

HTML5

Introducción a HTML5



One Web **W3C** *for All*

El documento que lee el navegador está escrito en un lenguaje de marcado llamado **HTML**, que son las siglas de **HyperText Markup Language** (Lenguaje de marcas de hipertexto), o lo que es lo mismo, un lenguaje de etiquetas que permite incluir o hacer referencia a todo tipo de información.

Dicho documento esta formado por **etiquetas**, que son la base del lenguaje HTML. Existen muchas etiquetas y cada una se utiliza para contener información y darle un cierto significado a dicha información.

```
<!-- INCORRECTO -->
<etiqueta>contenido</etiqueta>

<!-- Correcto -->
<p>Párrafo</p>
```

En **HTML** no se puede utilizar cualquier palabra como etiqueta (en el ejemplo anterior, es incorrecto utilizar la etiqueta «etiqueta»). En su lugar, existen una serie de etiquetas concretas, cada una de ellas con su finalidad y características propias, que tendremos que utilizar según requiera la ocasión. Por norma general, las etiquetas deben cerrarse para indicar donde finaliza su contenido.

En los navegadores, tienes varias formas de acceder al código HTML de la página:

- Pulsando la combinación de teclas `Ctrl + U` . Te aparecerá el código fuente tal cuál lo recibe el navegador.
- Pulsando `Ctrl + Shift + I` aparecerá la consola del navegador.

Uno de los principales objetivos de HTML5 es introducir información en un documento HTML5 de forma que sea **semántico** y no visual. Con esto queremos decir que todos los aspectos visuales deben dejarse para el apartado de presentación, que se gestiona desde el lenguaje CSS.

En el documento HTML debe aparecer información correctamente individualizada, de modo que al leer una página HTML comprendamos su significado, y si queremos cambiar la apariencia, lo hagamos en el documento CSS. Esto es lo que comunmente se conoce como **separación de la presentación del contenido**.

```
Hola, quiero resaltar esta <b>palabra</b>.
```

En este caso se está utilizando la etiqueta `` de HTML4 y anteriores para poner en negrita un texto. Se está utilizando una propiedad de presentación (visual) en el HTML, algo que no se debe hacer en HTML5. La misión de HTML5 es mantener sólo contenido e información semántica en HTML5. Por dicha razón, la forma de hacerlo en HTML5 es utilizando una etiqueta que añade esa información semántica como es ``:

```
Hola, quiero resaltar esta <strong>palabra</strong>.
```

El objetivo de crear documentos HTML semánticos es que, aunque estamos acostumbrados a crear páginas para usuarios (o más concretamente, para navegadores), cada vez tendemos más a una Internet capaz de procesar información de forma autónoma.

Estructura de etiqueta HTML

La estructura de las etiquetas HTML tiene el siguiente formato:

```
<etiqueta atributo="valor">contenido</etiqueta>
```

Etiqueta HTML

La parte esencial de una etiqueta HTML es lo que se denomina la **etiqueta de apertura**. Se trata de escribir el nombre de la etiqueta en cuestión, colocándola entre los caracteres `<` y `>`. Aunque el navegador lo permita si no es así, las etiquetas HTML se deberían escribir siempre en minúsculas.

En HTML5 no se puede colocar cualquier palabra como etiqueta.

La mayoría de las etiquetas requieren que se especifique un **cierre de etiqueta** para saber donde termina de actuar. Se caracteriza en que se escribe igual que la etiqueta de apertura, pero con la barra `/` inmediatamente después del `<`.

Atributos HTML

En algunas etiquetas HTML, existen algunos atributos específicos (que pueden ser **opcionales u obligatorios**). Los atributos determinan cierta información sobre la etiqueta y generalmente van asociados a un valor determinado. Este par atributo-valor se escribe después del nombre de la etiqueta, separándola por espacio y antes del carácter `>` de la etiqueta de apertura:

```
<strong id="dato">Contenido</strong>

<strong id="dato" class="clase1" lang="es">Contenido de texto</strong>
```

Una etiqueta puede tener varios pares *atributo-valor*, como se ve en el ejemplo anterior, pero nunca se debe repetir el mismo atributo en una misma etiqueta varias veces, ya que **sobrecribiría** al anterior. El orden de los atributos no importa.

Aunque los valores pueden ir rodeados por comillas simples, se recomienda escribir el valor siempre entre **comillas dobles**.

Existen 3 tipos de atributos dependiendo de sus valores:

1. **Conjunto de valores**: Son aquellos atributos en los que sólo se permiten unos valores concretos. Cualquier otro valor diferente, no será válido.
2. **Valores libres**: Son los atributos en los que puedes especificar un valor libremente, como una dirección URL o un texto, y no existe una serie de valores específicos para escribir.
3. **Valores booleanos**: Son los atributos que deben tener un valor verdadero (true) o un valor falso (false). En HTML5 un atributo sin valor (solo el atributo) significa verdadero y si se omite el atributo se considera falso.

Contenido de la etiqueta HTML

En el interior de la etiqueta HTML (después de la etiqueta de apertura y antes de la etiqueta de cierre) se debe colocar la información que queremos que sea afectada por dicha etiqueta. Una etiqueta puede contener desde un fragmento de texto hasta un grupo de etiquetas.

```
<div id="pagina">
  <!-- Bloque de etiquetas -->
  <strong>Contenido importante</strong>
</div>
```

Comentario HTML

Para introducir estos comentarios en el código HTML, basta con escribir los fragmentos de texto `<!-- y -->` entre el comentario en cuestión que queramos incluir.

 **NOTA**: Los comentarios HTML son **visibles** si se accede al código fuente del documento.

Atributos comunes en HTML

Los atributos son palabras clave de texto que modifican ligeramente el comportamiento de la etiqueta que lo contiene. Los atributos comunes son atributos que pueden estar prácticamente en **cualquier etiqueta HTML**.

Atributos CSS

- **id**: Establece un identificador único a la etiqueta HTML. Sólo el mismo nombre una vez por documento.
- **class**: Establece una clase (género) a una etiqueta HTML. Puede repetirse por documento.
- **style**: Aplica propiedades CSS directamente al elemento HTML en cuestión.

El atributo *id* (identificador)

En HTML, podemos darle un identificador a una etiqueta HTML y de esta forma darle un nombre. Simplemente, añadimos el atributo `id` y colocamos el nombre como valor de ese atributo.

```
<div id="header">
  <div>Aquí irá un anuncio</div>
  <div id="articulo">Aquí irá el contenido de texto del artículo</div>
  <div>Aquí irá un anuncio</div>
</div>
```

Ese identificador único tiene unas normas:

- No debe empezar nunca por un número, pero puede contenerlos más adelante.
- El texto debe estar en *kebab-case*, es decir, minúsculas y separados por un guión.
- Es preferible que no contenga caracteres raros, acentuados o emojis.
- En un documento HTML no pueden existir dos elementos con el mismo id.

Los identificadores se suele utilizar para zonas específicas que no se van a repetir, como **header** o **footer**.

La recomendación es utilizar clases en vez de identificadores **siempre que sea posible**.

El atributo *class* (clase)

Las clases funcionan de una forma muy similar a los identificadores, pero son mucho más flexibles ya que no tiene la limitación de ser únicos. La idea de las clases es establecer géneros o tipos de etiquetas, a los que les asociemos características comunes.

```
<div id="pagina">
  <div class="anuncio">Aquí irá un anuncio</div>
  <div id="articulo">Aquí irá el contenido de texto del artículo</div>
  <div class="anuncio ultimo">Aquí irá un anuncio</div>
</div>
```

La ventaja de utilizar clases es que mediante CSS se puede aplicar un estilo concreto a todas las etiquetas HTML que tengan esa clase.

Además, a diferencia de los identificadores, una etiqueta puede tener múltiples clases diferentes separadas por un espacio. Si se indican múltiples atributos `class` en la misma etiqueta, como ya se ha indicado el navegador ignorará los primeros y sólo utilizará el valor del último `class`.

El atributo *style* (estilos en línea)

El atributo `style` es un atributo que se utiliza en las etiquetas HTML para incrustar código CSS directamente en la propia etiqueta.

```
<p style="background: indigo; color: white;">Esto es un mensaje con estilos CSS</p>
```

En la mayoría de los casos, no se recomienda añadir los estilos de esta forma ya que suele ser considerado una mala práctica. Es mejor colocar el código CSS en un documento .css separado.

Atributos de idioma

- **lang**: Indica el idioma del contenido de la etiqueta HTML.
- **translate**: Indica si el contenido de la etiqueta se debería traducir o no.
- **dir**: Establece la direccionalidad del texto.

El atributo *lang* (idioma)

Mediante el atributo lang podemos indicar el **idioma** del contenido de la etiqueta. El valor de dicho atributo tendrá que ser el [código ISO 639-1](#) del idioma al que queremos hacer referencia.

Aunque en principio podemos hacer esto en cualquier etiqueta HTML, es **obligatorio** hacerlo en la etiqueta `<html>` que es la etiqueta que abarca todo el documento, estableciendo el idioma del mismo:

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

```
...
</html>
```

El atributo *translate* (traducción)

En las etiquetas HTML se puede indicar el atributo `translate`, el cuál acepta los valores 'yes' y 'no'. Por defecto, aunque este atributo no sea añadido en una etiqueta, el valor por **defecto** que tiene es 'yes'. Por lo tanto, todas las etiquetas están marcadas como «traducibles».

```
<p>
  Hace algunos días fui a ver la nueva película de <span translate="no">StarWars</span>.
</p>
```

Por ejemplo, mediante este atributo podemos indicar a herramientas como 'Google Translate' que no traduzcan una etiqueta que puede estar indicando el título de un libro, que debe conservarse tal cual.

El atributo *dir* (direccionalidad)

Existe un atributo `dir` que permite al desarrollador indicar la **direccionalidad** del texto en el documento. Esto es ideal para idiomas en los que se escribe o lee de derecha a izquierda, en lugar de izquierda a derecha.

El valor por **defecto** de este atributo es 'ltr' (left to right, de izquierda a derecha), pero podemos modificarlo y establecer el valor 'rtl' (right to left, de derecha a izquierda).

Otros atributos comunes

- **title**: Mensaje mostrado en un tooltip (aviso emergente) al mover el ratón encima.
- **data-***: Metadatos en la propia etiqueta. Se puede usar cualquier nombre con prefijo 'data-'.
- **accesskey**: Combinación de teclas que puede pulsar el usuario para activar el elemento.

El atributo *title* (título)

Aunque se suele hacer sobre todo en las etiquetas de imágenes ``, en la mayoría de las etiquetas HTML podemos indicar el atributo `title` para especificar un mensaje de texto que aparezca cuando el usuario detenga el ratón sobre el elemento un instante.

```
<figure>
  
  <figcaption title="Pie de foto">One Web W3C for All</figcaption>
</figure>
```

Aunque los atributos `alt` y `title` en las etiquetas `` tienen objetivos parecidos:

- El atributo `alt` debe ser un texto alternativo que describa la imagen (en el caso de que no se pueda ver visualmente)
- El atributo `title` puede describir la imagen, pero no tiene porque ser una descripción alternativa.

El atributo *data-* (metadatos)

En un documento HTML, la mayoría de los **metadatos** (información adicional) se incluyen en el interior de la etiqueta `<head>` del documento HTML. Sin embargo, también se pueden incluir metadatos en una etiqueta HTML a través de un atributo que comienza con el prefijo `data-`.

Los [atributos de datos personalizados](#) están destinados a almacenar datos personalizados, estado, anotaciones y similares, privados a la página o aplicación, para los que no hay atributos o elementos más apropiados.

Desde Javascript, si queremos hacer referencia a estos elementos, podemos hacerlo de varias formas:

```
<!-- HTML5 -->
<div class="spaceship" data-ship-id="92432"
    data-weapons="laser 2" data-shields="50%"
    data-x="30" data-y="10" data-z="90">
</div>
```

```
// JavaScript
const spaceship = document.getElementsByClassName("spaceship");

console.log(spaceship[0].dataset.weapons) // "Laser 2"
console.log(spaceship[0].getAttribute("data-shields")); // "50%"
```

En la primera, accedemos a la propiedad `.dataset` donde tendremos una lista de propiedades dependiendo de los diferentes atributos con prefijo `data-` que tenga el elemento. Por otro lado, podemos también utilizar el método `.getAttribute()` del DOM.

El atributo `accesskey` (atajo)

En HTML es posible añadir el atributo `accesskey` para indicar un atajo de teclado que puede pulsar el usuario para activar ese elemento. De esta forma, al pulsar la combinación `Alt + <tecla>` se activa ese elemento:

```
<form>
  <input accesskey="N" placeholder="Nombre (ALT+N)" />      <!-- Campo de datos -->
  <input accesskey="A" placeholder="Apellidos (ALT+A)" />    <!-- Campo de datos -->
</form>
```

⚠ Debido a que no está estandarizado entre sistemas operativos y navegadores no se aconseja su uso.

Estructura del documento HTML

Un **documento HTML** debe estar bien formado sintácticamente para que el navegador pueda leerlo correctamente. Para ello, debe tener una estructura inicial bien definida, con ciertas etiquetas HTML obligatorias y algunas características recomendables. En principio, necesitaremos diferenciar la estructura del documento en **tres partes**.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Título del documento</title>
  <meta charset="utf8">
</head>
<body>
  <!-- ... -->
</body>
</html>
```

El **DOCTYPE** (tipo de documento)

El `!DOCTYPE` o **tipo de documento** es una etiqueta especial que se escribe en la primera línea del documento HTML. Debe ir especificado siempre para que el navegador sepa de que tipo de documento HTML se trata.

Es una etiqueta **obligatoria**, pero de no indicarla, la página web probablemente continuaría visualizándose de forma correcta. En el caso de no indicar el tipo de documento en una página HTML, el navegador entra en lo que se llama '**Quirk mode**' (modo peculiar o modo no estándar), donde se activa un modo de retrocompatibilidad con páginas antiguas, por lo que procesará de forma diferente muchas etiquetas HTML o propiedades CSS.

Para los documentos **HTML5** tiene esta forma:

```
<!DOCTYPE html>
```

En versiones anteriores, como **HTML4** o **XHTML**, el tipo de documento se especificaba en la primera línea de una forma más compleja:

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

La etiqueta *head* (metadatos)

Cuando hablamos de la etiqueta `<head>` de un documento HTML, muchas veces la denominamos **cabecera HTML**. Sin embargo, esta cabecera no es una parte visual de la página web, sino una parte de nuestro código HTML donde se incluyen ciertas etiquetas de metadatos, es decir, unas etiquetas que establecen ciertos datos que no tienen que verse necesariamente de forma visual.

Algunos metadatos son el **título**, la **descripción** de la página, el **icono** o favicon, etcétera...

La etiqueta *body* (cuerpo)

La otra parte principal de un documento HTML es la etiqueta `<body>`. Todos los elementos visuales de una página se encuentran en el interior de la etiqueta `<body>`, por lo que es una de las partes más importantes de una página web. Esta sección va inmediatamente después del cierre de la etiqueta `</head>`.

Validación de errores en HTML

En los lenguajes de marcas como HTML, los navegadores son más permisivos, ya que en el caso de encontrar un error, intentan «deducir» lo que realmente se quería indicar y continúan con la carga del documento.

En nuestro código HTML podemos tener varios tipos de problemas:

- **Sintaxis:** El código está mal escrito y es incorrecto. El navegador podría no mostrar correctamente ciertos detalles.
- **Accesibilidad:** El código no tiene porque estar mal escrito o ser incorrecto, sin embargo, tiene problemas que harán que no se vean bien en determinados dispositivos o que un usuario invidente o similar no pueda utilizarla.
- **Posicionamiento (SEO):** El código no tiene porque estar mal escrito o ser incorrecto, sin embargo, los buscadores como Google no aceptarán favorablemente la página y podría tener un rendimiento menor de posicionamiento SEO.
- **Rendimiento:** El código no tiene porque estar mal escrito o ser incorrecto, sin embargo, el navegador puede ser lento a la hora de cargarlo o tardar en realizar sus tareas.
- **Usabilidad:** El código no tiene porque estar mal escrito o ser incorrecto, sin embargo, la forma en la que se disponen los elementos o está pensado puede frustrar al usuario final que utiliza la web.

Para asegurarnos de que nuestro código está correctamente escrito, podemos utilizar un **Validador HTML**, que no es más que un sistema que analiza nuestro código y nos dice el número de errores que tenemos, junto a una breve descripción del mismo para facilitar su corrección.

Este proceso de validación se puede realizar mediante la herramienta oficial [HTML Validator de W3C](#) u otras vías de validación mediante [plugins para el IDE](#) o [paquetes NPM](#).

Etiquetas obsoletas

Con el paso del tiempo y la transición desde versiones anteriores a HTML5 (por ejemplo, desde HTML4 o XHTML), hay muchas etiquetas HTML que han sido marcadas como [obsoletas](#) y se recomienda dejar de utilizarlas o utilizar otras [alternativas](#).

Etiquetas de cabecera

(TODO)

Enlaces de interés

- <https://lenguajehtml.com/html/>
- <https://html.spec.whatwg.org/dev/>
- <https://html.spec.whatwg.org/>
- <https://developer.mozilla.org/es/docs/Web/HTML>
- <https://www.w3schools.com/html/>
- <https://htmlreference.io/>
- <https://html.com/>
- <https://cheatsheets.shecodes.io/html>
- <https://overapi.com/html>
- <https://htmlcheatsheet.com/>
- <https://devhints.io/>
- <https://internetingishard.netlify.app/html-and-css/>
- <https://www.theodinproject.com/>
- <https://roadmap.sh/frontend>
- <https://jsfiddle.net>

Licencia



Esta obra está bajo una [licencia de Creative Commons Reconocimiento-Compartir Igual 4.0 Internacional](#).