Ejemplo:

```

```

**img** es una etiqueta **sin cierre**, la imagen se coloca directamente en la posición de esta etiqueta. Se la considera como texto al maquetarla. De modo que aparece entre el texto si ponemos la etiqueta entre el texto.

#### Atributos de la etiqueta **img**

| atributo      | significado   |
|---------------|---|
| <b>alt</b>    | <p><b>Obligatorio.</b> Indica un texto alternativo. Ese texto aparece cuando la imagen no se ha podido cargar (o durante la carga).</p> <p>Es un texto también tenido en cuenta por los buscadores a fin de identificar lo que muestra la imagen.</p> <p>Deberíamos tomarnos este atributo como obligatorio</p> |
| <b>width</b>  | Anchura de la imagen.   |
| <b>height</b> | Altura de la imagen.  |

**Ejemplo:**

```
<h2>Primera imagen</h2>
 <br>
<h2>Segunda imagen</h2>
<!--El texto alt aparece cuando la imagen no se puede mostrar-->
 Emoticono
<hr>
```

## 7. Mapas de imagen

Se trata de una técnica que permite dividir en trozos una imagen, de modo que podremos asignar un enlace a cada trozo.

Se utiliza en mapas propiamente dichos, en los que el usuario selecciona regiones del mapa y en imágenes donde hay elementos claramente destacados que al hacer clic nos permitan ir a una nueva URL.

Hoy en día está en desuso ya que el uso del lenguaje JavaScript ofrece capacidades muy superiores a los mapas de imágenes; pero la ventaja de los mapas es que no requieren conocer ningún lenguaje aparte de HTML y a veces se utiliza.

Para crear un mapa de imagen se usa la etiqueta **map**.

Los mapas se basan en una imagen previamente colocada en la página web mediante la etiqueta **img**. A partir de esa imagen se crea un elemento nuevo en la página de tipo **map**.

El elemento **map** debe tener este atributo:

- **name**. Nombre del mapa de imágenes. Identifica el mapa de imágenes (no se debe de utilizar para este fin el atributo **id**)

### Atributos para mapas en el elemento *img*

Por otro lado, en la etiqueta **img** asociada a la imagen del mapa se debe utilizar este atributo:

- **usemap**. A la que se le da como valor el nombre del mapa (el que se utilice en el atributo **name** del elemento **map**), pero anteponiendo el símbolo **#**.

### Elemento *area*

Con los elementos **img** y **map** sólo se está indicando que la imagen es un mapa, pero falta decir qué zonas tiene el mapa y qué hacer cuando se hace clic en ellas. Esa es la labor de los elementos de tipo **area**.

La etiqueta **map** debe contener dentro tantos elementos **area** como secciones en el mapa queramos crear. En cada elemento *area* rellenaremos estos atributos:

- **coords**. **Coordenadas del área**. La forma de indicarlo depende de lo que se haya elegido en el atributo **shape**.
- **shape**. **Permite indicar la forma del área. Puede ser:**
  - **shape="rect"**. El área será rectangular. En este caso el atributo **coords** indicará cuatro coordenadas. Las dos primeras marcan la coordenada X y la coordenada Y de la esquina superior izquierda de la imagen, las dos últimas las coordenadas

X e Y de la esquina inferior derecha. Las coordenadas cuentan desde la esquina superior izquierda de la imagen (que serían las coordenadas 0,0).

- **shape="circle"**. El área es una región circular en la imagen. El atributo **coords** tendría tres coordenadas: las dos primeras son la coordenada X e Y del centro del círculo y la tercera el radio de este.
- **shape="poly"**. Permite indicar una región poligonal. El atributo **coords** permite indicar las coordenadas X e Y de cada punto del polígono. Al final se cerrará el polígono.
- **area="default"**. Hace lo mismo que la opción **rect**.
- **alt**. Texto alternativo en el área de la imagen. Funciona igual que el atributo **alt** de la etiqueta **img**. No es obligatorio, pero sí muy interesante.
- **href**. Funciona como el atributo del mismo nombre en la etiqueta **a**: indica una URL a la que se dirigirá el navegador cuando el usuario haga clic sobre la imagen. Es decir, cada área del mapa de imagen será un enlace cuyo destino lo marca este atributo (que tiene las mismas posibilidades que el atributo **href** del elemento **a**)

### Ejemplo de mapa

En la siguiente imagen se detallan las coordenadas necesarias para hacer un mapa con tres áreas sobre la imagen siguiente. Como se observa, aunque es una sola imagen, se distingue muy bien visualmente lo que va a ser cada área.

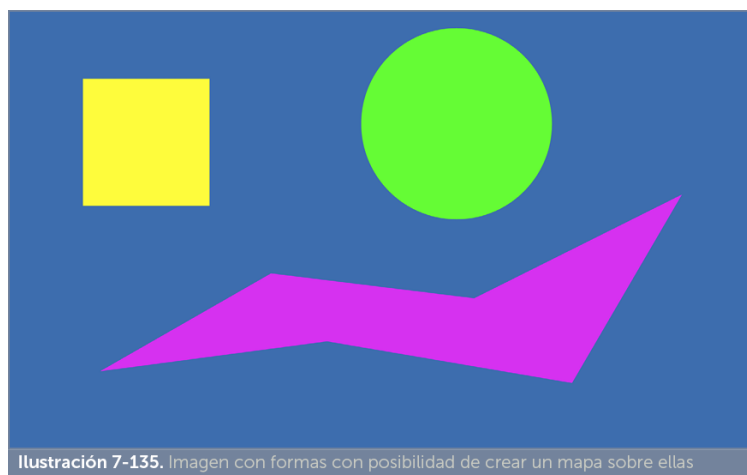
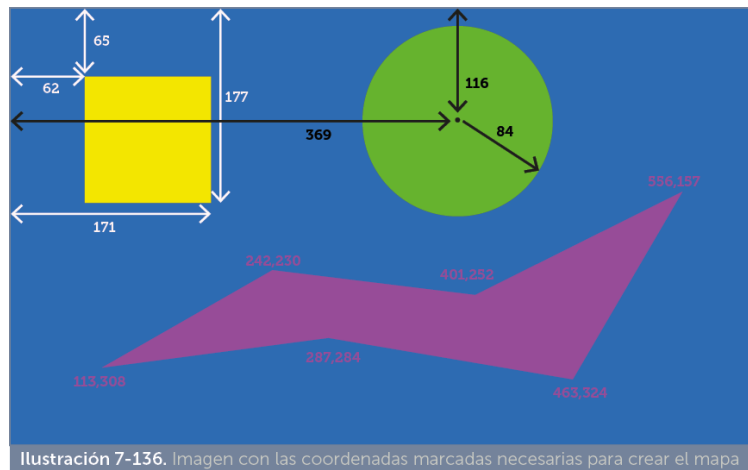


Ilustración 7-135. Imagen con formas con posibilidad de crear un mapa sobre ellas

Si deseamos que, al hacer clic en el cuadrado, el círculo y la forma libre saltemos a una URL, necesitamos conocer las coordenadas que tienen estas formas (en esto nos pueden ayudar los programas de retoque gráfico como *Adobe Photoshop* o *Adobe Fireworks*). En esta imagen se muestran las coordenadas necesarias:



#### Ejemplo.

```

<map name="formas" id="formas">
  <area shape="rect" coords="62,65,171,177"
        href="rectangulo.html" alt="Rectángulo">
  <area shape="circle" coords="369,116,84"
        href="circulo.html" alt="Círculo">
  <area shape="poly"
        coords="113,308,242,230,401,252,556,157,463,324,287,284"
        href="poligono.html" alt="Polígono" >
</map>
```

El uso del atributo **id** en el elemento **map** no es obligatorio para que el mapa funcione, pero sí es necesario para cumplir las nuevas normas HTML donde se intenta retirar el atributo **name** para sustituirle por **id** (aunque los mapas siguen funcionando con el atributo *name*); es decir se aconseja para una posible compatibilidad futura.

## 8. Enlaces

Una URL (**Uniform Resource Location**) es la dirección concreta de un recurso (una página web, una imagen, un vídeo, un directorio...) en Internet.

Cada URL es única, con lo que es un identificador absoluto para un determinado recurso. Si la URL es correcta sólo habrá un solo recurso posible al que se puede referir.

### Enlaces globales

Los enlaces permiten colocar un texto (u otro elemento, como una imagen o un botón) resaltado de forma especial, de modo que cuando se le hace clic, el navegador web nos llevará al destino de la URL (si está disponible).

La etiqueta que permite realizar enlaces es la etiqueta **a**. El atributo **href** permite indicar la URL a la que se realiza el salto.

#### Ejemplo:

```
<p>A Augusto le sucedió el emperador
<a href="http://es.wikipedia.org/wiki/Tiberio">Tiberio</a></p>
```

## Enlaces locales

Muchos enlaces de un sitio web en Internet no saltan a otros sitios, sino que son saltos a recursos del mismo servicio web. Es decir, se salta muy a menudo a recursos del mismo servidor web.

En ese caso no se indica una URL global tal cual se explicó anteriormente, sino que se indica un salto local. Es decir, un salto a un recurso en nuestro propio servidor web que toma como punto de partida el directorio en el que se encuentra la página web que realiza el salto.

Por ejemplo, imaginemos que estamos creando una página web relacionada con la URL <http://www.miservidor.com/noticias/n1.html> y queremos en esa página hacer un enlace a la URL indicada mediante <http://www.miservidor.com/noticias/n2.html>

Podríamos hacerlo con el código:

```
<a href="http://www.miservidor.com/noticias/n2.html">Ver segunda noticia</a>
```

El enlace funcionaría perfectamente. Pero hay que tener en cuenta que la página a la que saltamos está en el mismo directorio. Si algún día movemos todo nuestro sitio web al dominio *nuevodominio.com* resulta que tendremos que cambiar todo el código para hacer frente a la nueva situación. Por ello lo lógico es usar rutas locales.

El enlace usando una ruta local sería:

```
<a href="n2.html">Ver segunda noticia</a>
```

### Ejemplo:

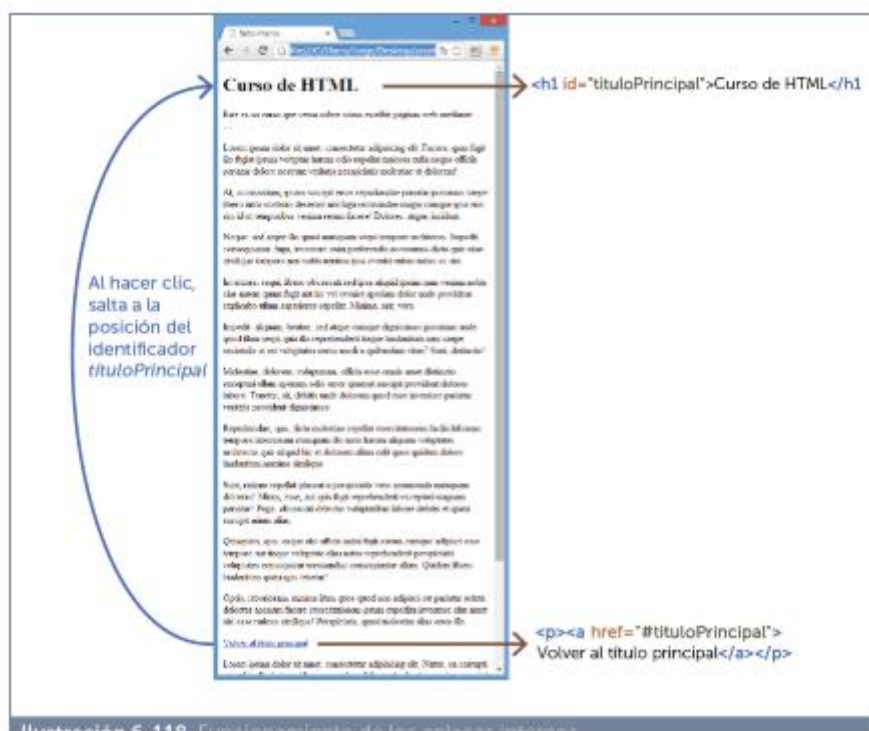
```
<a href="http://www.microsoft.com">Microsoft</a>
<ul>
  <!-- Salto a la página Ejemplo1.html que estará en el mismo directorio que la actual -->
  <li><a href="Ejemplo1.html">Mi primera página Web</a></li>
  <!-- Salto a la página Ejemplo2.html que estará dentro del directorio Ejemplos que estará, a su vez, dentro del directorio actual -->
  <li><a href="Ejemplos/Ejemplo2.html" target="_blank">Mi segunda página Web</a></li>
  <!-- Salto a la página Ejemplo3.html que estará dentro del directorio Ejemplos que estará, a su vez, dentro del directorio actual -->
  <li><a href="Ejemplos/Ejemplo3.html">Mi tercera página Web</a></li>
  <!-- Salto a la página Ejemplo4.html que estará en el directorio padre, es decir el directorio que contiene al actual -->
  <li><a href=" ../Ejemplo4.html">Mi cuarta página Web</a></li>
  <!-- Salto a la página fondo2.jpg que está en el directorio Imagenes, dentro del directorio padre -->
  <li><a href=" ../Imagenes/fondo2.jpg">Mi imagen</a></li>
</ul>
```

## Enlaces internos

Hay un tipo de enlace que permite posicionarnos en un punto concreto del documento. Este tipo de enlace se denomina interno, ya que salta dentro del propio documento.

Este salto requiere primero marcar la posición a la que deseamos saltar. Para realizar esto se utiliza el atributo **id**.

**id** es un atributo presente en cualquier elemento de la página web. Sirve para identificar a dicho elemento dentro de la página, por ello no podemos repetir valor para ese atributo en ningún otro elemento.



Supongamos que tenemos este código:

```
<h1 id="tituloPrincipal">Curso de HTML</h1>
<p>
    Este es un curso que versa sobre cómo escribir páginas web mediante...
    ...
</p>
```

El párrafo de tipo **h1** está identificado con el identificador *tituloPrincipal*. Gracias a ello si, en otra parte de la página, tenemos el código:

```
<p><a href="#tituloPrincipal"> Volver al título principal</a></p>
```

Al hacer clic en ese enlace (**Volver al título principal**), se usa el identificador como un marcador de posición y así nos colocaremos en dicho párrafo.



## Atributo target

|        |   |
|--------|---|
| target | <p>Permite indicar cómo se muestra la página de destino. Posibilidades:</p> <ul style="list-style-type: none"><li>▪ <code>_blank</code>. Abre el enlace en una nueva página. Es una opción muy utilizada al crear enlaces.</li><li>▪ <code>_parent</code>. Abre el enlace en el marco de la página padre de ella.</li><li>▪ <code>_top</code>. Abre la página en el marco superior</li><li>▪ <code>_self</code>. Abre la página en el marco actual</li><li>▪ <i>nombre</i>. EL nombre indicado será el del marco en el que se abrirá la página. Salvo el primero, el resto no se usan por referirse a marcos.</li></ul> |
|--------|---|

Los valores `_parent` y `_top` se usan cuando la página está hecha con marcos.

## Bibliografía:

<https://uniwebsidad.com/libros/xhtml/capitulo-1>

Apuntes de Jorge Sánchez