

# Estructura básica de una página

Diferenciaremos:

- **Página web:** como una página individual dentro de un sitio web (p.e: manualhtml.info/etiquetas)
- **Sitio web (o web):** como el conjunto de todas las páginas en las que podemos navegar dentro de un mismo dominio (p.e: manualhtml.info es un sitio web que incluye: manualhtml.info/etiquetas, manualhtml.info/tablas, etc).

Toda *página web* o *documento HTML* que hagamos en HTML5 debe tener una estructura inicial similar a la siguiente:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  ...
  <title>Título de la página</title>
</head>
<body>
  ...
</body>
</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 lo que se denomina **etiquetas**.

Las etiquetas sirven para delimitar elementos de la página. Los elementos son cada uno de los elementos de la misma; 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, por ejemplo).

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 < y >.

A continuación explicamos la función que cumple cada etiqueta del documento HTML anterior:

- **<!DOCTYPE html>:** indica al navegador que el código HTML en el que está escrita la página es en la versión 5, osea que es HTML5.
- **<html lang="es"> . . . </html>:** indica la raíz del documento y **todas** las etiquetas deben estar incluidas dentro. Además se especifica el idioma en el que está escrita, es = Español.
- **<head> . . . </head>:** se usa para envolver otras etiquetas que ofrecen información principalmente a: el navegador, a los buscadores y a otras páginas (como pueden ser redes sociales, etc). La información especificada dentro del *head* no se muestra *dentro* de la página web que ve el usuario.

- `<meta charset="UTF-8">`: indica al navegador qué tipo de caracteres contiene la página. Con el atributo `charset` indicamos cuál de todos los juegos de caracteres disponibles usamos. Con el valor UTF-8 podremos crear contenido en la mayoría de los sistemas de escritura: español, inglés, francés, etc.
- `<title> ... </title>`: indica el título de nuestra página. Este se muestra en: la pestaña del navegador, el enlace que indexan los buscadores, etc.
- `<body> ... </body>`: contiene todo el contenido visible por el usuario *dentro* de nuestra página.

Observa que la etiqueta *html* contiene dos hijas: *head* y *body*.

## Etiquetas

Las etiquetas solo pueden ser de dos tipos:

1) Las que tienen una apertura y un cierre como en el siguiente caso:

```
<elemento atributo="valor">Contenido</elemento>
```

Por ejemplo:

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

En este caso decimos que: "*tenemos un elemento **a** donde el valor del atributo **href** es **http://www.google.com**, y que su contenido es **Google***". No hace falta que te preocupes aún por el significado, luego haremos incapié en esto.

Si nos fijamos las etiquetas **siempre** están contenidas entre los símbolos `< >`, y el cierre solo incluye el nombre del elemento precedido de una barra lateral `/` (p.e. `</elemento>`).

2) Por otro lado están los **elementos auto-contenidos** (los que no se cierran explícitamente):

```
<elemento atributo="valor">
```

Por ejemplo:

```

```

Es importante destacar que el *atributo* y el *valor* son opcionales.

Vamos a ver algunos ejemplos de nombres de etiquetas:

- **elemento**: *html, head, meta, title, body, img...*
- **atributo**: *charset, src, alt, ...*
- **valor**: *UTF-8, "url"* (la URL a un recurso), *"texto", ...*

## Anidación de etiquetas

También es importante saber que unas etiquetas pueden contener a otras (una o varias), o como se suele decir "que se pueden anidar".

Por ejemplo:

```
<head>
  <title>Título de la página</title>
  <meta name="author" content="el nombre del autor">
</head>
```

En este caso vemos que la etiqueta **head** tiene como contenido a otra etiqueta **title**. En este caso se dice que:

- La etiqueta *head* es el **padre** de la etiqueta *title* y *meta*.
- Y que la etiqueta *title* y *meta* son **hijas** de la etiqueta *head*.
- La etiqueta *title* y *meta* son **hermanas**.

Si nos fijamos, además, la etiqueta anidada (*title*) está en una nueva línea y con un nivel de indentación/sangrado mayor. Esto es así por una convención mundial a la que se ha llegado para que los programadores escribamos código de una manera similar, tanto para hacernos más fácil y comprensible el código cuando este crezca, como para cuando tengamos que compartirlo con otros programadores.

## Orden de apertura y cierre

Cuando anidamos etiquetas unas dentro de otras es muy importante cerrarlas *en orden*. Esto quiere decir que la primera etiqueta en cerrarse tiene que ser la última que se abrió, así por ejemplo este ejemplo sería **incorrecto**:

```
<p>Texto normal <strong>texto negrita</p></strong>
```

La forma **correcta** de hacerlo sería:

```
<p>Texto normal <strong>texto negrita</strong></p>
```

Recordemos que hay etiquetas que no es necesario cerrarlas por lo que esto sería **correcto**:

```
<p>
  El orden es <strong>muy importante</strong>. <br>
  Así introducíamos un salto de línea.
</p>
```

## Atributos

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

```
<elemento nombreDeAtributo="valor1" nombreDeAtributo2="valor2" ...>
```

A los atributos se les asignan valores que deben de ir entre comillas.

Por ejemplo:

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

## Etiquetas básicas

Vamos a aprender el significado de ocho de las etiquetas que con más frecuencia tendremos que usar cuando creemos páginas web:

- `<p></p>`: representa un **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. Todo texto dentro de una etiqueta `p`, queda marcado como texto dentro de un párrafo normal.

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

- `<br>`: representa un **salto de línea**.
- `<h1></h1>`: esta etiqueta se utiliza para representar el **encabezado** de una página, como si fuera el índice de un libro. Puede variar desde 1 hasta 6 para diferenciar subniveles.
- `<hr>`: representa una **línea horizontal**.

Por ejemplo, el código:

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

Se muestra así:

Primer párrafo



Segundo párrafo

- `<ul></ul>`: representa una **lista de elementos**, donde el orden de los elementos no es importante - esto quiere decir que el cambio del orden no modifica el significado.
- `<ol></ol>`: representa una **lista de elementos**, donde el orden de los elementos sí es importante - esto quiere decir que el cambio del orden modifica el significado.
- `<li></li>`: representa un **elemento de la lista** y su padre siempre tiene que ser una etiqueta `li` o `ul`.
- `<strong></strong>`: representa **texto importante**, serio (para avisos o precauciones) o urgente (que debe ser leído antes).
- `<em></em>`: sirve para **enfaticar** en el contenido.
- `<mark></mark>`: para **remarcar** texto.

- `<sub></sub>`: permite que el **texto** aparezca **por debajo de la línea base** y en un tamaño más pequeño.
- `<sup></sup>`: permite que el **texto** aparezca **por encima de la línea base** y en un tamaño más pequeño.
- `<!-- -->`: se utiliza para añadir **comentarios** dentro del código que el usuario no podrá ver. Por ejemplo para añadir notas de tareas pendientes, aclaraciones que nos ayuden a nosotros o a otras personas a entender el código, etc.

**Truco:** Para que recuerdes mejor qué significa cada elemento, las etiquetas piensa en los acrónimos en inglés:

- `h1` = **h**eading**1**; `h2` = **h**eading**2**; ...
- `p` = **p**aragraph
- `br` = **b**reak **l**ine
- `ul` = **u**nordered **l**ist
- `ol` = **o**rdered **l**ist
- `li` = **l**ist **i**tem
- `em` = **e**mphasis

El siguiente ejemplo muestra una página web que combina todas ellas:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Ejemplo con etiquetas básicas</title>
</head>
<body>
  <h1>Etiquetas HTML</h1>

  <p>
    Este ejemplo muestra cómo combinar algunas de las etiquetas más básicas
    de HTML5. <br>
    Recuerda que <strong>es importante entender la diferencias entre
    ellas</strong>.
  </p>

  <h2>Etiqueta ul+li</h2>
  <p>
    Si listamos personas nominadas a los Oscars, dado que el orden no altera
    el significado, debemos usar <em>ul</em>.
  </p>
  <ul>
    <li>David Verdaguer</li>
    <li>Jesús Castro</li>
    <li>Israel Elejalde</li>
    <li>Dani Rovira</li>
  </ul>

  <h2>Etiqueta ol+li</h2>
```

```

<p>
  En el caso de que estemos listando elementos donde el orden es
  importante, como por ejemplo la clasificación de un mundial de fútbol, debemos
  usar <em>ol</em>.
</p>
<ol>
  <li>España</li>
  <li>Países Bajos</li>
  <li>Alemania</li>
  <li>Uruguay</li>
</ol>
</body>
</html>

```

### Actividad:

Comprueba como se visualiza en el navegador.

Es importante destacar que ***el navegador le añade un estilo (CSS) por defecto a las etiquetas***; por ejemplo:

- **h1** y **h2** una fuente mayor y negrita
- **strong** en negrita
- **ul** y **li** con un margen a la izquierda y un punto o número respectivamente
- **em** en cursiva

Esto no es responsabilidad del HTML, esto lo podremos personalizar en un futuro con CSS.

Así que recuerda que HTML **sólo sirve para dotar de estructura y semántica al contenido**.

Este valor semántico es **muy importante** entre otras cosas para que, por ejemplo:

- Los buscadores (que no son más que programas automatizados) puedan "entender" el contenido de nuestra página y así poder detectar de qué estamos hablando y qué es importante.
- Otras herramientas, como por ejemplo los navegadores para invidentes, puedan leer el contenido al usuario.

### Nota:

Como el nombre de las etiquetas va entre el signo “menor que” y el signo “mayor que”, si queremos que aparezcan estos signos como parte de un texto debemos de utilizar **&lt;** y **&gt;** respectivamente.

Ejemplo:

2 < 5 => 2 &lt; 5

5 > 2 => 5 &gt; 2

# Espacios en blanco

El texto dentro de las páginas web no respeta los espacios en blanco ni tabuladores que coloquemos en el código, a la hora de mostrar el contenido por pantalla. Solo se considera el primer espacio en blanco, el resto se eliminan.

Por ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Título de la página</title>
</head>
<body>
  Este texto      tiene      muchos
                  espacios
</body>
</html>
```

El resultado mostraría en la página este texto:

Este texto tiene muchos espacios

La razón es permitir una mayor legibilidad del código, permitiendo en él que coloquemos el código con sangrías y espacios que mejoren su lectura, para que el mantenimiento del mismo sea más cómodo.

## Etiqueta pre

Si deseamos que los espacios que colocamos en el código sí sean tenidos en cuenta, podemos utilizar la etiqueta **pre**.

Realmente, con carácter general, no se aconseja el uso de **pre** porque ayuda a adquirir malos hábitos y dificulta el aprendizaje de las formas avanzadas de maquetar páginas web.

Sí hay contextos en el que es obligado su uso, por ejemplo para mostrar textos referidos a ejemplos de código de lenguajes de programación.

Ejemplo:

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <title>Hola mundo</title>

</head>
```

```
<body>

    <p>Programa <strong>Hola mundo</strong>

    escrito en Java

<pre>

public class HolaMundo{

    public string void main(String[] args){

        System.out.println("Hola Mundo");

    }

}

</pre>

</body>

</html>
```

Como la página anterior muestra un ejemplo de código en Java, lo más apropiado es usar **pre** para mantener las tabulaciones tal cual se hicieron al escribir el código. El resultado, sería:

Programa **Hola mundo** escrito en Java

```
public class HolaMundo{
    public string void main(String[] args){
        System.out.println("Hola Mundo");
    }
}
```

## Uso de &nbsp;

Otra forma de obligar a tener en cuenta espacios en blanco es el código **&nbsp;**. Dicho código es una entidad (las entidades se explican más adelante). Si, por ejemplo, escribimos cuatro veces seguidas *&nbsp;*; estaremos dejando cuatro espacios en blanco que el navegador sí tendrá en cuenta.

Ejemplo:

texto &nbsp; &nbsp; &nbsp; &nbsp; otro texto

El segundo texto aparecerá cuatro espacios en blanco detrás del primero.



# Contenido y contenedores

En HTML los elementos que muestran algo en la página, se colocan dentro de la sección **body**.

La mayoría de elementos en HTML son contenedores de otros textos y elementos. Hay que conocer que tipos de elementos tenemos. Los tipos están relacionados con la propiedad CSS **display**, que se encarga de decir cómo tiene que mostrarse un elemento. Por defecto los elementos funcionan de esta forma:

- Elementos de tipo **inline**. Son elementos que se muestran seguidos en la misma línea. Los elementos de este tipo son los que marcan texto simple. Todas las etiquetas de marcado de texto tienen este display: strong, em, span, etc.

Por ejemplo, el código:

```
<strong>Hola</strong> <em>Hola</em>
```

Se muestra así:

**Hola** *Hola*

- Elementos de tipo **block**. Son contenedores más grandes. Los elementos que marcan párrafos enteros (como p, h1, h2, ...) y los contenedores de secciones (como div, section, article, ...) tienen este display. Su característica es que, por defecto, tienen forma rectangular y abarcan toda la anchura de su contenedor. Si indicamos para ellos un color de fondo, este se muestra en forma de rectángulo (en los inline, el color de fondo abarca exactamente al texto, como si lo hubiéramos pintado con un rotulador).

Por ejemplo, el código:

```
<h1>Hola</h1> <p>Hola</p>
```

Se muestra así:

**Hola**

Hola

- Elementos de tipo **inline-block**. Son rectangulares como los block, pero no abarcan todo su contenedor. Se adaptan a su contenido. Ejemplos son los elementos button o image.

Por ejemplo, el código:

```
<strong>Hola</strong> <button>Hola</button>
```

El resultado muestra el texto tipo strong seguido de un botón (elemento rectangular). Si el botón tuviera un display de tipo block, saltaría a la línea siguiente.



En todo caso, el **display** de un elemento se puede modificar desde lenguaje CSS, pero no conviene hacer combinaciones anti naturales ya que se pierde el sentido semántico del código.