

## 1. INTRODUCCIÓN

La World Wide Web, más conocida como web, es el servicio de la red internet más popular. Su principal objetivo es permitir la distribución de documentos entre personas, con independencia de la plataforma o del sistema operativo que utilicen. La web es, junto con el correo electrónico, uno de los servicios que han hecho de internet la red global que es.

HTML (junto con XHTML, su versión XML) es el lenguaje que se utiliza para dar forma a los documentos publicados en la web. Es gratuito, sencillo, potente y no requiere de ningún software sofisticado para su edición, ya que un editor de texto es suficiente para crear cualquier tipo de documento.

### 2.1. Introducción, evolución y estado actual de HTML

HTML es, sin ninguna duda, el lenguaje de marcas más conocido y utilizado. Sus siglas significan en castellano «lenguaje de marcado de hipertexto» (HyperText Markup Language) y es el lenguaje en el que están escritas la gran mayoría de páginas de la World Wide Web (lo que comúnmente se conoce como web). Obviamente, todos los navegadores web reconocen HTML y lo saben interpretar y traducir, generando la representación de la página web tal y como la vemos los usuarios. El lenguaje HTML y la web son términos que van necesariamente de la mano, ya que de manera independiente no habrían tenido la repercusión que han conseguido conjuntamente.

En la actualidad, HTML está gestionado por el W3C (World Wide Web Consortium), un consorcio dedicado a la estandarización de la web para lograr que esta sea accesible, universal, fiable, fácil de usar y de uso libre. HTML es, por lo tanto, gratuito. Ninguna persona ni empresa debe pagar por utilizarlo.

En 1991 Tim Berners-Lee diseñó HTML como el lenguaje de marcado capaz de proporcionar un mecanismo sencillo, flexible y universal para almacenar y transmitir información entre distintos sistemas informáticos. Aunque las redes de ordenadores ya existían desde muchos años antes, internet proponía una red global de comunicaciones y había que dotarla de soluciones para que las personas que las fuesen a utilizar pudiesen compartir información fácilmente. Ese primer diseño era muy simple e incluía 18 etiquetas, algunas de las cuales aún se mantienen en la versión 5 de HTML. Difícilmente Tim Berners-Lee pudo anticipar el éxito de su creación, lo cual no le quita ningún mérito. HTML está bien diseñado, es sólido y se fundamenta en unas bases acertadas.

Existen dos hechos fundamentales que han marcado el destino de HTML:

- Los destinatarios de HTML han sido, desde el principio, las personas.

- La información proporcionada por HTML es de naturaleza estética y no semántica.

Estas dos características tienen algunas consecuencias. Una de ellas es que los navegadores, al procesar el código HTML, si encuentran algún error de sintaxis, pueden intentar solucionarlo, y si la solución propuesta no es correcta, los resultados no serán catastróficos.

Las personas tenemos una capacidad de interpretación de la información mucho más flexible que las máquinas, por lo que, ante un error de interpretación por parte del navegador, el resultado nos resultará inconsistente o poco atractivo estéticamente, pero seguro que seremos capaces de interpretarlo de forma correcta. La semántica la aporta el usuario en función de lo que ve y de su buen criterio. Si la página no está bien maquetada, el autor de esta se dará cuenta visualmente del error. Si no lo percibe, es que el error no es relevante y las consecuencias no serán graves.

Por otro lado, si es un programa informático el que tiene que analizar y procesar el texto, este ha de ser riguroso y carente de errores sintácticos. Un programa de ordenador es determinista (ante las mismas entradas debe generar las mismas salidas) y esto implica que no dispone de la capacidad de interpretar en el sentido humano.

Es por estas circunstancias por las que los navegadores web son laxos en la interpretación del código HTML y es una de las razones de la popularidad y alcance de la web. Si hacer páginas HTML fuese muy difícil, su aceptación habría sido probablemente más lenta. Por suerte, los diseñadores y programadores de los navegadores entendieron que el proceso de escribir una página web no puede ser igual de estricto que el proceso de escribir un programa informático en cualquier lenguaje de programación.

HTML, desde su nacimiento, ha pasado por diversas revisiones que han ido ampliando y mejorando la capacidad del lenguaje.

Año	Versión	Observaciones
1991	Diseño inicial	Incluye 18 etiquetas. Se considera como un dialecto de SGML. Se plasma en un documento interno e informal del CERN (Conseil Européen pour la Recherche Nucléaire u Organización Europea para la Investigación Nuclear, en español).
1992	HTML 1.1	Primer borrador informal.
1995	HTML 2.0	Publicado por el IETF (Internet Engineering Task Force o Grupo de Trabajo de Ingeniería de Internet, en español).
1997	HTML 3.2	Publicado como recomendación del W3C (World Wide Web Consortium).
1997	HTML 4.0	Publicado como recomendación del W3C.
1999	HTML 4.1	Publicado como recomendación del W3C.
2000	XHTML	Publicado como recomendación del W3C. Tenía como objetivo sustituir a HTML. En HTML5 se incorpora como parte de la recomendación y deja de desarrollarse por separado.
2014	HTML 5.0	Publicado como recomendación del W3C.
2016	HTML 5.1	Publicado como recomendación del W3C.
2017	HTML 5.2	Publicado como recomendación del W3C.

Cada versión mejora a la anterior. Incorpora nuevas etiquetas, elimina otras obsoletas y se acomoda a las capacidades tecnológicas. Hay que tener en cuenta que el avance en las tecnologías relacionadas con la informática y las telecomunicaciones es de gran importancia y se da en cortos periodos de tiempo.

Por ejemplo, en los primeros años de existencia de internet, el acceso a la red se hacía a través de la línea telefónica analógica mediante el uso de un módem (convertor de señales digitales-analógicas y analógicas-digitales) a una velocidad impensable para las velocidades de las líneas digitales actuales. Se ha pasado en treinta años de limitar el número de bytes que tiene una fotografía para agilizar la carga de la página a retransmitir vídeo en streaming en alta definición.

HTML ha ido adaptándose con éxito a esos avances para seguir siendo el lenguaje estándar- dar de marcado para la difusión de contenidos a través de la web, aceptado por toda la industria tecnológica sin objeciones.

La versión 5 (y posteriores) de HTML ha recogido las necesidades de un mercado dinámico y vertiginoso y ha abierto un nuevo e importante conjunto de oportunidades gracias a su potencial y a su capacidad, respetando los principios de su fundación.

## 2.2. Estructura y Sintaxis del lenguaje HTML

El lenguaje de marcado HTML define el contenido de una página web. Los navegadores interpretan las marcas contenidas en los documentos HTML y representan la información para que los usuarios la puedan consultar e interactuar con ella. Trabajando junto con las hojas de estilo CSS, HTML es un increíble lenguaje de representación de información.

El lenguaje HTML ha tenido un enorme éxito debido, en parte, a su sencilla estructura y sintaxis.

### 2.2.1. Estructura y sintaxis

Los elementos que forman las páginas HTML están identificados por marcas o etiquetas. Estas marcas «envuelven» el contenido de la página, ya sea este texto, imágenes o cualquier otro tipo de elemento. Las marcas o etiquetas están compuestas por un nombre rodeado de los símbolos «<» y «>».

<etiqueta>

El nombre de la etiqueta puede estar escrito en mayúsculas o minúsculas, ya que no se distingue entre ambos tipos de letra. No obstante, se recomienda que el etiquetado sea uniforme.

Existen dos tipos principales de etiquetas: las etiquetas de apertura y las de cierre. Un contenido debe estar rodeado por ambas etiquetas. La etiqueta de apertura se incluye antes del contenido y la etiqueta de cierre después de este. La etiqueta de apertura se diferencia de la etiqueta de cierre en que esta última incluye el símbolo "/" justo antes del nombre.

```
</etiqueta>
```

Ambas etiquetas tendrán necesariamente el mismo nombre y rodearán un contenido. El conjunto formado por el contenido y las etiquetas de apertura y cierre forman un elemento.

En HTML no todas las etiquetas necesitan apertura y cierre, ya que no todas son contenedoras. Un ejemplo es la etiqueta «<br>». Esta etiqueta provoca un salto de línea en el texto, pero no debe rodear contenido, ya que el salto de línea es independiente del contexto. Por lo tanto, algunos elementos solo estarán formados por una etiqueta de apertura. A estos elementos se les denomina elementos vacíos.

```
<etiquetasincierre>
```

Dentro de un elemento se pueden incluir atributos. Los atributos permiten caracterizar el elemento de cara a identificarlo, representarlo o tratarlo de manera específica. La existencia de atributos es opcional y su número puede variar. Todos los elementos de un documento HTML pueden disponer de atributos, pero no todos los atributos son aplicables a cualquier elemento.

Los atributos se incluyen en la etiqueta de apertura y están compuestos por un nombre, el símbolo «=» y el valor del atributo entrecomillados. Las comillas pueden ser simples o dobles. De nuevo, se recomienda que el uso de comillas simples o dobles sea homogéneo a lo largo del documento.

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

Los elementos de un documento HTML se pueden anidar. Un elemento puede contener a uno o más elementos en su interior.

```
<elementonivel1>
  <elementonivel2>
    <elementonivel3>
      Contenido
    </elementonivel3>
  </elementonivel2>
</elementonivel1>
```

Esta anidación debe estar bien construida. No se permite que los elementos se solapen. En el siguiente código se muestra un ejemplo de una anidación mal construida:

```
<elementonivel1>
    <elementonivel2>
        <elementonivel3>
            Contenido
        </elementonivel2>
    </elementonivel3>
</elementonivel1>
```

Una página HTML está compuesta por un conjunto de elementos y alguna información adicional que se indicará a continuación. Este conjunto de elementos debe respetar una estructura definida en el estándar de HTML.

En el siguiente código se muestra la estructura básica de una página HTML:

```
<!DOCTYPE html>
<html>
<head>

</head>
<body>

</body>
</html>
```

El primer elemento (línea 1) identifica el tipo del documento. Este elemento es un elemento vacío y, por lo tanto, no requiere de etiqueta de cierre. Aunque no es exactamente un elemento de HTML se incluye en las páginas HTML para indicar al navegador de qué tipo es el documento que va a procesar.

El siguiente elemento (entre las líneas 2 y 9) se define con la etiqueta `<html>`. Es el elemento raíz y dentro de él se encuentra todo el contenido de la página.

El elemento etiquetado como `<head>` (entre las líneas 3 y 5) incluye información no visible en la representación de la página dentro del navegador pero que sí afecta a cómo esta va a ser presentada. En esta sección se indica, por ejemplo, el título que aparece en la pestaña del navegador, la hoja de estilos CSS que utilizar o el juego de caracteres que se usará para la representación de los textos.

Por último, el elemento `<body>` (entre las líneas 6 y 8) albergará el contenido visible

de la página, ya sea este texto, tablas, listas, imágenes o elementos multimedia, por citar algunos ejemplos.

#### Ejemplo:

```
<!DOCTYPE html>

<html lang="es">

  <head>

    <meta charset="UTF-8">

    <title>Documento básico</title>

  </head>

  <body>

    Este es un documento básico.

  </body>

</html>
```

### 2.2.2 Entidades

Algunos caracteres tienen un significado especial en HTML, son invisibles o, sencillamente, no existen en el lenguaje en el que está escrita la página. Por ejemplo, el símbolo "menor que" (<) es el comienzo de una etiqueta, los espacios en blanco múltiples son ignorados o las tildes no existen en algunos idiomas. Las entidades permiten indicar al navegador que debe representar un carácter determinado, solucionando los problemas derivados de estas circunstancias.

Por ejemplo, utilizando el juego de caracteres ISO-8859-1, el texto ¡Soy un campeón! se muestra en el navegador como ÁiSoy un campeA3n!. Como se puede observar, el símbolo "i" y el carácter "Ó" no se han visualizado correctamente. Aunque este problema se podría resolver utilizando el juego de caracteres *UTF-8*, también se podría solucionar sustituyendo los caracteres problemáticos por su entidad equivalente: el carácter "i" se sustituye por la entidad &iexcl; y el carácter «Ó» por &oacute;. El texto en el documento HTML quedaría de la siguiente manera: &iexcl;Soy un campe&oacute;n!

Otro ejemplo tiene que ver con cómo se interpretan los espacios múltiples en HTML, ya que solo se respeta el primero de ellos. Según este comportamiento, el texto Dejemos espacio entre la A y la B: **A B** se muestra en el navegador como Dejemos espacio entre la A y la B: **A B**. Se observa que no se han respetado los espacios en blanco múltiples entre las letras A y B, quedándose los cinco espacios del código HTML original en uno único mostrado en el navegador. La entidad &nbsp; representa un espacio en blanco, por lo que la solución consistiría en sustituir el código HTML original por Dejemos espacio entre la A y la B:

A      B

Una entidad en HTML es un conjunto de caracteres que permiten indicar al navegador que represente caracteres que, por diversas razones, no se representan correctamente. Comienzan por el símbolo **&** y terminan por el símbolo **;**.

Algunas de las entidades más comunes son las siguientes:

&	&amp;
<	&lt;
>	&gt;
"	&quot;
€	&euro;
'	&apos;
©	&copy;

### 2.2.3 Comentarios

HTML admite incluir líneas de comentarios. Estos elementos no tienen ningún efecto sobre la página, ya que su único objetivo es proporcionar información a los diseñadores y programadores cuando están consultando el código interno de esta. El contenido se delimita con los caracteres **<!--** y **-->** y será ignorado por el navegador. A continuación, se muestra un ejemplo de un comentario incrustado en un bloque de código HTML:

```
<article id="capitulol">
  <!-- CONTENIDO DEL CAPÍTULO 1-->
</article>
```

Ejemplo:

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

```
<head>

  <meta charset="UTF-8">

  <title>Mi primera página</title>
</head>

<body>

  <!--Esta es mi primera página HTML-->

  <h1> ¡HOLA!</h1>

</body>
</html>
```

## 2.3. Elementos de HTML

Un documento HTML está compuesto de elementos organizados de manera jerárquica. El elemento `<html>` es la raíz y todos los demás descienden directa o indirectamente de él.

Como se ha indicado anteriormente, los elementos `<html>`, `<head>` y `<body>` son fundamentales dentro de la construcción de la página para indicar que el documento es HTML (elemento `<html>`), cómo se ha de representar la información (elemento `<head>`) y qué información se debe representar (elemento `<body>`), pero solo son la punta de un enorme iceberg.

La relación de elementos de HTML es muy extensa y admite múltiples clasificaciones. En esta sección se muestran los elementos organizados por familias en función de su funcionalidad. En algunos casos, debido a su singularidad y relevancia, se muestran elementos de manera individual y no agrupados.

Cada elemento se identifica por una etiqueta y se caracteriza por un conjunto de atributos, muchos de los cuales son específicos del elemento en concreto. No obstante, existen algunos atributos globales que es importante conocer, ya que son determinantes para el correcto funcionamiento de tecnologías relacionadas con HTML como las hojas de estilo CSS o el lenguaje de programación JavaScript. Los atributos globales son comunes a todos los elementos HTML.

Estos son los atributos globales más utilizados:



Atributo	Descripción
class	Permite agrupar elementos. El valor de este atributo es una cadena de caracteres definida por el autor de la página. Si varios elementos comparten el mismo valor, se dice que son de la misma clase. Se utiliza para asignar estilos y durante la programación con JavaScript.
contenteditable	Si existe, el contenido del elemento es editable.
hidden	Si existe, el elemento no se representa en el navegador.
id	Identifica al elemento de manera única en el documento. El valor es una cadena de caracteres definida por el autor de la página. Se utiliza para asignar estilos y durante la programación con JavaScript.
lang	Determina el idioma en el que está escrito el contenido del elemento.
spellcheck	Determina si el elemento debe ser analizado ortográficamente y gramaticalmente por el navegador.
style	Permite asignar un estilo al componente.
tabindex	Determina el orden de selección de elementos cuando se pulsa el tabulador.
title	Permite especificar información extra al elemento.
translate	Determina si el elemento debe ser traducido o no.

En el siguiente código de ejemplo se pueden ver algunos de los atributos globales de HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h2 id="encabezado" class="rojo">Encabezado</h2>
  <p class="rojo" style="background-color: aquamarine;">
    Este párrafo será visible en el navegador
  </p>
  <p id="explicacion" hidden>
    Este párrafo no será visible en el navegador
  </p>
  <p contenteditable spellcheck="true" tabindex="2">
    Este párrafo tiene faltas de ortografía y es editable
  </p>
  <p lang="en" translate="no">
    This paragraph is written in English
  </p>
  <input type="text" tabindex="1">
  <input type="text" tabindex="3">
</body>
</html>
```

La representación generada por el navegador es la siguiente:

## Encabezado

Este párrafo será visible en el navegador

Este párrafo tiene faltas de ortografía y ES MODIFICABLE

This paragraph is written in English

Algunas de las características que proporcionan los atributos globales no son visibles. Por ejemplo, el párrafo que tiene el atributo *hidden* no se muestra en el navegador. Tampoco la navegación con las pulsaciones de la tecla de tabulación: la primera pulsación posiciona el cursor en el primer elemento `<input>`; la segunda pulsación posiciona el cursor en el párrafo que tiene el atributo *contenteditable*; la tercera pulsación posiciona el cursor en el último elemento `<input>`. También hay atributos globales que tienen un efecto visible en la página: el color de fondo o el resultado de realizar la corrección ortográfica (depende del navegador). El resto de los atributos globales no tienen un efecto visible ni interno a priori, como el atributo *id* y el atributo *class*, que sirven para poder referenciar a los elementos desde las hojas de estilo CSS y desde programas escritos en JavaScript.

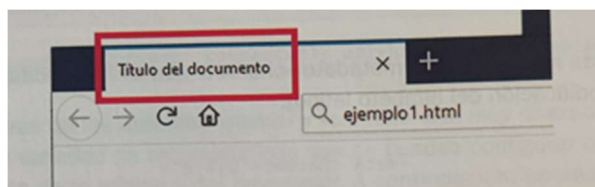
### 2.3.1. Elemento HTML

Es el elemento principal y la raíz del documento HTML. Todo el contenido del documento HTML, agrupado en los elementos `<head>` y `<body>`, está incluido dentro de este elemento.

### 2.3.2. Elemento Head

Es la primera de las secciones de una página web. No será directamente visible en la representación del documento ni en el navegador propiamente dicho (con alguna excepción) pero sí tendrá efecto directo y relevante en dicha representación.

El elemento más habitual que se incluye en esta sección es `<title>`. Contiene el título:



Además de mostrarse en la pestaña correspondiente del navegador, el título es el nombre que se asigna a una dirección web cuando se almacena como favorito en el navegador. El título también lo tienen en cuenta los buscadores de internet para clasificar la

página, por lo que una correcta asignación de este hará que el posicionamiento sea más acorde al contenido de dicha página.

El elemento «title» no es obligatorio, pero sí altamente recomendable, ya que en caso de no utilizarse el navegador, mostrará el nombre del fichero y esto, de cara al usuario final, es una información irrelevante y probablemente confusa.

La optimización del posicionamiento en buscadores, conocido como SEO (Search Engine Optimizati3n) est3 formada por un conjunto de t3cnicas que pretenden mejorar la posici3n en la que aparecer3 una p3gina web al hacer una b3squeda relacionada en un buscador como Google. La etiqueta <title> es uno de los elementos que utilizan los buscadores para posicionar la p3gina.

Adem3s del t3tulo, en esta secci3n se especifican los denominados metadatos. Un meta-dato es, en este contexto y utilizando una definici3n muy simplificada, un dato descriptivo que va a afectar a c3mo se interpretan los datos contenidos en el documento.

Los metadatos se identifican con la etiqueta <meta> seguida de los atributos y valores correspondientes a la informaci3n que se quiere proporcionar al navegador.

A continuaci3n, se indican algunos de los metadatos m3s habituales:

METADATO	DESCRIPCI3N
charset	Juego de caracteres. El est3ndar es UTF-8
http-equiv	Se utiliza para especificar una directiva. Todos los valores permitidos son nombres de encabezados HTTP
name	Nombre del metadato. Se debe usar juntamente con el metadato <i>content</i>
content	El valor asociado a atributos como <b>name</b> y <b>http-equiv</b> . Dependiendo del tipo de atributo, ser3 un texto o una opci3n dentro de un conjunto

Por su parte, el metadato «http-equiv» tiene más alcance, por lo que la definición es sustancialmente más compleja. Mediante este metadato se pueden especificar aspectos tan diferentes como el tipo de caché que requiere la página, qué debe hacer el navegador ante la descarga de determinado documento, si el contenido está codificado con algún algoritmo de compresión, la fecha de publicación o cada cuanto tiempo debería recargar la página el navegador de manera automática.

En el siguiente listado se muestran algunos ejemplos de uso de este metadato junto con la descripción de su comportamiento.

- **Content-type.** Especifica el tipo de contenido y el juego de caracteres. En el ejemplo que se muestra a continuación, es de tipo texto utilizando el lenguaje html la codificación UTF-8

```
<meta http-equiv="Content -Type" content="text/html; charset=UTF-8">
```

- **Cache-control.** Especifica cómo ha de gestionar el navegador la página HTML desde el punto de vista de la caché. Se puede indicar una fecha de expiración, que solo se almacene en memoria y no en soporte físico o desactivar por completo el almacena miento en caché con el objetivo de que cada vez que se consulte la página se vuelva a descargar al servidor. En el siguiente ejemplo se está indicando al navegador que no almacene la página en caché.

```
<meta http-equiv="cache-control" content="no-store">
```

- **refresh.** Indica al navegador que, pasado un determinado periodo de tiempo, refres que la página o, alternativamente, cargue una página referenciada a través de una URL. En la etiqueta que se muestra como ejemplo a continuación: (3 seg).

```
<meta http-equiv="refresh" content="3; url=http://www.marca.es/">
```

Los posibles valores de los atributos **name** y **content** son muy diversos y hacen referencia a una gran variedad de características que se pueden configurar para determinar el comportamiento de la página y del navegador. A continuación, se muestran algunos de ellos:

- **robots.** Hace recomendaciones a los robots utilizados por los motores de búsqueda. Permite indicar, por ejemplo, que no se desea que se indexe una página o las páginas a las que dirigen los enlaces de la página actual. También se puede hacer saber al robot que se quiere que la página sea indexada, aunque

esto lo suelen hacer por defecto. En el ejemplo siguiente se indica a los potenciales robots que no indexen la página:

```
<meta name="robots" content="noindex"/>
```

- **keywords.** Indica palabras clave que servirán para que los indexadores de los motores de búsqueda sean capaces de clasificar de una manera más eficiente la página. El número de palabras clave puede ser variable y deberán hacer referencia al contenido de la página.

```
<meta name="keywords" content="editorial, marcas, html"/>
```

- **author.** Permite indicar el nombre del autor.

```
<meta name="author" content="El nombre del autor" />
```

- **copyright.** Permite indicar el propietario de los derechos de la página web.

```
<meta name="copyright" content="Ediciones Santillana" />
```

**viewport.** Permite determinar el ancho de la zona de representación (viewport) de la página, la escala inicial o las posibilidades de escalado que se le aplican a la página.

```
<meta name="viewport" content="width=device-width, initialscale=1.0">
```

### 2.3.3. Elemento Body

El elemento `<body>` delimita la sección que contiene los elementos que forman la parte visible de la página HTML. Es, por lo tanto, la sección más importante, ya que en ella se encuentran los contenidos que se van a presentar en el navegador.

El número de etiquetas que se pueden incluir en esta sección es ingente y en esta unidad se muestra una importante relación de estas.

Los elementos de la sección delimitada por el elemento `<body>` se clasifican en elementos de bloque y elementos de línea. Dichos tipos tienen las siguientes características.

- Elementos de bloque
  - ✓ Ocupan todo el espacio de su elemento padre (su contenedor).
  - ✓ Salvo excepciones, permiten contener a otros elementos, ya sean estos

de bloque o de línea.

- ✓ Salvo excepciones, siempre comienzan y terminan con un salto de línea.

—

- Elementos de línea

- ✓ Ocupan lo que ocupa su contenido.
- ✓ Contienen datos u otros elementos de línea.
- ✓ Pueden comenzar en cualquier lugar de una línea y no generan salto de línea.
- ✓

#### **2.3.4. Elementos de bloque**

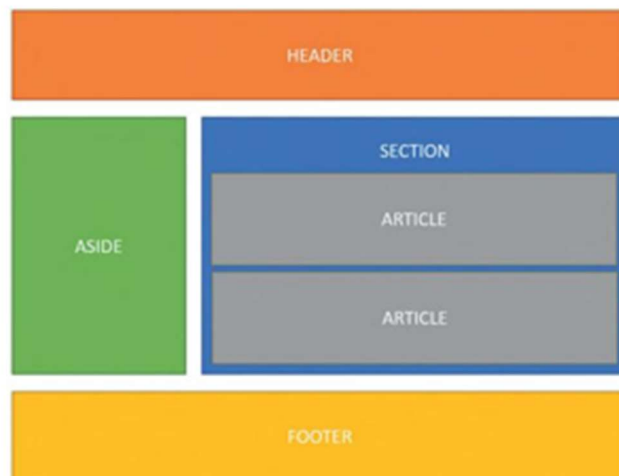
Los elementos de bloque forman un conjunto muy heterogéneo, ya que tienen en común características relacionadas con el espacio que ocupan y su distribución en la página. No obstante, es una de las categorías que se pueden utilizar para clasificar los elementos de HTML.

En esta sección, se muestra la relación de elementos de bloque, ordenados alfabéticamente. Se incluye su descripción, referencias a los atributos más significativos (exceptuando los atributos globales), ocasionalmente código HTML de ejemplo junto con su representación visual y, cuando proceda, información complementaria que ayude a comprender el componente. La lista de atributos no es exhaustiva, ya que esta es muy extensa, por lo que se sugiere acudir a manuales de referencia o a documentación técnica para profundizar en las características de cada componente.



Los elementos de bloque permiten, entre otras cosas, estructurar y organizar la página para facilitar su correcta interpretación por parte de los programas informáticos. Una posible estructura de página se muestra en la Figura 2.4. En dicha estructura, el contenido del bloque HEADER podría contener, por ejemplo, los enlaces para la navegación general; el bloque ASIDE, los enlaces para navegación referente a la sección en la que se encuentre el usuario; el bloque SECTION, una sección contenedora de dos artículos incluidos en sendos bloques ARTICLE, y el bloque FOOTER, podría incluir referencias a la empresa propietaria de los derechos de la página, además de otra información.

Cada uno de los contenedores, desde el más general hasta el más específico, es un elemento de bloque.



Esta estructura, como se ha comentado, es una referencia y admite múltiples variantes. Lo realmente importante es ubicar cada contenido en un bloque semántico correspondiente para facilitar el procesamiento automático del documento.

Se detallan a continuación los elementos que se engloban en esta categoría, excepto los elementos `<form>` y `<fieldset>` y el elemento `<table>`, que se exponen en apartados posteriores

### **`<address>`**

Descripción: Representa información de contacto (dirección de correo electrónico, número de teléfono, etc.) para el elemento `<article>` más cercano o para el elemento `<body>` en el que se encuentra. El texto se muestra en cursiva.

Ejemplo:

```
<address>

Puede contactar con nosotros a través del sitio web <a
href="https://www.nuestraempresa.es/">www.nuestraempresa.es< /a>

</address>
```

### **`<article>`**

Descripción: Este contenedor agrupa información concebida como una unidad independiente. Algunos ejemplos serían una noticia de un periódico, un artículo de una revista, una entrada en un blog o un comentario en un foro. No genera ninguna representación gráfica específica.

## **<aside>**

Descripción: agrupa contenidos que están relacionados con el contenido principal del documento. Al ser un elemento de orientación puramente semántica, no tiene representación visual específica. No obstante, se representa habitualmente como una barra lateral en la que se muestran enlaces relacionados, publicidad o información complementaria al contenido principal del documento.

## **<blockquote>**

Descripción: contiene unacita.

### **Ejemplo:**

```
<blockquote cite='https://webfoundation.org/2017/03/web-turns-28-letter/'>
  <p>El modelo comercial actual de muchos sitios web ofrece contenido
gratuito a cambio de datos personales. Muchos de nosotros estamos de acuerdo
con esto, aunque a menudo al aceptar documentos de términos y condiciones
largos y confusos, pero fundamentalmente no nos importa que se recopile
cierta información a cambio de servicios gratuitos.</p>
</blockquote>
```

## **<details>**

Descripción: Almacena un contenido textual que se hace visible cuando se «despliega» el componente.

Puede incluir como hijo el elemento <summary> para representar una etiqueta que identifique el contenido. De no usarse <summary>, se mostrará una etiqueta genérica.

Se trata de una tecnología experimental, por lo que algunos navegadores pueden no interpretar este elemento correctamente.



```
<details>
  <summary>Nota para el usuario</summary>
  El contenido de este manual está actualizado en relación a la versión
  10.8 del producto.
</details>
```

## <dialog>

Descripción: muestra una caja de diálogo. El atributo "open" se pone cuando queremos abrir el diálogo.

```
<dialog open> <p> ¡La compra se ha realizado correctamente! </p></dialog>
```

## <div>

Descripción: permite crear divisiones y agrupar contenidos, sean estos del tipo que sean. utiliza, junto con las hojas de estilo css, para maquetar y estructurar las páginas. No tiene una representación visual particular.

## <dl> <dt> y <dd>

Descripción: usados conjuntamente permiten definir una lista de descripciones. <dl> marca la lista de definición, <dt> cada uno de los términos definidos y <dd> la definición de cada término.

Ejemplo:

```
<p>Géneros de videojuegos: </p>
<dl>
  <dt>Arcade</dt>
  <dd>Tienen como objetivo alcanzar una meta salvando obstáculo y/o
  enemigos. Suelen ser muy dinámicos y tienen mecánicas sencillas </dd>
  <dt>Estrategia</dt>
  <dd>Tiene como objetivo organizar una serie de recursos y maximizar su
  rendimiento, ya sea este bélico, económico, social, etc.</dd>
  <dt>Disparos</dt>
  <dd>Tiene como objetivo alcanzar una meta o cumplir una misión en un
  entorno hostil. También conocidos como <em>shooters</em></dd>
</dl>
```

## **<figure> y <figcaption>**

Descripción: representan una ilustración, figura, diagrama, etc., y su leyenda o subtítulo (mediante el elemento <figcaption>)

```
<figure>
  
  <figcaption>El logotipo del World Wide Web Consortium (W3C)</figcaption>
</figure>
```

La accesibilidad de una página web es la capacidad que tiene para garantizar el acceso a la información contenida en ella a personas con algún tipo de discapacidad. El W3C define una serie de pautas para garantizar que una página es accesible. Puedes profundizar en la accesibilidad web a través de este enlace:

<https://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

## **<footer>**

Descripción: este contenedor representa un pie de página. Típicamente contiene información sobre la propiedad, autoría y los derechos del documento. Es un elemento semántico por lo que agrupa, pero no proporciona ninguna aportación visual.

## **<h1>,<h2>,<h3>,<h4>,<h5>,<h6>**

Descripción: representan elementos de encabezado. Tienen distintos niveles de importancia. El elemento <h1> es el más importante y el elemento <h6>, el menos importante.

Cada elemento tiene una representación distinta en el navegador (a mayor importancia, mayor tamaño del texto) que se puede replicar mediante hojas de estilo, pero esto no es una buena práctica.

Los agentes de usuario utilizan los elementos de encabezado para construir las tablas de contenido, lo que mejora la visibilidad de la página en los buscadores como Google. Utilizar estos elementos mejora, además, el posicionamiento (SEO), por lo que es muy importante utilizar los de manera correcta.

Ejemplo:

```
<h1>Encabezado de nivel 1</h1>
<h2>Encabezado de nivel 2</h2>
<h3>Encabezado de nivel 3</h3>
<h4>Encabezado de nivel 4</h4>
<h5>Encabezado de nivel 5</h5>
<h6>Encabezado de nivel 6</h6>
```

## <header>

Descripción: contiene el encabezado de un documento o de una sección. En este elemento, de naturaleza semántica, se suelen ubicar los logotipos e imágenes corporativas, los formularios de búsqueda, la cesta de la compra en sitios web de venta online o enlaces de navegación de alto nivel. No tiene una representación visual específica.

## <hr>

Descripción: representa un cambio de tema entre párrafos mediante una línea de división. Aunque se representa como una línea, su principal objetivo es aportar información semántica. No puede ser contenedor.

```
<p>Este texto está en un párrafo.</p>
<hr>
<p>Este texto está en otro párrafo.</p>
```

## <LI>, <OL>, <UL>

Descripción: representan una lista y sus elementos. La etiqueta <li> identifica cada uno de los elementos de la lista. Por su parte, las etiquetas <ol> y <ul> identifican una lista ordenada (sus elementos aparecen numerados) y una lista no ordenada (sus elementos aparecen identificados con viñetas). Las listas se pueden anidar.

```
<ol>
  <li>Subir al Kilimanjaro</li>
  <li>Visitar Nueva York</li>
  <li>Aprender a jugar al ajedrez</li>
  <li>Viajar en el tren Transiberiano </li>
</ol>
```

1. Subir al Kilimanjaro
2. Visitar Nueva York
3. Aprender a jugar al ajedrez
4. Viajar en el tren Transiberiano

**Ejemplo (lista no ordenada):**

---

```
<ul>
  <li>Lunes</li>
  <li>Martes</li>
  <li>Miércoles</li>
  <li>Jueves</li>
  <li>Viernes</li>
</ul>
```

- Lunes
- Martes
- Miércoles
- Jueves
- Viernes

---

**Figura 2.14.** Las listas sin ordenar se utilizan en relaciones de elementos cuyo orden no es importante.

**<main>**

Descripción: representa el contenido principal del documento. La motivación de esta etiqueta es principalmente semántica, ya que identifica de manera inequívoca el bloque de información principal de la página. Debe incluirse un único contenedor main dentro de un documento, y este no debe estar dentro de elementos <article>, <aside>, <footer>,

<header> ni <nav>, por razones semánticas. No tiene representación específica.

**<nav>**

Descripción: este contenedor está concebido para albergar enlaces de navegación, ya sean estos internos al documento o a otros documentos. Habitualmente se muestran las tablas de contenido, los índices, menús o sistemas de navegación alternativos. No tiene representación específica.

## <P>

Descripción: delimita un párrafo de texto. No tiene representación específica.

## <pre>

Descripción: representa un texto tal y como está en el documento HTML, respetando saltos de línea, tabulaciones y espacios múltiples. Utiliza una fuente de letra de ancho fijo.

```
<p>
  Este texto está dentro de un párrafo.
  Las tabulaciones, saltos de línea y los espacios      múltiples se ignoran.
</p>
<pre>
Este texto está dentro de un elemento pre.
  Los saltos de línea y los espacios      múltiples se respetan.
</pre>
```

Este texto está dentro de un párrafo. Las tabulaciones, saltos de línea y los espacios múltiples se ignoran.

Este texto está dentro de un elemento pre.  
 Los saltos de línea y los espacios múltiples se respetan.

## <section>

Descripción: representa una sección genérica del documento. En ocasiones su uso se confunde con el de <article>, pero su diferencia es importante. Los elementos agrupados por <article> están relacionados entre sí y forman un bloque autocontenido e independiente de información. En cambio, <section> tiene como objetivo agrupar información relacionada y dependiente, aportando estructura semántica. No tiene una representación específica.