

TEMA 3.

Elementos para la creación de contenido en HTML5

Lenguajes de Marcas
CFGS DAM 1

Autor: Pascual Ligeró

Revisado por:

Diana Bautista - d.bautistarayo@edu.gva.es

Óscar Villar - or.villarfernandez@edu.gva.es

2022/2023

Versión:171022.2254

Licencia



**CC BY-NC-SA 3.0 ES Reconocimiento – NoComercial –
CompartirIgual (by-nc-sa):**

No se permite un uso comercial de la obra original ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula la obra original.

NOTA: Esta es una obra derivada de la original realizada por Pascual Ligero.

Nomenclatura

A lo largo de este tema se utilizarán distintos símbolos para distinguir elementos importantes dentro del contenido. Estos símbolos son:



Importante



Atención



Interesante

ÍNDICE DE CONTENIDO

Contenido

1. ENLACES	4
1.1 URL	4
1.2 Enlaces relativos y absolutos	5
1.3 Enlaces básicos	10
1.4 Otros tipos de enlaces	13
1.5 Ejemplos de enlaces habituales	15
2. LISTAS	17
2.1 Listas no ordenadas	18
2.2 Listas ordenadas	19
3. IMÁGENES Y OBJETOS	20
3.1 Imágenes	20
3.2 Objetos	22
4. EMBEBIDOS Y MULTIMEDIA	23
4.1 Embebidos	23
4.2 Multimedia	24
5. TABLAS	24
5.1 Tablas básicas	25
5.2 Tablas avanzadas	31
7. FORMULARIOS	37
8. ELEMENTOS DE BLOQUE Y DE LÍNEA	39
9. SEMÁNTICA	41
10. BIBLIOGRAFÍA	42

UD03. Elementos para la creación de contenido en HTML5

1. ENLACES

El lenguaje de marcado HTML se definió teniendo en cuenta algunas de las características que existían en ese momento para la publicación digital de contenidos. Entre los conceptos utilizados en su creación, se encuentra el mecanismo de "hipertexto".

La incorporación del hipertexto fue una de las claves del éxito del lenguaje HTML, ya que permitió crear documentos interactivos que proporcionan información adicional cuando se solicita. El elemento principal del hipertexto es el "hiperenlace", también llamado "enlace web" o simplemente "enlace".

Los enlaces se utilizan para establecer relaciones entre dos recursos. Aunque la mayoría de enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos.

1.1 URL

Antes de empezar a crear enlaces, es necesario comprender y dominar el concepto de URL. El acrónimo URL (del inglés Uniform Resource Locator) hace referencia al identificador único de cada recurso disponible en Internet. Las URL son esenciales para crear los enlaces, pero también se utilizan en otros elementos HTML como las imágenes y los formularios.

La URL de un recurso tiene dos objetivos principales:

- Identificar de forma única a ese recurso
- Permitir localizar de forma eficiente ese recurso

En primer lugar, las URL permiten que cada página HTML publicada en Internet tenga un nombre único que permita diferenciarla de las demás. De esta forma es posible crear enlaces que apunten de forma inequívoca a una determinada página.

La cadena de texto `http://www.google.com` es la URL completa de la página principal de Google. La URL permite distinguir una página de otra.

El segundo objetivo de las URL es el de permitir la localización eficiente de cada recurso de Internet. Para ello es necesario comprender las diferentes partes que forman las URL.

Una URL sencilla siempre está formada por las mismas tres partes. Si por ejemplo se considera la siguiente URL del consorcio W3Schools:

`https://www.w3schools.com/html/html_intro.asp`

Las partes que componen la URL anterior son:

- Protocolo (https://): el mecanismo que debe utilizar el navegador para acceder a ese recurso. Todas las páginas web utilizan http://. Las páginas web seguras (por ejemplo las de los bancos y las de los servicios de email) utilizan https:// (Por fortuna cada vez son mas las páginas que utilizan el protocolo https).
- Servidor (www.w3schools.com): simplificando mucho su explicación, se trata del ordenador en el que se encuentra guardada la página que se quiere acceder. Los navegadores son capaces de obtener la dirección de cada servidor a partir de su nombre.
- Ruta (/html/html_intro.asp): camino que se debe seguir, una vez que se ha llegado al servidor, para localizar el recurso específico que se quiere acceder. (En este caso, la página de introducción al curso de HTML de W3Schools).

Por tanto, las URL no solo identifican de forma única a cada recurso de Internet, sino que también proporcionan a los navegadores la información necesaria para poder llegar hasta ese recurso.

La mayoría de URL son tan sencillas como la URL mostrada anteriormente. No obstante, existen URL complejas formadas por más partes.

<https://www.discogs.com/search/?q=paulina+rubio&type=all&page=2>

Las cinco partes que forman la URL anterior son:

- Protocolo (https://)
- Servidor (www.discogs.com)
- Ruta (/search/)
- Consulta (?q=paulina+rubio&type=all): información adicional necesaria para que el servidor localice correctamente el recurso que se quiere acceder. Siempre comienza con el carácter ? y contiene una sucesión de palabras separadas por = y &
- Paginación (&page=2): permite que el navegador se posicione automáticamente en una sección de la página web.

Como las URL utilizan los caracteres :, =, & y / para separar sus partes, estos caracteres están reservados y no se pueden utilizar libremente. Además, algunos caracteres no están reservados pero pueden ser problemáticos si se utilizan en la propia URL.

Carácter original Carácter codificado Carácter original Carácter codificado

'	%60	~	%7E
(espacio en blanco)	%20	#	%23

```
<!-- URL problemática --> http://www.ejemplo.com/ruta/nombre página.html
<!-- URL correcta -->
http://www.ejemplo.com/ruta/nombre%20p%E1gina.html
```

1.2 Enlaces relativos y absolutos

Las páginas web habituales suelen contener decenas de enlaces de diferentes tipos. La siguiente imagen muestra algunos de los tipos de enlaces de la página principal de un sitio web de prueba.

En esa página, cuando se pincha sobre algunos enlaces, el navegador abandona el sitio web para acceder a páginas que se encuentran en otros sitios. Estos enlaces se conocen como "enlaces externos". Sin embargo, la mayoría de enlaces de un sitio web apuntan a páginas del propio sitio web, por lo que se denominan "enlaces internos".

Además de internos/externos, la otra característica que diferencia a los enlaces (y por tanto, también a las URL) es si el enlace es absoluto o relativo.

Las URL absolutas incluyen todas las partes de la URL (protocolo, servidor y ruta) por lo que no se necesita más información para obtener el recurso enlazado. Las URL absolutas suelen corresponderse con enlaces externos. Las URL relativas prescinden de algunas partes de las URL para hacerlas más breves. Como se trata de URL incompletas, es necesario disponer de información adicional para obtener el recurso enlazado. En concreto, para que una URL relativa sea útil es imprescindible conocer la URL del origen del enlace. En este caso, el ámbito de las relativas suele ser el de los enlaces internos.

Las URL relativas se construyen a partir de las URL absolutas y prescinden de la parte del protocolo, del nombre del servidor e incluso de parte o toda la ruta del recurso enlazado. Aunque las URL relativas pueden ser difíciles de entender para los que comienzan con HTML, son tan útiles que todos los sitios web las utilizan.

Imagina que dispones de una página publicada en:

<http://www.ejemplo.com/ruta1/ruta2/pagina1.html> y quieres incluir en ella un enlace a otra página que se encuentra en <http://www.ejemplo.com/ruta1/ruta2/pagina2.html>. Como las URL identifican de forma única a los recursos de Internet y proporcionan la información necesaria para llegar hasta ellos, el enlace debería utilizar la URL completa de la segunda página. Las URL completas también se llaman URL absolutas, ya que el navegador no necesita disponer de información adicional para localizar el recurso enlazado. Si se utilizan siempre las URL absolutas, los enlaces están completamente definidos.

Sin embargo, escribir siempre las URL completas es bastante aburrido, cuesta mucho tiempo y hace imposible cambiar la ubicación de los contenidos de un sitio web. Por ese motivo, casi todos los sitios web de Internet utilizan URL relativas siempre que es posible.

Una URL relativa es una versión abreviada de una URL absoluta. Su objetivo es eliminar todas las partes de la URL absoluta que se pueden adivinar a partir de la información de contexto de la página web. En otras palabras, las URL relativas aprovechan la inteligencia de los navegadores para crear URL incompletas que los navegadores pueden completar deduciendo la información que falta.

Considerando de nuevo el ejemplo anterior, la URL a la que se quiere enlazar utiliza el mismo protocolo y se encuentra en el mismo servidor que la página origen, por lo que la URL relativa puede prescindir de esas partes:

URL absoluta: <http://www.ejemplo.com/ruta1/ruta2/pagina2.html> **URL relativa (enlace interno):** </ruta1/ruta2/pagina2.html>

En el ejemplo anterior, las dos URL son equivalentes porque cuando no se indica el protocolo y el servidor de una URL, los navegadores suponen que son los mismos que los de la página origen. Por lo tanto, cuando el navegador encuentra la URL </ruta1/ruta2/pagina2.html>, realiza el siguiente proceso:

1. La URL no es absoluta, por lo que se debe determinar la URL absoluta a partir de la URL relativa para poder cargar el recurso enlazado.
2. A la URL relativa le falta el protocolo y el servidor, por lo que se supone que son los mismos que los de la página origen (<http://> y www.ejemplo.com).
3. Se añaden las partes que faltan a la URL relativa para obtener la URL absoluta:
4. <http://> + www.ejemplo.com + </ruta1/ruta2/pagina2.html> = <http://www.ejemplo.com/ruta1/ruta2/pagina2.html>.

Aunque el ejemplo mostrado es el caso más sencillo de URL relativa, existen otros casos más avanzados en los que se prescinde de parte o toda la ruta del recurso que se enlaza. A continuación se muestran los cuatro tipos diferentes de URL relativas:

- 1) El origen y el destino del enlace se encuentran en el mismo directorio
- Si desde una página web se quiere enlazar un recurso que se encuentra en el mismo directorio del servidor, la URL relativa puede prescindir de todas las partes de la URL absoluta salvo el nombre del recurso enlazado.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	URL absoluta:
Página web llamada pagina2.html que se encuentra en el mismo directorio	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina2.html
	URL relativa:
	pagina2.html
	<p>Cuando el navegador encuentra una URL relativa que solo consiste en el nombre de un recurso, supone que el protocolo, servidor y directorio del recurso enlazado son los del origen del enlace.</p>

1. El destino del enlace se encuentra cerca de su origen y en un nivel superior

En este caso, el recurso que se enlaza no está en el mismo directorio que el origen del enlace pero sí que está *cerca* y en algún directorio superior. La URL relativa debe indicar de alguna manera que es necesario *subir* un nivel en la jerarquía de directorios para llegar hasta el recurso.

Para indicar al navegador que debe subir un nivel, se incluyen dos puntos y una barra (../) en la ruta del recurso enlazado. Cada vez que aparece ../ en una URL relativa, significa que se debe subir un nivel.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	URL absoluta:
Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta2	http://www.ejemplo.com/ruta1/ruta2/pagina2.html
	URL relativa:
	../pagina2.html
	<p>Cuando el navegador encuentra la URL relativa ../pagina2.html, sabe que para encontrar el recurso enlazado (pagina2.html) tiene que subir un nivel desde el lugar en el que se encuentra esa URL relativa. La página que incluye esa URL se encuentra en el</p>

directorio ruta1/ruta2/ruta3, por lo que subir un nivel equivale entrar en el directorio ruta1/ruta2.

De la misma forma, si el destino se encuentra un par de niveles por encima, se debe incluir ../ dos veces seguidas:

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada Página web llamada pagina2.html y que se encuentra en el directorio superior llamado ruta1	<p>URL absoluta: http://www.ejemplo.com/ruta1/pagina2.html</p> <p>URL relativa: .././pagina2.html</p> <p>Cuando el navegador encuentra la URL relativa ../pagina2.html, sabe que para encontrar el recurso enlazado (pagina2.html) tiene que subir un nivel desde el lugar en el que se encuentra esa URL relativa. La página que incluye esa URL se encuentra en el directorio ruta1/ruta2/ruta3, por lo que subir un nivel equivale entrar en el directorio ruta1/ruta2.</p>

Además de subir niveles, también se puede entrar en otros directorios para obtener los recursos:

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada Página web llamada pagina2.html y que se encuentra en un directorio llamado ruta4 que se encuentra en la raíz del servidor	<p>URL absoluta: http://www.ejemplo.com/ruta4/pagina2.html</p> <p>URL relativa: ../././ruta4/pagina2.html</p> <p>Si se intentan subir más niveles de los que es posible, el navegador ignora todos los ../ sobrantes. Si la página que tiene el enlace es http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html</p>

[pagina1.html](#) y la URL relativa que se incluye es `.././.././../pagina2.html`, el navegador realmente la interpreta como `.././../pagina2.html`.

El objetivo de las URL relativas es crear URL más cortas y sencillas que las URL absolutas, este método solo se puede utilizar cuando el origen y el destino se encuentran cerca, porque de otro modo la URL relativa se complica demasiado.

2. El destino del enlace se encuentra cerca de su origen y en un nivel inferior

Este caso es muy similar al anterior, pero más sencillo. Si el recurso enlazado se encuentra en algún directorio inferior al que se encuentra el origen, solo es necesario indicar el nombre de los directorios a los que debe entrar el navegador.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	URL absoluta:
Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta4	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/pagina2.html
	URL relativa:
	/ruta4/pagina2.html

De la misma forma, se pueden indicar varios directorios seguidos para que el navegador descienda jerárquicamente por la estructura de directorios:

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	URL absoluta:
Página web llamada pagina2.html y que se encuentra en un directorio inferior llamado ruta6 que está dentro del directorio ruta5 y que a su vez está dentro del directorio ruta4	http://www.ejemplo.com/ruta1/ruta2/ruta3/ruta4/ruta5/ruta6/pagina2.html
	URL relativa:
	/ruta4/ruta5/ruta6/pagina2.html

3. El origen y el destino del enlace se encuentran muy alejados

Cuando el origen y el destino de un enlace se encuentran muy alejados (pero en el mismo servidor) las URL relativas se pueden complicar en exceso. Aunque es posible utilizar ../para subir por la jerarquía de directorios y se puede entrar en cualquier directorio indicando su nombre, las URL relativas que se obtienen son demasiado largas y complicadas.

En estos casos, lo más sencillo es indicar la ruta completa hasta el recurso enlazado comenzando desde la raíz del servidor web. Por lo tanto, estas URL relativas solo omiten el protocolo y el nombre del servidor.

Elemento	Valor
Página origen	http://www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html
Página enlazada	URL absoluta:
Página web llamada pagina2.html y que se guarda en un directorio llamado ruta7 que se encuentra en la raíz del servidor	http://www.ejemplo.com/ruta7/pagina2.html
	URL relativa:
	/ruta7/pagina2.html
	Cuando la URL relativa comienza por /, el navegador considera que es la ruta completa comenzando desde la raíz del servidor, por lo que solo le añade el protocolo y el nombre del servidor origen.

1.3 Enlaces básicos

Los enlaces en HTML se crean mediante la etiqueta <a> (su nombre viene del inglés "anchor", literalmente traducido como "ancla"). A continuación se muestra la definición simplificada de <a> y más adelante se muestra su definición completa:

Etiqueta	<a>
Atributos comunes	básicos , internacionalización , eventos y foco
Atributos propios	<ul style="list-style-type: none"> href = "url"- Indica la URL del recurso que se quiere enlazar download = - Especifica si el recurso puede ser descargable al hacer clic en el enlace. Target = "_blank" (apertura en una nueva ventana) "_self" (apertura en la misma ventana)
Tipo de elemento	En línea
Descripción	Se emplea para enlazar todo tipo de recursos

El atributo más importante de la etiqueta `<a>` es `href`, que se utiliza para indicar la URL a la que apunta el enlace. Cuando el usuario pincha sobre un enlace, el navegador se dirige a la URL del recurso indicado mediante `href`. Las URL de los enlaces pueden ser absolutas, relativas, internas y externas.

Con la definición anterior, para crear un enlace que apunte a la página principal de Google solamente habría que incluir lo siguiente en un documento HTML:

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

Como el atributo `href` indica una URL, un enlace puede apuntar a cualquier tipo de recurso al que pueda acceder el navegador. El siguiente enlace apunta a una imagen, que se mostrará en el navegador cuando el usuario pinche sobre el enlace:

```
<a href="http://www.ejemplo.com/fondo_escritorio.jpg">Imagen interesante para un fondo de escritorio</a>
```

De la misma forma, los enlaces pueden apuntar directamente a documentos PDF, Word, etc.

```
<a href="http://www.ejemplo.com/informe.pdf" download>Descargar informe completo[PDF]</a>  
<a href="http://www.ejemplo.com/informe.doc" download>Descargar informe completo[DOC]</a>
```

Un truco muy útil con los enlaces es el uso de URL relativas para poder volver al inicio del sitio web desde cualquier página web interior:

```
<a href="/">Volver al inicio</a>
```

El enlace anterior utiliza una URL relativa con una ruta que apunta directamente a la raíz del servidor. Para obtener la URL absoluta, el navegador añade el mismo protocolo y el mismo nombre de servidor de la página en la que se encuentra el enlace. El resultado es que cuando se pincha ese enlace, siempre se vuelve al inicio del sitio web, independientemente de la página en la que se incluya el enlace.

La etiqueta `<a>` permite definir enlaces internos que referencian apartados de una misma página web. Si una página es muy larga, puede ser útil mostrar enlaces de tipo "Volver a la segunda sección", "Volver al principio de la página", etc.

Esta técnica puede conseguirse mediante la adición de atributos `id` a aquellos elementos sobre los que queramos definir un punto de retorno o ancla. Una vez definidos, podremos especificar enlaces que permitan navegar hasta dichas anclas desde un mismo documento html. Estudiaremos con calma los atributos de tipo `id` en las unidades correspondientes a estilos CSS. A continuación se muestra un ejemplo de documento HTML que utiliza esta técnica:

Mi web sencilla con anclas internas**Capítulo 1**

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem eaque officiis adipisci, cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!

Capítulo 2

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem eaque officiis adipisci, cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!

Capítulo 3

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem eaque officiis adipisci, cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!

Capítulo 4

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem eaque officiis adipisci, cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!

Capítulo 5

Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem eaque officiis adipisci, cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!

[Ir a capítulo 1](#)

```

<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My simple web</title>
</head>
<body>
  <style>
    h1{ text-align: center;}
    p, h1, h2, a{
      font-size: 1.5em;
    }
  </style>
  <h1>Mi web sencilla con anclas internas</h1>

  <h2 id="first">Capítulo 1</h2>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem adipisci,
    cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!</p>

  <h2>Capítulo 2</h2>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem adipisci,
    cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!</p>

  <h2>Capítulo 3</h2>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem adipisci,
    cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!</p>

  <h2>Capítulo 4</h2>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem adipisci,
    cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!</p>

  <h2>Capítulo 5</h2>
  <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Odit, atque voluptate iure voluptatem adipisci,
    cumque veritatis nesciunt ipsa amet cum ex ullam consectetur cupiditate aspernatur id? Mollitia, labore!</p>

  <a href="#first">Ir a capítulo 1</a>

</body>
</html>

```

1.4 Otros tipos de enlaces

Los enlaces mostrados en las secciones anteriores son los más utilizados por las páginas web. Los enlaces creados con la etiqueta <a> permiten enlazar cualquier tipo de recurso desde cualquier página. La característica más importante de estos enlaces es que el usuario debe activar la carga de los recursos. En otras palabras, el navegador no carga ningún recurso enlazado con la etiqueta <a> a menos que el usuario pinche sobre el enlace.

Además de estos enlaces, las páginas HTML pueden incluir otro tipo de enlaces que cargan los recursos automáticamente. Si una página HTML utiliza archivos CSS para aplicar estilos a sus contenidos, no es lógico que los enlace con la etiqueta <a> y espere a que el usuario pinche sobre el enlace para así cargar los archivos CSS. De la misma forma, muchas páginas web dinámicas necesitan que el navegador cargue varios archivos JavaScript para funcionar correctamente.

HTML define las etiquetas <script> y <link> para enlazar recursos que se deben cargar automáticamente. Cuando el navegador encuentra alguna de estas dos etiquetas, **descarga los recursos enlazados y los aplica a la página web.**

La etiqueta `<script>` tiene dos modos de funcionamiento, ya que se emplea tanto para **insertar un bloque de código JavaScript** en la página como para **enlazar un archivo JavaScript externo**.

Etiqueta	<code><script></code>
Atributos comunes	básicos , internacionalización , eventos y foco
Atributos propios	<ul style="list-style-type: none"> <code>src = "url"</code> - Indica la dirección del archivo que contiene el código <code>defer = "defer"</code> – El script será ejecutado cuando la página finalice su carga (scripts externos).
Tipo de elemento	Bloque y en línea (también puede ser una etiqueta vacía)
Descripción	Se emplea para enlazar o definir un bloque de código (normalmente JavaScript)

Descripción Se emplea para enlazar o definir un bloque de código (normalmente JavaScript)

Aunque la etiqueta `<script>` permite enlazar código de varios lenguajes de programación, el uso habitual de `<script>` consiste en enlazar un archivo JavaScript externo:

```
<head>
<script src="http://www.ejemplo.com/js/inicializar.js"></script>
</head>
```

El atributo `src` es equivalente al atributo `href` de los enlaces creados con la etiqueta `<a>`. La URL indicada en el atributo `src` puede ser absoluta o relativa y externa o interna.

Además de enlazar un archivo JavaScript externo, la misma etiqueta `<script>` también permite incluir en la página web un bloque de código JavaScript:

```
<!DOCTYPE html>
<html>
<head>
<script>
    window.onload = function() {
        alert("La página se ha cargado completamente");
    }
</script>
</head>
<body>
    ...
</body>
</html>
```

La inclusión del código JavaScript se recomienda realizarla justo antes de finalizar el apartado del `body`. Esto permite una pronta interpretación de los elementos de la página, así como de sus estilos.

La segunda etiqueta de HTML5 para enlazar recursos es `<link>`, que permite enlazar y relacionar la página con otros recursos externos.

Etiqueta	<link>
Atributos comunes	Básicos, internacionalización y eventos
Atributos propios	<ul style="list-style-type: none"> Los siguientes con el mismo significado que para la etiqueta "a": charset, href, hreflang, type, rel y rev media = "tipo_de_medio" - Indica el medio para el que debe aplicarse la relación
Tipo de elemento	Etiqueta vacía
Descripción	Se emplea para enlazar y establecer relaciones entre el documento y otros recursos

Al contrario que <script>, la etiqueta <link> solamente se puede incluir dentro de la cabecera del documento.

Se pueden añadir tantas etiquetas <link> como sean necesarias, pero siempre dentro de <head>...</head>.

El atributo media hace referencia al medio para el que es válida la relación con el recurso enlazado. Los medios disponibles también están estandarizados, siendo los más comunes screen para los contenidos mostrados en pantalla, print para las impresoras y handheld para los dispositivos móviles.

Un uso habitual de la etiqueta <link> es el de enlazar las hojas de estilos CSS utilizadas por las páginas web:

```
<head>
...
<link rel="stylesheet" href="/css/estilos.css" />
</head>
```

La URL del recurso enlazado se indica en el atributo href, que admite tanto URL absolutas como relativas.

1.5 Ejemplos de enlaces habituales

1.5.1 Enlace al inicio del sitio web

```
<a href="/">Inicio</a>
```

Al pulsar el enlace anterior desde cualquier página web, se vuelve directamente a la página de inicio, *home* o página principal del sitio web.

1.5.2 Enlace a un email

```
<a href="mailto:nombre@direccion.com" title="Dirección de email para solicitar más información"> Solicita más información </a>
```

Al pinchar sobre el enlace anterior, se abre automáticamente el programa de correo electrónico del ordenador del usuario y se establece la dirección de envío al valor indicado después de mailto: La sintaxis es la misma que la de un enlace normal, salvo que se cambia el prefijo http://por mailto:

La sintaxis de mailto:permite utilizarlo para otros ejemplos más complejos:

```

<!-- Envío del correo electrónico a varias direcciones a la vez -->
<a href="mailto:nombre@direccion.com,otro_nombre@direccion.com">Solicita más información</a>

<!-- Añadir un "asunto" inicial al correo electrónico -->
<a href="mailto:nombre@direccion.com?subject=Solicitud de más información">Solicita más
información</a>

<!-- Añadir un texto inicial en el cuerpo del correo electrónico -->
<a href="mailto:nombre@direccion.com?body=Estaría interesado en solicitar más información sobre
sus productos">Solicita más información</a>

```

Todas las opciones anteriores se pueden combinar entre sí para realizar ejemplos más avanzados. Aunque el uso de mailto: puede parecer una ventaja, **su uso está desaconsejado**. Si se incluye una dirección de correo electrónico directamente en una página web, es muy probable que en poco tiempo esa dirección de email se encuentre llena de correo electrónico basura o "**spam**", ya que existen programas automáticos encargados de rastrear sistemáticamente todas las páginas web de Internet para encontrar direcciones de correo electrónico válidas.

La forma de mostrar las direcciones de correo electrónico en las páginas web consiste en incluir la dirección en una imagen o indicarla de forma que solamente los usuarios puedan entenderlo:

```

<p>La dirección de correo es <strong>nombre (arroba) direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre_arroba_direccion.com</strong></p>
<p>La dirección de correo es <strong>nombreQUITAESTO@direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre(ARROBA)direccion.com</strong></p>
<p>La dirección de correo es <strong>nombre @ direccion . com</strong></p>

```

1.5.3 Enlace a un archivo FTP

Para enlazar un archivo almacenado en un servidor FTP, la parte del protocolo de la URL debe cambiar de http://a ftp://:

```

<a href="ftp://ftp.ejemplo.com/ruta/archivo.zip" title="Archivo comprimido de los contenidos">
Descarga un ZIP con todos los contenidos
</a>

```

1.5.4 Enlazar varias hojas de estilos CSS

```

<link rel="stylesheet" href="/css/comun.css" />
<link rel="stylesheet" href="/css/secciones.css" />

```

1.5.5 Enlazar hojas de estilos CSS para diferentes medios

```

<link rel="stylesheet" href="/css/comun.css" media="screen, projection" />
<link rel="stylesheet" href="/css/impresora.css" media="print"/>
<link rel="stylesheet" href="/css/movil.css" media="handheld" />

```


1.5.6 Enlazar el favicon

El *favicon* o **icono** para favoritos es el pequeño icono que muestran las páginas en varias partes del navegador. Dependiendo del navegador que se utilice, este icono se muestra en la barra de direcciones, en la barra de título del navegador y/o en el menú de favoritos/marcadores.

```
<link rel="shortcut icon" href="/favicon.ico" type="image/ico" />
```

Aunque en principio la imagen debería ser de tipo .ICO (formato gráfico de los iconos), algunos navegadores soportan favicons en otros formatos gráficos más habituales (como por ejemplo .PNG).

1.5.7 Enlazar un archivo RSS

```
<link rel="alternate" type="application/rss+xml" title="Resumen de todos los artículos del blog" ref="/feed.xml" />
```

1.5.8 Enlazar hojas de estilos, favicon y RSS

En **una misma página se pueden incluir varias etiquetas <link>**, por lo que es habitual que las páginas enlacen hojas de estilos, favicon y archivos RSS de forma conjunta en el encabezado HTML.

1.5.9 Indicar que existe una versión de la página en otro idioma

```
<head>
<title>English tutorial</title>
<link lang="es" xml:lang="es" title="El tutorial en español" type="text/html" rel="alternate" hreflang="es"
href="http://www.ejemplo.com/tutorial/espanol.html"
/>
</head>
```

1.5.10 Indicar que existe una versión de la página preparada para imprimir

```
<head>
<link media="print" title="El tutorial en PDF" type="application/pdf" rel="alternate"
href="http://www.ejemplo.com/tutorial/documento.pdf" />
</head>
```

1.5.11 Indicar que existe una página que es índice de la página actual

```
<head>
<title>Tutorial – Capítulo 5</title>
<link rel="start" title="El índice del tutorial" type="text/html"
href="http://www.ejemplo.com/tutorial/indice.html" />
</head>
```

2. LISTAS

En ocasiones, es posible agrupar determinadas palabras o frases en un conjunto de elementos que tienen más significado de forma conjunta.

El menú de navegación de un sitio web por ejemplo está formado por un grupo de palabras. Aunque cada palabra por separado tiene sentido, de forma conjunta constituyen el menú de navegación de la página, por lo que su significado conjunto es mayor que por separado.

El lenguaje HTML define tres tipos diferentes de listas para agrupar los elementos: listas no ordenadas, listas ordenadas y listas de definición.

2.1 Listas no ordenadas

Las listas no ordenadas son las más sencillas y las que más se utilizan. Una lista no ordenada es un conjunto de elementos relacionados entre sí pero para los que no se indica un orden o secuencia determinados. La etiqueta `` encierra todos los elementos de la lista y la etiqueta `` cada uno de sus elementos.

Etiqueta	<code></code>
Atributos comunes	básicos , internacionalización y eventos
Tipo de elemento	Bloque
Descripción	Se emplea para definir listas no ordenadas

Etiqueta	<code></code>
Atributos comunes	básicos , internacionalización y eventos
Tipo de elemento	Bloque
Descripción	Se emplea para definir los elementos de las listas (ordenadas y no ordenadas)

El siguiente código HTML muestra un ejemplo sencillo de lista no ordenada:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de etiqueta UL</title>
</head>
<body>

  <h1>Menú</h1>

  <ul>
    <li>Inicio</li>
    <li>Noticias</li>
    <li>Artículos</li>
    <li>Contacto</li>
  </ul>

</body>
</html>
```



Figura 2.1 Ejemplo de uso de la etiqueta ul

El navegador por defecto muestra los elementos de la lista tabulados y con una pequeña viñeta formada por un círculo negro. Como ya se sabe, el aspecto con el que se muestran los elementos de las listas se puede modificar mediante las hojas de estilos CSS.

2.2 Listas ordenadas

Las listas ordenadas son casi idénticas a las listas no ordenadas, salvo que en este caso los elementos relacionados se muestran siguiendo un orden determinado. Cuando se crea por ejemplo una lista con las instrucciones de un producto, es importante el orden en el que se realiza cada paso. Cuando se muestra un índice o tabla de contenidos en un libro, es importante el orden de cada elemento del índice.

En todos estos casos, la lista más adecuada es la lista ordenada, que se define mediante la etiqueta ``. Los elementos de la lista se definen mediante la etiqueta ``, la misma que se utiliza en las listas no ordenadas.

Etiqueta	<code></code>
Atributos comunes	básicos , internacionalización y eventos
Tipo de elemento	Bloque
Descripción	Se emplea para definir listas ordenadas

El siguiente código HTML muestra un ejemplo sencillo de lista ordenada:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de etiqueta OL</title>
</head>
<body>

  <h1>Instrucciones</h1>

  <ol>
    <li>Enchufar correctamente</li>
    <li>Comprobar conexiones</li>
    <li>Encender el aparato</li>
  </ol>

</body>
</html>
```

Instrucciones

1. Enchufar correctamente
2. Comprobar conexiones
3. Encender el aparato

Figura 2.2 Ejemplo de uso de la etiqueta ol

El navegador muestra la lista de forma muy parecida a las listas no ordenadas, salvo que en este

caso no se emplean viñetas gráficas en los elementos, sino que se numeran de forma consecutiva. El tipo de numeración empleada también se puede modificar aplicando hojas de estilos CSS a los elementos de la lista.

3. IMÁGENES Y OBJETOS

3.1 Imágenes

Las imágenes son uno de los elementos más importantes de las páginas web. De hecho, prácticamente todas las páginas web contienen alguna imagen y la mayoría incluyen decenas de imágenes. Dentro de las imágenes que se pueden incluir en una página HTML se deben distinguir dos tipos: las imágenes de contenido y las imágenes *de adorno*.

Las imágenes de contenido son las que proporcionan información y complementan la información textual. Las imágenes *de adorno* son las que se utilizan para hacer bordes redondeados, para mostrar pequeños iconos en las listas de elementos, para mostrar fondos de página, etc. Las imágenes de contenido se incluyen directamente en el código HTML mediante la etiqueta `` y las imágenes de adorno no se deberían incluir en el código HTML, sino que deberían emplearse hojas de estilos CSS para mostrarlas.

A continuación, se muestra la definición de la etiqueta ``, utilizada para incluir las imágenes en las páginas HTML:

Etiqueta	<code></code>
Atributos comunes	básicos , internacionalización y eventos
Atributos propios	<ul style="list-style-type: none"> • <code>src = "url"</code> - Indica la URL de la imagen que se muestra • <code>alt = "texto"</code> - Descripción corta de la imagen • <code>longdesc = "url"</code> - Indica una URL en la que puede encontrarse una descripción más detallada de la imagen • <code>name = "texto"</code> - Nombre del elemento imagen • <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar la imagen (no es obligatorio que coincida con la altura original de la imagen) • <code>width = "unidad_de_medida"</code> - Indica la anchura con la que se debe mostrar la imagen (no es obligatorio que coincida con la anchura original de la imagen)
Tipo de elemento	En línea y etiqueta vacía
Descripción	Se emplea para incluir imágenes en los documentos

Los dos atributos requeridos son src y alt. El atributo src es similar al atributo href de los enlaces, ya que establece la URL de la imagen que se va a mostrar en la página.

Las URL indicadas pueden ser absolutas o relativas. El atributo alt permite describir el contenido de la imagen mediante un texto breve. Las descripciones deben tener una longitud inferior a 1024 caracteres y son útiles para las personas y dispositivos discapacitados que no pueden acceder a las imágenes.

Ejemplo sencillo para incluir una imagen:

```

```

Cabe resaltar que es una etiqueta vacía, es decir, **no tiene etiqueta de cierre**.

HTML no impone ninguna restricción sobre el formato gráfico que se puede utilizar en las imágenes, por lo que en principio la etiqueta puede incluir cualquier formato gráfico existente. Sin embargo, si la imagen utiliza un formato poco habitual, todos o algunos navegadores no serán capaces de mostrar esa imagen.

La recomendación es utilizar uno de los tres siguientes formatos gráficos que entienden todos los navegadores modernos: GIF, JPG y PNG.

El atributo longdesc no se utiliza de forma habitual, pero permite indicar la URL en la que se puede encontrar más información sobre la imagen. Como el atributo alt solo permite incluir descripciones de hasta 1024 caracteres, el atributo longdesc se emplea con las imágenes complejas que necesitan mucha información para ser descritas:

```

```

```

```

En el ejemplo anterior, las dos imágenes se encuentran en el mismo directorio del servidor (/imagenes/). Se trata de una estrategia habitual en la mayoría de sitios web: guardar todas las imágenes de contenido en un directorio especial independiente del resto de contenidos HTML. Además, el directorio siempre suele llamarse de la misma manera: "imagenes" o "images" en inglés.

Los atributos width y height se utilizan para indicar la anchura y altura con la que se muestran las imágenes, por lo que son los más contradictorios. Como ya se ha comentado, HTML estructura de forma correcta los contenidos de la página y CSS define el aspecto gráfico con el que se muestran los contenidos. En principio, la anchura y la altura con la que se muestra una imagen es parte de su aspecto gráfico, por lo que debería ser definidos mediante propiedades CSS y no de HTML5.

Sin embargo, en la práctica no es viable establecer la anchura y altura de todas las imágenes de contenidos mediante CSS. Si el sitio web dispone de muchas imágenes, la sobrecarga de estilos diferentes que debería definir CSS sería contraproducente. Por este motivo, los atributos width y height son la excepción a la norma de que el código HTML no haga referencia al aspecto de los contenidos.

```

```

```

```

Si el valor del atributo `width` o `height` se indica mediante un número entero, el navegador supone que hace referencia a la unidad de medida píxel. Por tanto, en el ejemplo anterior, la primera foto se muestra con una anchura de 200píxel y una altura de 350píxel.

También es posible indicar la anchura y altura en forma de porcentaje. En este caso, el porcentaje hace referencia a la altura/anchura del elemento en el que está contenida la imagen. Si la imagen no se encuentra dentro de ningún otro elemento, hace referencia a la anchura/altura total de la página.

```
<div>

</div>
```

El ejemplo anterior mezcla los dos tipos de medidas que se pueden utilizar, para indicar que la foto tiene una anchura igual al 30% de la anchura del elemento `<div>` que la contiene y una altura de 350píxel.

La anchura/altura con la que se muestra una imagen no tiene que coincidir obligatoriamente con la anchura/altura real de la imagen. Sin embargo, cuando estos valores no coinciden, las imágenes se muestran deformadas y el aspecto final es muy desagradable.

Si solamente se establece la altura de la imagen, el navegador calcula la anchura necesaria para que se mantenga la proporción de la imagen. De la misma forma, si solo se establece la anchura de la imagen, el navegador calcula la altura que hace que la imagen se siga viendo con las mismas proporciones.

3.2 Objetos

Además de las imágenes, HTML permite incluir en las páginas web otros elementos mucho más complejos, como *otras páginas web incrustadas o embebidas y elementos multimedia como audio o vídeo*. Muchos navegadores tienen por defecto el desactivar que se interpreten audios o videos de manera directa, dado que gran cantidad de usuarios consideran esto como una molestia.

La etiqueta `<object>` es la que permite "*embeber*" o incluir en las páginas HTML cualquier tipo de contenido complejo. Cabe destacar que con la llegada de HTML5 se pueden definir imágenes mediante la etiqueta ``, embeber páginas HTML con la etiqueta `<iframe>` y también multimedia mediante las etiquetas `<video>` y `<audio>`. Sin embargo, y pese a la gran repercusión que estas últimas etiquetas tienen en los desarrollos web actuales, la web del consorcio W3C indica la preferencia de embeber contenido frente a usar estas etiquetas.

Etiqueta	<code><object></code>
Atributos comunes	básicos , internacionalización y eventos
Atributos propios	<ul style="list-style-type: none"> • <code>data = "url"</code> - Indica la URL de los datos que utiliza el objeto • <code>classid</code>, <code>codebase</code>, <code>codetype</code> - Información específica que depende del tipo de objeto • <code>type</code> - Indica el tipo de contenido de los datos • <code>height = "unidad_de_medida"</code> - Indica la altura con la que se debe mostrar el objeto

	<ul style="list-style-type: none"> width = "unidad_de_medida" - Indica la anchura con la que se debe mostrar el objeto
Tipo de elemento	Bloque y en línea
Descripción	Se emplea para embeber objetos en los documentos

El atributo `data` se emplea para indicar la URL del recurso que se va a incluir. El atributo `type` indica el tipo de contenido de los datos del objeto.

Los posibles valores de `type` están estandarizados y coinciden con los del atributo `type` de la etiqueta `<a>` que se explicó anteriormente.

El propio estándar de HTML incluye ejemplos de uso de esta etiqueta. Incluir un vídeo en formato MPEG:

```
<object data="PlanetaTierra.mpeg" type="application/mpeg" />
```

También se pueden incluir varias versiones alternativas de un mismo contenido. Así, si el navegador no es capaz de interpretar el formato por defecto, puede optar por cualquiera de los otros formatos alternativos:

```
<object title="La Tierra vista desde el espacio" classid="http://www.observer.mars/TheEarth.py">

<!-- Formato alternativo en forma de vídeo -->
<object data="PlanetaTierra.mpeg" type="application/mpeg">
<!-- Otro formato alternativo mediante una imagen GIF -->
<object data="PlanetaTierra.gif" type="image/gif">
<!-- Si el navegador no soporta ningún formato, se muestra el siguiente texto -->
La <strong>Tierra</strong> vista desde el espacio.
</object>
</object>
</object>
```

4. EMBEBIDOS Y MULTIMEDIA

4.1 Embebidos

Para poder embeber contenido multimedia en una página web se utiliza la siguiente etiqueta:

Etiqueta	<code><iframe ></code>
Atributos comunes	básicos, internacionalización y eventos
Atributos propios	<ul style="list-style-type: none"> src = "url" – Especifica la dirección del documento embebido. Width, height name = "text" – Especifica el nombre del iframe.
Tipo de elemento	Elemento de línea
Descripción	Se emplea para embeber contenido en la página web. (Por ejemplo un vídeo de YouTube)

```
<iframe width="420" height="315"
src="https://www.youtube.com/embed/tgbNymZ7vqY?autoplay=1&mute=1">
</iframe>
```

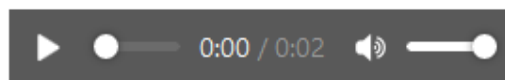
4.2 Multimedia

Entre otras, las novedades que nos trae HTML5 son las etiquetas <vídeo> y <audio>. Estas permiten embeber audiovisuales que tengamos como recursos en nuestro servidor. Lo realmente provechoso, es enlazar estos recursos multimedia desde plataformas externas como YouTube o Vimeo, pues de este modo delegaremos la transmisión del flujo de datos a estas plataformas y descargará la gestión que deba realizar nuestro servidor al ofrecer nuestra página web. Debe adjuntarse el atributo **controls** con el fin que aparezcan los botones (a modo reproductor de contenido) y se pueda pausar/reanudar dichos contenidos, así como ajustar el volumen / silenciar o ajustar a pantalla completa. (Recodad: muchos navegadores tienen por defecto la reproducción automática de contenido multimedia para proporcionar mejor experiencia de usuario).

```
<video width="320" height="240" controls>
<source src="movie.mp4" type="video/mp4">
<source src="movie.ogg" type="video/ogg">
  Tu navegador no soporta este contenido.
</video>
```



```
<audio controls>
<source src="horse.ogg" type="audio/ogg">
<source src="horse.mp3" type="audio/mpeg">
  Tu navegador no soporta este contenido.
</audio>
```



5. TABLAS

Desde sus primeras versiones, HTML incluyó el soporte para crear tablas de datos en las páginas web. Además de ser sencillo, el modelo definido por HTML es muy flexible y bastante completo.

Las tablas en HTML utilizan los mismos conceptos de filas, columnas, cabeceras y títulos que los que se utilizan en cualquier otro entorno de publicación de documentos:

Cursos de diseño gráfico				
Nombre		Horas	Plazas	Horario
Introducción a XHTML		20	20	09:00 – 13:00
CSS avanzado		40	15	16:00 – 20:00
Taller de usabilidad		40	10	16:00 – 20:00
Introducción a AJAX		60	20	08:30 – 12:30

Figura 3.2.6.1 Partes que componen una tabla compleja

Las tablas de HTML pueden contener elementos simples, agrupaciones de filas y de columnas, cabeceras y pies de tabla, subdivisiones, cabeceras múltiples y otros elementos complejos.

A pesar de que las tablas HTML son fáciles de comprender y utilizar, son uno de los elementos más polémicos de HTML.

Aunque parezca obvio, las tablas se deben utilizar para mostrar información tabular. Queda fuera de lugar utilizar las tablas en HTML5 para definir el layout (o estructura) de las páginas web. Si bien esta fue una técnica empleada en el pasado (y por fortuna obsoleta), para conseguir una armónica distribución de los elementos, en la actualidad se ha de recurrir a CSS Flex o Grid.

5.1 Tablas básicas

Las tablas más sencillas de HTML se definen con tres etiquetas: `<table>` para crear la tabla, `<tr>` para crear cada fila y `<td>` para crear cada columna.

A continuación, se muestra el código HTML de una tabla sencilla y su correspondiente visualización en el navegador:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de tabla sencilla</title>
  </head>
  <body>

    <h1>Listado de cursos</h1>
    <table>
      <tr>
        <th>Curso</th>
        <th>Horas</th>
        <th>Horario</th>
      </tr>

      <tr>
        <td>CSS</td>
        <td>20</td>
        <td>16:00 - 20:00</td>
      </tr>

      <tr>
        <td>HTML</td>
        <td>20</td>
        <td>16:00 - 20:00</td>
      </tr>
    </table>
```

Listado de cursos

Curso	Horas	Horario
CSS	20	16:00 - 20:00
HTML	20	16:00 - 20:00

```
</body>
</html>
```

La etiqueta `<table>` encierra todas las filas y columnas de la tabla. Las etiquetas `<tr>` (del inglés *"table row"*) definen cada fila de la tabla y encierran todas las columnas. Por último, la etiqueta `<td>` (del inglés *"table data cell"*) define cada una de las columnas de las filas, aunque realmente HTML no define columnas sino *celdas de datos*.

Al definir una tabla, se debe pensar en primer lugar en las filas que la forman y a continuación en las columnas. El motivo es que HTML procesa primero las filas y por eso las etiquetas `<tr>` aparecen antes que las etiquetas `<td>`.

Etiqueta	<code><table></code>
Atributos comunes	básicos , internacionalización y eventos
Atributos propios	<code>summary = "texto"</code> - Permite describir el contenido de la tabla (lo utilizan los buscadores y las personas discapacitadas)
Tipo de elemento	Bloque
Descripción	Se emplea para definir tablas de datos

Etiqueta	<code><tr></code>
Atributos comunes	básicos , internacionalización y eventos
Tipo de elemento	Bloque
Descripción	Se emplea para definir cada fila de las tablas de datos

Etiqueta	<code><td></code>
Atributos comunes	básicos , internacionalización y eventos
Atributos propios	<ul style="list-style-type: none"> <code>abbr = "texto"</code> - Permite describir el contenido de la celda (empleado sobre todo en los navegadores utilizados por personas discapacitadas) <code>headers = "lista_de_id"</code> - Indica las celdas que actúan como cabeceras para esta celda (los títulos de las columnas y filas). Se indica como una lista de valores del atributo <code>"id"</code> de celdas <code>scope = "col, row, colgroup, rowgroup"</code> - Indica las celdas para las que esta celda será su cabecera. Ej: <code>scope="col"</code> indica que esta celda es la cabecera de todas las demás celdas que están en la misma columna <code>colspan = "numero"</code> - Número de columnas que ocupa esta celda <code>rowspan = "numero"</code> - Número de filas que ocupa esta celda

Tipo de elemento	Bloque
Descripción	Se emplea para definir cada una de las celdas que forman las filas de una tabla, es decir, las columnas de la tabla

De todos los atributos disponibles para las celdas, los más utilizados son `rowspan` y `colspan`, que se emplean para construir tablas complejas como las que se ven más adelante. Entre los demás atributos, solo se utiliza de forma habitual el atributo `scope`, sobre todo con las celdas de cabecera que se ven a continuación.

Normalmente, algunas de las celdas de la tabla se utilizan como cabecera de las demás celdas de la fila o de la columna. En este caso, HTML define la etiqueta `<th>` (del inglés *"table header cell"*) para indicar que una celda es cabecera de otras celdas.

Etiqueta	<code><th></code>
Atributos comunes	básicos , internacionalización y eventos
Atributos propios	<ul style="list-style-type: none"> <code>abbr = "texto"</code> - Permite describir el contenido de la celda (empleado sobre todo en los navegadores utilizados por personas discapacitadas) <code>headers = "lista_de_id"</code> - Indica las celdas que actúan como cabeceras para esta celda (los títulos de las columnas y filas). Se indica como una lista de ID de celdas <code>scope = "col, row, colgroup, rowgroup"</code> - Indica las celdas para las que esta celda será su cabecera. Ej: <code>scope="col"</code> indica que esta celda es la cabecera de todas las demás celdas que están en la misma columna <code>colspan = "numero"</code> - Número de columnas que ocupa esta celda
Tipo de elemento	Bloque
Descripción	Se emplea para definir las celdas que son cabecera de una fila o de una columna de la tabla

Los atributos de la etiqueta `<th>` son idénticos que los atributos definidos para la etiqueta `<td>`. En este caso, el atributo más utilizado es `scope`, que permite indicar si la celda es cabecera de la fila o de la columna (`<th scope="row">` y `<th scope="col">` respectivamente).

Por otra parte, HTML define la etiqueta `<caption>` para establecer la leyenda o título de una tabla. La etiqueta debe colocarse inmediatamente después de la etiqueta `<table>` y cada tabla solo puede incluir una etiqueta `<caption>`.

Etiqueta	<code><caption></code>
Atributos comunes	básicos , internacionalización y eventos
Tipo de elemento	En línea
Descripción	caption

Las tablas complejas suelen disponer de una estructura irregular que junta varias columnas para formar una columna ancha o une varias filas para formar una fila más alta que las demás. Para fusionar filas o columnas, se utilizan los atributos `rowspan` y `colspan` respectivamente.

La siguiente imagen muestra una tabla compleja que ha fusionado dos columnas simples para formar una columna más ancha:

Fusión de columnas	
A	
B	C

Figura 3.2.6.5 Ejemplo sencillo de fusión de columnas

Para obtener una tabla como la de la imagen anterior, se debe utilizar el siguiente código:

```
<table>
<tr>
  <td colspan="2">A</td>
</tr>
<tr>
  <td>B</td>
  <td>C</td>
</tr>
</table>
```

La primera fila de la tabla está formada solo por una columna, mientras que la segunda fila está formada por dos columnas. **En principio, podría pensarse en utilizar el siguiente código HTML para definir la tabla:**

```
<table>
<tr>
  <td>A</td>
</tr>
<tr>
  <td>B</td>
  <td>C</td>
</tr>
</table>
```

Sin embargo, **si se utiliza el código anterior, el navegador visualiza de forma incorrecta la tabla**, ya que las tablas en HTML deben disponer de una estructura regular. En otras palabras, **todas las filas de una tabla HTML deben tener el mismo número de columnas**. Por lo tanto, si se quieren mostrar

menos columnas en una fila, se fusionan mediante el atributo colspan, que indica el número de columnas simples que va a ocupar una determinada celda.

En el ejemplo anterior, la celda de la primera fila debe ocupar el espacio de dos columnas simples, por lo que el código HTML debe ser `<td colspan="2">A</td>`.

De forma equivalente, si se quiere diseñar una tabla HTML que fusiona filas, el código HTML que se debe utilizar para obtener la tabla que muestra la siguiente imagen es:

```
<table>
<tr>
  <td>A</td>
  <td rowspan="2">B</td>
</tr>

<tr>
  <td>C</td>
</tr>
</table>
```

Fusión de filas	
A	B
C	

Figura 6.2 Ejemplo sencillo de fusión de filas

De forma análoga a la fusión de columnas del ejemplo anterior, la fusión de filas debe indicarse de forma especial. Como las tablas HTML tienen que ser regulares, todas las columnas deben tener el mismo número de filas. Así, si en el ejemplo anterior se utilizara el siguiente código:

```
<table>
<tr>
  <td>A</td>
  <td>B</td>
</tr>

<tr>
  <td>C</td>
</tr>
</table>
```

La tabla anterior no se visualizaría correctamente. Como la segunda columna de la tabla ocupa el espacio de las dos filas, el código HTML debe indicar claramente que esa celda va a ocupar dos filas, de manera que todas las columnas de la tabla cuenten con el mismo número de filas.

Utilizando los atributos rowspan y colspan, es posible diseñar tablas tan complejas como las que se muestran en los siguientes ejemplos.

```
<!DOCTYPE html>
<html>
<head><title>Ejemplo de columnas
fusionadas</title></head>
<body>

<table>
  <tr>
    <td colspan="3">A</td>
    <td>B</td>
  </tr>

  <tr>
    <td>C</td>
    <td colspan="2">D</td>
    <td>E</td>
  </tr>

  <tr>
    <td colspan="4">F</td>
  </tr>

  <tr>
    <td>G</td>
    <td>H</td>
    <td>I</td>
    <td>J</td>
  </tr>
</table>

</body>
</html>
```

A			B
C	D		E
F			
G	H	I	J

Figura 6.3 Ejemplo sencillo de fusión de columnas

```

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

<h1>Fusión de filas</h1>

<table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td rowspan="3">C</td>
    <td>D</td>
  </tr>
  <tr>
    <td rowspan="2">E</td>
    <td>F</td>
    <td rowspan="3">G</td>
  </tr>
  <tr>
    <td>H</td>
  </tr>
  <tr>
    <td>I</td>
    <td>J</td>
    <td>K</td>
  </tr>
</table>

</body>
</html>
    
```

A	B	C	D
E	F		G
	H		
I	J	K	

Figura 6.4 Ejemplo complejo de fusión de filas

```

<style>
  td,
  th {
    border: 1px solid black;
    padding: 0.2em;
    text-align: center;
  }

  table {
    border-collapse: collapse;
  }
</style>
    
```

Estilos CSS a añadir en <head>

5.2 Tablas avanzadas

Algunas tablas complejas están formadas por más elementos que filas y celdas de datos. Así, es común que las tablas más avanzadas dispongan de una sección de cabecera, una sección de pie y varias secciones de datos. Además, también es posible agrupar varias columnas de forma lógica para poder aplicar estilos similares a un determinado grupo de columnas.

Un ejemplo clásico de tablas avanzadas es el de las tablas utilizadas en contabilidad, como por ejemplo la tabla que muestra el balance de una empresa:

CONSOLIDATED BALANCE SHEET	2008			
	March	June	September	December
Non-current assets	87.249	87.126	90.426	91.269
Intangible assets	21.810	21.145	20.986	20.758
Goodwill	17.914	19.660	21.828	21.739
Property, plant and equipment and Investment property	33.245	32.332	33.428	33.888
Long-term financial assets and other non-current assets	5.723	5.687	5.981	6.183
Deferred tax assets	8.557	8.303	8.202	8.702
Current assets	18.042	17.979	19.128	17.713
Inventories	1.154	1.134	1.052	1.012
Trade and other receivables	9.244	9.495	9.709	9.666
Current tax receivable	1.288	1.565	1.468	1.555
Short-term financial investments	1.877	1.803	1.788	1.679
Cash and cash equivalents	4.468	3.557	5.101	3.792
Non-current assets classified as held for sale	11	425	9	9
Total Assets = Total Equity and Liabilities	105.291	105.106	109.554	108.982

Figura 6.5 Ejemplo parcial de una tabla compleja correspondiente al balance de una empresa

Las partes que componen las tablas complejas se definen mediante las etiquetas <thead>, <tbody> y <tfoot>. Cabe citar que estas etiquetas cumplen una función meramente semántica (de significado) y por tanto no crean ningún efecto visual allá donde son declaradas. La cabecera de la tabla se define con la etiqueta <thead>, el pie de la tabla se define mediante <tfoot> y cada sección de datos se define con una etiqueta <tbody>.

Etiqueta	<thead><tbody><tfoot>
Atributos comunes	básicos , internacionalización y eventos
Tipo de elemento	Bloque
Descripción	Se emplean para agrupar varias filas en una cabecera (thead) un pie (tfoot) o una sección (tbody) de una tabla

Cada tabla puede contener solamente una cabecera y un pie, pero puede incluir un número ilimitado de secciones. Si se define una cabecera y/o un pie, las etiquetas <thead> y/o <tfoot> deben colocarse inmediatamente antes que cualquier etiqueta <tbody>.

La siguiente imagen muestra una tabla avanzada con cabecera, pie y una sección de datos:

Análisis de ventas anuales				
AÑO	Expansión de ventas			
	Producto A	Producto B	Producto C	Producto D
N-2	-	-	-	-
N-1	4	4	7	3
N	5	7	6	2
TOTAL	9	11	13	5

Un ejemplo de código HTML necesario para crear una tabla de la imagen anterior hace uso de las etiquetas <thead>, <tbody> y <tfoot>:

```
<!DOCTYPE html>
<html>

<head>
  <title>Ejemplo de tabla avanzada</title>
  <style> /* Reglas CSS */
    td,th{ border: 1px solid black;
      padding:0.2em; text-align: center;}
    table { border-collapse: collapse;}
  </style>
</head>

<body>
  <table summary="Análisis de ventas anuales">
    <caption>Análisis de ventas anuales</caption>
    <thead>
      <tr>
        <th rowspan="2" scope="col">AÑO</th>
        <th colspan="4" scope="col">Expansión de ventas</th>
      </tr>
      <tr>
        <th scope="col">Producto A</th>
        <th scope="col">Producto B</th>
        <th scope="col">Producto C</th>
        <th scope="col">Producto D</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row">N-2</th>
        <td>-</td>
        <td>-</td>
        <td>-</td>
        <td>-</td>
      </tr>
```

```

        <td>-</td>
    </tr>
    <tr>
        <th scope="row">N-1</th>
        <td>4</td>
        <td>4</td>
        <td>7</td>
        <td>3</td>
    </tr>
    <tr>
        <th scope="row">N</th>
        <td>5</td>
        <td>7</td>
        <td>6</td>
        <td>2</td>
    </tr>
</tbody>
<tfoot>
    <tr>
        <th rowspan="1" scope="col">TOTAL</th>
        <th scope="col">9</th>
        <th scope="col">11</th>
        <th scope="col">13</th>
        <th scope="col">5</th>
    </tr>
</tfoot>
</table>
</body>
</html>

```

La etiqueta <tbody> permite realizar agrupaciones de filas, pero en ocasiones se necesitan agrupar columnas. Aunque su uso no es muy común, HTML define dos etiquetas similares para agrupar columnas: <col> y <colgroup>.

La etiqueta <col> se utiliza para asignar los mismos atributos a varias columnas de forma simultánea. De esta forma, la etiqueta <col> no agrupa columnas, sino que solo asigna atributos comunes a varias columnas.

La siguiente imagen muestra una tabla que hace uso de la etiqueta <col>:

Análisis de ventas anuales				
AÑO	Producto A	Producto B	Producto C	Producto D
N-3	-	-	-	-
N-2	3	5	8	4
N-1	4	4	7	3
N	5	7	6	2

Figura 6.7 Ejemplo de tabla avanzada que usa la etiqueta col

El código HTML necesario para crear la tabla anterior se muestra a continuación (**OJO: Falta declarar las reglas CSS incluidas en el ejemplo anterior**):

```
<table summary="Análisis de ventas anuales">
  <caption>Análisis de ventas anuales</caption>
  <col style="width:10%;" />
  <col style="width:30%;" />

  <thead>
    <tr>
      <th scope="col">AÑO</th>
      <th scope="col">Producto A</th>
      <th scope="col">Producto B</th>
      <th scope="col">Producto C</th>
      <th scope="col">Producto D</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <th scope="row">N-3</th><td>-</td><td>-</td><td>-</td><td>-</td>
    </tr>
    <tr>
      <th scope="row">N-2</th><td>3</td><td>5</td><td>8</td><td>4</td>
    </tr>
    <tr>
      <th scope="row">N-1</th><td>4</td><td>4</td><td>7</td><td>3</td>
    </tr>
    <tr>
      <th scope="row">N</th><td>5</td><td>7</td><td>6</td><td>2</td>
    </tr>
  </tbody>
</table>
```

Por otra parte, la etiqueta `<colgroup>` se emplea para agrupar de forma estructural varias columnas de la tabla. La forma habitual de indicar el número de columnas que abarca la agrupación es utilizar el atributo `span`, que establece el número de columnas de cada agrupación.

La siguiente imagen muestra una tabla avanzada con una agrupación de columnas realizada con la etiqueta `<colgroup>`:

Figura 6.8 Ejemplo de tabla avanzada que usa la etiqueta `colgroup`

El código HTML necesario para crear la tabla anterior se muestra a continuación:

AÑO	Producto A	Producto B	Producto C	Producto D
N-1	4	4	7	3
N	5	7	6	2

```
<table summary="Análisis de ventas anuales">
  <caption>Análisis de ventas anuales</caption>

  <colgroup>
    <col style="background-color: red;">
    <col span="3" style="background-color: yellow;">
    <col style="background-color: lightgreen;">
  </colgroup>

  <thead>
    <tr>
      <th scope="col">AÑO</th>
      <th scope="col">Producto A</th>
      <th scope="col">Producto B</th>
      <th scope="col">Producto C</th>
      <th scope="col">Producto D</th>
    </tr>
  </thead>

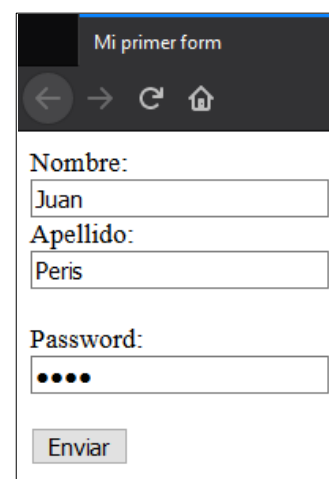
  <tbody>
    <tr>
      <th scope="row">N-1</th>
      <td>4</td>
      <td>4</td>
      <td>7</td>
      <td>3</td>
    </tr>
    <tr>
      <th scope="row">N</th>
      <td>5</td>
      <td>7</td>
      <td>6</td>
      <td>2</td>
    </tr>
  </tbody>
</table>
```

El uso de las etiquetas `<col>` y `<colgroup>` no está muy extendido, debido a que la mayoría de navegadores no soportan muchas de sus funcionalidades.

7. FORMULARIOS

Los formularios ocupan un lugar de suma importancia en el desarrollo web dado que son los elementos que nos permiten realizar la recogida de datos del usuario que se requiera en cada caso. Los datos introducidos por el usuario serán enviados a un proceso servidor que realizará un tratamiento de los mismos. Los formularios envían los datos especificando la URL destino mediante el atributo `action` e indicando el tipo de método GET/POST, dependiendo si estos van a participar en una consulta o en la generación de nuevos recursos en el lado servidor.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Mi primer form</title>
</head>
<body>
  <form action="/registroUsuario.php" method="POST">
    <label for="nombre">Nombre:</label><br>
    <input type="text" id="nombre" name="fname"><br>
    <label for="apellido">Apellido:</label><br>
    <input type="text" id="apellido" name="lname"><br><br>
    <label for="password">Password:</label><br>
    <input type="password" id="password" name="passwd"><br><br>
    <input type="submit" value="Enviar">
  </form>
</body>
</html>
```



HTML5 pone a disposición del desarrollador unas herramientas básicas de filtrado de la información sobre las cajas de texto empleadas a modo de atributos y propiedades para efectuar una validación inicial, la cual en ningún caso substituirá la propia efectuada mediante JavaScript o el lenguaje backend del lado servidor. La validación puede efectuarse mediante la adición de atributos como `required` (campo requerido) o `checked` (campo marcado) y con la definición de expresiones regulares que permitan un detalle mas exhaustivo respecto al texto introducido por el usuario. Esta validación resultará en caso que el usuario no introduzca los datos como esperábamos en un mensaje automático visible, que guiará al mismo en como debe cumplimentar el formulario que atiende.

Existe una larga lista de atributos de interés para los elementos `<input>`. Algunos son: `checked`, `disabled`, `minlength/maxlength`, `placeholder`, `readonly`, `required`, `size`, `pattern`.

```

<!DOCTYPE html>
<html>
<body>

<h1>Introduce tu matrícula</h1>

<form action="/action_page.php">
  <label for="matricula">Matrícula:</label>
  <input type="text" id="country_code"
        name="country_code"
        pattern="[1-9]{4}[A-Z]{3}"
        title="Matrícula DGT España"><br><br>
  <input type="submit">
</form>

<p><strong>Nota:</strong> El atributo de patrón no
esta soportado en Safari 10 (o anteriores)</p>

</body>
</html>

```

Introduce tu matrícula

Matrícula:

Ajústese al formato solicitado: Matrícula DGT España.

Nota: El atributo de patrón no esta soportado en Safari 10 (o anteriores)

Submitted Form Data

Your input was received as:

`country_code=1234ABC`

The server has processed your input and returned this answer.

Los formularios se apoyan en su versión mas básica en elementos de cajas de texto y botones que permitan realizar un envío de datos al proceso servidor. Las cajas de texto utilizan las etiquetas `<input>` de tipo texto, password o email. La acción de envío de datos del formulario suele definirse mediante un `<input>` de tipo "submit". Pueden añadirse etiquetas basadas en texto mediante la etiqueta `<label>`, siendo vinculadas estas a posteriori a las cajas de texto que les corresponda.

Os mostramos un subconjunto de los tipos `<input>` de mayor interés. Aconsejamos probarlos de tal manera que algunos de ellos si bien a priori puede que no les veáis la utilidad, en un futuro (esperemos próximo) os facilite la labor de interacción con el usuario y mejoréis su experiencia de navegación.

```

<input type="button" value="PressMe">
<input type="checkbox">
<input type="color">
<input type="date">
<input type="datetime-local">
<input type="email">
<input type="file">
<input type="hidden">
<input type="number">
<input type="password">
<input type="radio">
<input type="range">
<input type="reset">
<input type="search">
<input type="submit">
<input type="tel">
<input type="text"> (default value)
<input type="time">
<input type="url">
    
```

Recordamos una vez mas, que el navegador usa unos estilos visuales por defecto para mostrar ciertos elementos como los `<input>`, otorgando a los mismos una apariencia de cajas con bordes y sombreado si adquieren el foco o pasa el puntero del ratón por encima. Inclusive el efecto de enmascarado de los caracteres introducidos en un `<input>` de tipo “password” son simplemente eso, un efecto visual y en ningún caso una medida de seguridad estricta.

Adicionalmente existen etiquetas que refuerzan el sentido semántico de cada una de las partes del formulario, si este comprende varias secciones. Es el caso de las etiquetas `<fieldset>`, `<legend>` que distribuyen diferentes secciones e incluso permiten definir leyendas de texto para guiar al usuario. En lo visual, todo es reemplazable como hemos venido anunciando anteriormente.

8. ELEMENTOS DE BLOQUE Y DE LÍNEA

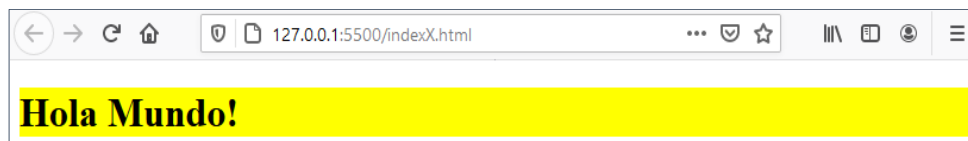
Debe quedar claro que todo elemento HTML5 ocupa un espacio en forma de caja rectangular, sea el

elemento que sea. Según hemos ido presentando en esta y la anterior unidad didáctica, la naturaleza de estos elementos no es la misma si los clasificamos en elementos en bloque y elementos de línea.

Los elementos en bloque se caracterizan por las siguientes propiedades:

- Intentan ocupar todo el ancho disponible del elemento que los contiene.
- Permiten definir un alto y ancho.
- Desplazan al siguiente elemento definido en el documento HTML a la siguiente línea, es decir provocan un salto de línea.

Ejemplo: un encabezado de primer nivel (<h1>) es un elemento de bloque que ocupa todo el espacio horizontal disponible y que fuerza a que el siguiente elemento definido se sitúe en una nueva línea.



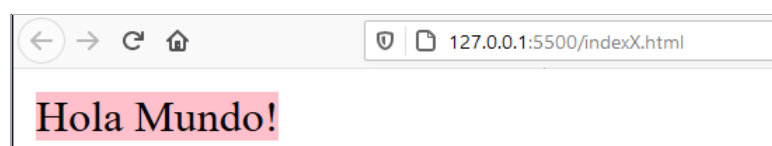
Los **elementos de bloque** de HTML5 son:

<address>	<fieldset>	
<article>	<figcaption>	<p>
<aside>	<figure>	<pre>
<blockquote>	<footer>	<section>
<canvas>	<form>	<table>
<dd>	<hr>	<tfoot>
<div>		
<dl>	<main>	<video>
<dt>	<nav>	<header>
<h1>-<h6>		

Por contra, los elementos en línea se caracterizan por:

- Ocupar únicamente el espacio necesario en base a su contenido
- Disponer de los elementos de forma contigua hasta que no quepan en la misma línea. Será entonces cuando se produzca un salto de línea.
- No poder definir sus dimensiones (a excepción de las imágenes).

Muy generalmente se asocian la mayoría de los elementos en línea al tratamiento de texto en un fichero HTML. Ejemplo: un elemento definido para poder efectuar un marcado de texto con un fondo rosa.



Cabe resaltar que un elemento en línea no puede contener a un elemento en bloque (a la inversa si es posible).

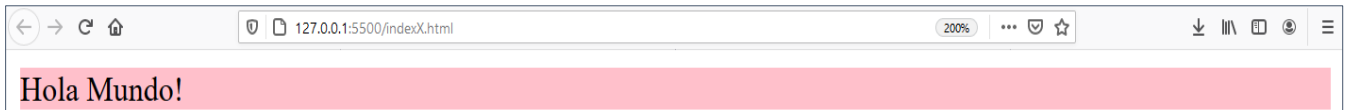
Los **elementos de línea** de HTML5 son:

<a>		<script>
<abbr>	<i>	<select>
<acronym>		<small>
	<input>	
<big>	<var>	
 	<object>	<sub>
<button>	<output>	<sup>
<cite>	<q>	<textarea>
<code>	<tt>	<time>
<dfn>	<map>	<label>

La propiedad **display** permite alterar la naturaleza de un elemento. Elementos bloque podrían comportarse como elementos en línea y viceversa. Además es posible definir un tercer estado: **inline-block**, el cual se caracteriza por:

- Poder definir alto y ancho.
- Los márgenes y espaciados interiores (o paddings) son respetados.
- No empuja los elementos contiguos a nuevas líneas.

```
<span style="display: inline-block;">Hola Mundo!</span>
```



Sobre la propiedad display, el valor **none** provoca una desaparición del elemento en el renderizado visual que produce el navegador y además, se produce una pérdida del espacio reservado. Este evento puede producir que elementos definidos a continuación ocupen dicho espacio ahora extinto.

9. SEMÁNTICA

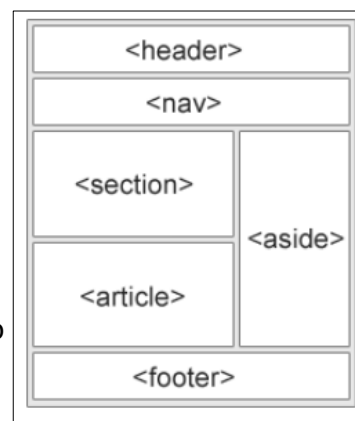
Los elementos semánticos son aquellos que hacen una descripción precisa al navegador que los interpreta y al propio desarrollador. Ejemplos de elementos no semánticos son: **<div>** y ****. Ambos, son contenedores, uno con comportamiento de elemento de bloque y el otro en línea. Ejemplos de elementos semánticos son: **<form>** **<table>** o **<article>**.

La gran mayoría de las páginas web de nueva creación siguen un patrón de layout (o disposición de elementos) como el que se aprecia en la imagen.

La propuesta de HTML5 consiste en definir un conjunto de elementos semánticos con el objetivo de recrear las áreas específicas comunes de la estructuración web moderna y así permitir la reutilización en aplicaciones, empresas y comunidades.

Estas etiquetas con significado puramente semántico sirven para que motores de búsqueda y técnicas de inteligencia artificial efectúen una serie de operaciones transparentes para el usuario final de la interfaz, pero con un propósito de procesamiento de la estructura HTML del documento y extracción de la información.

La etiqueta **<section>** define una sección para un grupo temático de contenido que usualmente porta un encabezado. Normalmente una página web puede ser dividida en secciones para introducción, contenido y información de contacto.



<article> define espacios independientes y autocontenidos. Estos deben tener sentido por sí mismos, debiendo ser posible su distribución de forma independiente por el resto de la página. (Ejemplos: publicaciones de foros / blog).

<section> y **<article>** pueden anidarse según el criterio del desarrollador tanto en cuanto la estructuración de la información respete una jerarquía lógica según su contenido.

La etiqueta **<header>** representa un espacio de contenido para un conjunto de enlaces navegacionales. Usualmente contiene: encabezados de distintos niveles, un logo o ícono, información de autoría.

El pie de página o **<footer>** define la sección final del documento. Tiende a facilitar información acerca de la autoría, derechos de copia, información de contacto, enlaces de redirección a zonas de inicio, documentos relacionados.

<nav> responde al conjunto principal de enlaces que usualmente corresponde con la barra de navegación.

<aside> define un área que se sitúa a un lado de la página. Su contenido debe estar indirectamente relacionado con el contenido principal **<main>**.

10. BIBLIOGRAFÍA

<http://librosweb.es/libro/xhtml/>
www.w3schools.com
<https://developer.mozilla.org/es/>