

# Estructura de un documento HTML/xHTML

---



## 1. Introducción a HTML

---

Una página web no es más que un archivo de texto (que podemos hacer con el mismo bloc de notas) donde, mediante una serie de **elementos de marcado**, configuramos la estructura y el contenido que va a tener dicha página. En otras palabras, el texto en una página web está *marcado* con estas etiquetas para dar instrucciones al navegador web sobre cómo mostrar dicho texto.

Estas etiquetas forman un lenguaje llamado **HTML** (*HyperText Markup Language*), que es el lenguaje con el que se escriben la mayoría de páginas web. Es un lenguaje estándar, reconocido en todo el mundo, y definido por el **W3C** (*World Wide Web Consortium*), una entidad sin ánimo de lucro que desarrolla estándares que aseguran el crecimiento de la Web a largo plazo.

### 1.1. Breve historia

El origen de HTML se remonta a los años 80, de la mano de Tim Berners-Lee, investigador del CERN que quería proponer un sistema para compartir y enlazar documentos a través del *hipertexto*, un concepto que aludía a la posibilidad de ir a un documento desde otro. El primer documento HTML se publicó en 1991 bajo el nombre *HTML Tags* (etiquetas HTML), y la primera propuesta de estándar oficial salió en 1993, aunque no se consiguió.

En 1995 se publicó HTML 2.0 que, a pesar de su nombre, es el primer estándar oficial de HTML. A partir de 1996, los estándares de HTML los publica el W3C. De este modo, HTML 3.2 se publicó en 1997 y HTML 4.0 en 1998. Ésta incorporó las hojas de estilos CSS y la posibilidad de incluir *scripts* en páginas web. Ambas cosas supusieron un gran avance en el desarrollo web. La última especificación oficial de HTML se publicó en 1999 con el nombre de HTML 4.01, sin novedades significativas respecto a la anterior.

Después, la actividad en torno a HTML se ralentizó, y el W3C se centró más en desarrollar otro estándar alternativo, el xHTML. Esto hizo que algunas empresas como Apple o Mozilla se organizaran en una asociación, llamada WHATWG (*Web Hypertext Application Technology Working Group*), centrada en el estándar HTML5. El primer borrador se publicó en 2008, ya con la colaboración del W3C, y a partir de ahí, diferentes módulos han ido aprobándose como estándares independientes, hasta alcanzar la versión final en 2014. La intención es construir una plataforma web en la que se puedan construir aplicaciones multiplataforma con HTML5, CSS3 y JavaScript. En cualquier caso, uno de los grandes aportes de HTML 5 al desarrollo de páginas web es la inclusión de nuevas etiquetas, que aportan una estructura semántica de la página, donde podemos identificar secciones, cabeceras, pies, artículos y otros elementos.

### 1.2 Elementos HTML: etiquetas y atributos.

Los elementos con que marcamos los documentos HTML se denominan **etiquetas**. Suelen dividirse en tres partes:

- **Etiqueta de apertura:** se usa para indicar dónde empieza un elemento; se escribe entre los símbolos de "menor que" (<) y "mayor que" (>). Por ejemplo, la etiqueta `<p>` indica el inicio de un párrafo.
- **Contenido:** es el texto que mostrará el navegador al usuario.
- **Etiqueta de cierre:** especifica dónde finaliza la acción de un elemento. Se escribe igual que la etiqueta de apertura, pero poniendo una barra inclinada después del símbolo "menor que" (<). La etiqueta `</p>` indica el final de un párrafo.

```
<etiqueta> texto </etiqueta>
```

Aunque la mayoría de elementos tienen etiqueta de apertura y de cierre, no todos tienen etiqueta de cierre. Es el caso de los **elementos vacíos**. Al contrario que el resto de elementos, éstos no pueden contener otros elementos. Algunos ejemplos de elementos vacíos son las etiquetas `<br>`, `<hr>` o `<img>` que trataremos más adelante.

## Atributos

Por otra parte, muchas etiquetas HTML tienen **atributos** que sirven para añadir ciertos detalles, por ejemplo, el ancho y alto del elemento, la ubicación de una imagen, el color del texto, etc. Los atributos se escriben dentro de la etiqueta de la siguiente forma:

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

Por ejemplo, para añadir una imagen se utiliza el elemento `img`. Este elemento no utiliza etiqueta de cierre pero sí necesita utilizar un atributo para indicar la ubicación de la imagen:

```

```

Para algunas etiquetas, como la del ejemplo, es obligatorio añadir el atributo, sin embargo, para otras puede ser opcional. Una etiqueta puede tener varios atributos, de los cuales, algunos serán obligatorios y otros opcionales.

Lo importante por ahora es saber que los contenidos se organizan en etiquetas, y que (normalmente) éstas van por parejas, una que inicia una sección, y otra que la cierra (precedida por una barra /). También es importante recalcar que las etiquetas van siempre en minúscula (*html*, *head*, *body*).

## 2. Estructura básica de un documento HTML

En general, cualquier página web escrita en HTML debe almacenarse en un archivo con extensión *.html* o, en su defecto, *.htm*. El contenido es procesado después por un *navegador*, software encargado de mostrar el contenido de las diferentes páginas web.

El documento HTML tiene una estructura básica similar a la siguiente:

```
<!DOCTYPE ... >
<html>
  <head>
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

En esta estructura podemos diferenciar los siguientes apartados:

- Lo primero que suele aparecer en un documento HTML es la declaración del tipo de documento o **doctype**; sirve para indicar al navegador qué tipo de documento está viendo y qué versión específica del lenguaje se está empleando. Veremos información más detallada sobre esto en el siguiente punto.
- Toda la estructura y contenido del documento se engloba dentro del par de etiquetas `<html>` y `</html>`. La etiqueta `<html>` representa la raíz de un documento HTML y contiene todos los demás elementos HTML (excepto la etiqueta `<!DOCTYPE>`).

Es recomendable incluir el atributo `lang` dentro de la etiqueta `<html>` y seguido del código del idioma para declarar el idioma de la página web. Esto facilita la tarea de los motores de búsqueda y navegadores a la hora de posicionar el documento. Algunos ejemplos de código de idioma son "en" para inglés, "es" para español y "fr" para francés y vienen detallados en la norma ISO 639.

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

- La sección englobada entre las etiquetas `<head>` y `</head>` se denomina **cabecera**, y en ella se incluye información sobre la propia página o *metadatos*, pero no su estructura y contenido. Se trata de información que no se visualiza pero que resulta útil a los navegadores. Analizaremos estos elementos más adelante.
- La sección entre las etiquetas `<body>` y `</body>` se denomina **cuerpo**; define la estructura y contenido de la página (lo que vemos en la ventana del navegador). Dentro de esta sección podemos utilizar una serie de etiquetas para definir ciertos tipos de contenidos, como por ejemplo encabezados,

párrafos, listas, tablas, imágenes, etc. Iremos familiarizándonos con estos elementos en sucesivos apartados.

## 2.1 El *doctype*

Como hemos comentado, es lo primero que suele aparecer en un documento HTML y sirve para indicar al navegador qué versión específica del lenguaje se está empleando.

En el pasado, muchos navegadores no mostraban correctamente el contenido cuando no se indicaba la **definición de tipo de documento (DTD, Document Type Definition)** o ésta era incorrecta. Una DTD es un documento SGML que incluye las reglas sintácticas para un tipo de documento específico. Incluye los elementos que se permiten y sus atributos, así como reglas que afectan a la anidación de los primeros y a los valores de los segundos. Contrastando un documento con su DTD se puede comprobar si éste es válido o no. Éste es el aspecto que presenta, por ejemplo, la [DTD de XHTML 1.0 \(Estricto\)](#).

La declaración del tipo de documento en algunas de las distintas versiones de HTML o XHTML quedaría así:

### HTML 4.01:

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

### XHTML 1.1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

### HTML 5:

A partir de HTML 5 la declaración es muy sencilla, ya que no requiere la especificación de una DTD para mostrar la web correctamente, sino más bien, para validarlo.

```
<!DOCTYPE html>
```

Más adelante veremos más en detalle cómo comprobar la validez de los documentos.

## 2.2 Elementos de la cabecera

Los elementos que se encuentran en la sección `head` incluyen:

- El título de la página web (`title`)

- Las metaetiquetas (*metatags*) que describen el documento, como la codificación de caracteres utilizada y la información a la que pueden acceder los motores de búsqueda
- Las referencias a archivos externos que necesita la página, tales como scripts y hojas de estilos ( `style` , `script` , `base` y `link` ). Estos elementos los veremos más adelante en el curso, a medida que los necesitemos.

### 2.2.1 El elemento title

El elemento `title` es el título de la página web; se escribe entre las etiquetas `<title>` y `</title>` y se muestra en la barra de título de la ventana del navegador. Aunque el texto puede ser de cualquier longitud, si sobrepasa los 60 caracteres puede que el navegador lo corte cuando lo muestre.

### 2.2.2 Las metaetiquetas

Las **metaetiquetas** siempre van dentro del elemento `head` , y se utilizan para indicar el juego de caracteres empleado, la descripción de la página, las palabras clave, el autor del documento y la configuración del área visible de una página (*viewport*), que será diferente para cada tipo de dispositivo.

Las metaetiquetas utilizan dos atributos: el atributo `name` , para indicar el nombre de la metaetiqueta, y el atributo `content` , para indicar su contenido, excepto para la metaetiqueta que indica la codificación de caracteres, que sólo utiliza el atributo `charset` .

```
<head>
  <meta charset="utf-8">
  <meta name="description"
        content="Apuntes de Lenguaje de Marcas: HTML, CSS, Javascript">
  <meta name="keywords" content="HTML, CSS, Javascript">
  <meta name="author" content="May Calle">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

Vamos a analizar el ejemplo anterior:

#### Metaetiqueta *charset*

Define la codificación de caracteres para el documento. Una codificación define cómo se traducen los bytes a texto y viceversa. Eligiendo una codificación en particular, decimos cómo debe ser interpretada la secuencia de bytes.

La codificación predeterminada en HTML5 es UTF-8, y cubre (casi) todos los caracteres, signos de puntuación y símbolos del mundo. Si una página web HTML5 utiliza un juego de caracteres diferente al UTF-8, debe especificarse mediante esta meta etiqueta. Por ejemplo, para una web en español se puede utilizar la codificación ISO-8859-1, que define la codificación para Europa occidental:

```
<head>
  <meta charset="iso-8859-1">
</head>
```

Es habitual incluir esta meta etiqueta dentro de los primeros 512 caracteres del documento, para asegurar que todos los navegadores la lean correctamente. Por ejemplo, puede ser la primera etiqueta dentro del bloque `head`, y debemos tener presente que sólo puede haber una (una página HTML sólo puede especificar una única codificación).

### Metaetiqueta *description*

Se utiliza para describir brevemente el contenido de una página web. Esta descripción aparece debajo del título y la URL de la página en los resultados del motor de búsqueda. Para permanecer visible dentro de Google, tiene que contener entre 140-160 caracteres. Además, para favorecer un buen posicionamiento es recomendable que describa el contenido de la página de forma significativa y que incluya alguna de sus palabras clave.

### Metaetiqueta *keywords*

Define una serie de palabras clave relacionadas con los contenidos de la página web que pueden resultar útiles para los motores de búsqueda, aunque su uso tiene cada vez menor influencia en el posicionamiento. Hoy en día es ignorado por Google y su importancia ha sido minimizada por Bing y otros buscadores populares.

### Metaetiqueta *author*

Indica el nombre del autor de la página.

### Metaetiqueta *viewport*

Configura el área visible de la página web para se visualice correctamente en todos los dispositivos.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Al añadir el anterior código en una página web le estamos indicando al navegador la siguiente información sobre cómo controlar las dimensiones y la escala de la página:

- `width=device-width` ajusta el ancho de la página al ancho de la pantalla del dispositivo con la que se está visualizando (es la opción más recomendable), pero si tu página está específicamente creada para móvil a 320px puedes indicar esa medida directamente con `width=320p`.
- `initial-scale=1` establece el nivel de zoom inicial cuando el navegador carga la página por primera vez. Al poner 1 no aplicamos ningún zoom. Podemos ir más allá y prevenir cualquier zoom por parte del usuario con `maximun-scale=1`.

## 2.3. Un ejemplo sencillo

A continuación se muestra el código HTML de una página web sencilla:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Mi página</title>
    <meta charset="utf-8">
    <meta name="author" content="May Calle">
  </head>
  <body>
    <p>Una página sencilla de prueba</p>
  </body>
</html>
```

## 2.4 Comentarios en HTML

HTML permite añadir comentarios en el código. Estos comentarios pueden servir para dos cosas: explicar una parte del código (en este caso, de la estructura o contenidos de la página web) que no quede clara, o para delimitar/marcar una zona del código para poder localizarla o identificarla más fácilmente.

Los comentarios en HTML comienzan con el símbolo `<!--` y terminan con el símbolo `-->`. Lo que haya entre estos símbolos no se muestra en el contenido de la página, sólo se puede leer mirando el código fuente. Veamos un ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <!-- Añadimos el juego de caracteres -->
    <meta charset="utf-8">
    <!-- Título de la página -->
    <title>Bienvenido/a</title>
  </head>
  <body>
    <!-- En este ejemplo, simplemente mostraremos un texto de saludo -->
    <p>Hola, buenos días</p>.
  </body>
</html>
```

### Ejercicio 1:

Crea con VS Code un nuevo documento HTML, y llámalo **pagina1.html**. Dentro, añade una estructura básica de etiquetas HTML para que el documento tenga lo siguiente:

- Un comentario al principio del documento con tu nombre
- Un título que sea "Mi primera página HTML"
- Un juego de caracteres ISO-8859-1 o UTF-8.
- Información sobre el autor.
- Configura el área visible de la página web para se visualice correctamente en todos los dispositivos.
- Un párrafo (en el *body*) con tu nombre.

Guarda la página y prueba a abrirla con un navegador para comprobar que se ve correctamente (incluyendo los acentos que pueda haber).

## Ejercicio 2:

Dado el siguiente documento HTML, indica qué cosas faltan o sobran para que sea correcto.

```
<html>
  <title>Mi página</title>
  <head>
    <meta name="charset" content="utf-8">
    <meta name="author" content="May Calle">
  </head>
  <p>Una página sencilla de prueba</p>
</html>
```

## 3. Documentos xHTML

Los documentos xHTML (*eXtended HyperText Markup Language*) son una variante de los documentos HTML asociada a XML, o dicho de otro modo, son documentos HTML expresados como XML válido. Para escribir documentos en formato xHTML, debemos añadir un par de cambios a la estructura base de documento que hemos visto antes.

El primero de ellos es que debemos etiquetar el documento como un documento XML, para lo que debe comenzar con esta línea:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Estamos indicando que usamos la versión 1.0 de XML (la más actualizada), y la codificación del documento en general (UTF-8).

Después, la línea de *DOCTYPE* es diferente a HTML5, y en ella debemos indicar el documento DTD que se empleará para comprobar la validez del documento, como hemos explicado anteriormente:

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

Finalmente, la etiqueta raíz `html` debe contener información sobre el espacio de nombres (*namespace*, abreviado *ns*) que se utilizará, es decir, el conjunto de etiquetas válidas que podemos emplear en el documento. Además, especificamos el idioma de la página en el atributo `lang` (como ocurre en HTML) y también en el atributo `xml:lang`:

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

El resto del documento, a grandes rasgos, es igual, pero teniendo siempre en cuenta que las etiquetas permitidas por XHTML no coinciden con las que permite HTML5 (las puedes consultar [aquí](#)). En resumen, dependiendo del lenguaje de marcado que utilicemos para definir nuestras páginas web, y de su versión, deberemos atenernos a un conjunto de reglas u otro para que estos documentos sean correctos.