

MP_0373.

**Lenguajes de marcas y sistemas de
gestión de información**

**UF2. Lenguajes para la visualización
de la información**

2.2. Etiquetas HTML

Índice

≡	Objetivos	4
≡	Introducción	5
≡	Las etiquetas o tags	7
≡	Los atributos	9
≡	Los comentarios	13
≡	Los caracteres especiales	14
≡	Estructura del código de la página	17
≡	El DOCTYPE	19
≡	HTML5	22
≡	La cabecera del documento <head>	24
≡	El cuerpo de la página <body>	26
≡	Los textos	27
≡	Listas	30
≡	Los enlaces	33
≡	Las tablas	36

≡	Las imágenes	39
≡	Introducción a SVG	41
≡	Etiquetas semánticas	48
≡	Otras etiquetas	51
≡	Creación de un formulario sencillo	52
≡	Campos de formulario	55
≡	Nuevos atributos para campos de formularios HTML5	61
≡	Nuevos tipos de campos HTML5 con validación automática	63
≡	Estructura de un documento HTML	64
≡	Resumen	65

Objetivos

Los objetivos de esta unidad son:

- 1 Identificar las etiquetas y atributos de HTML (*Hyper TextMarkup Language*).
- 2 Estructurar documentos HTML.
- 3 Conocer las etiquetas de contenido: títulos, párrafos, listas.
- 4 Conocer los elementos de formulario y controles HTML5: campos de texto, botones, desplegables.
- 5 Otros elementos de formato y agrupamiento: tablas, capas.

Introducción

HTML (*HyperText Markup Language*) es un lenguaje de etiquetas usado por los diseñadores y programadores para crear páginas web.

Este lenguaje es un estándar reconocido por todo el mundo. Sus normas las define un organismo sin ánimo de lucro llamado **World Wide Web Consortium**

[IR A SITIO WEB](#)

Se ha convertido en el estándar de visualización de la información, siendo la vista de muchas aplicaciones, tanto de las páginas web más tradicionales como de aplicaciones de servidor para servicios empresariales.

Actualmente, los lenguajes de programación más modernos como Java, .NET, Python o PHP disponen de librerías y tienen interiorizado este sistema de representación para mostrar la capa de vista de los programas.

Un documento HTML está formado por su contenido y etiquetas que delimitan sus partes para dar formato.

Empecemos por mostrar un poco de código de una página HTML en su formato más básico:

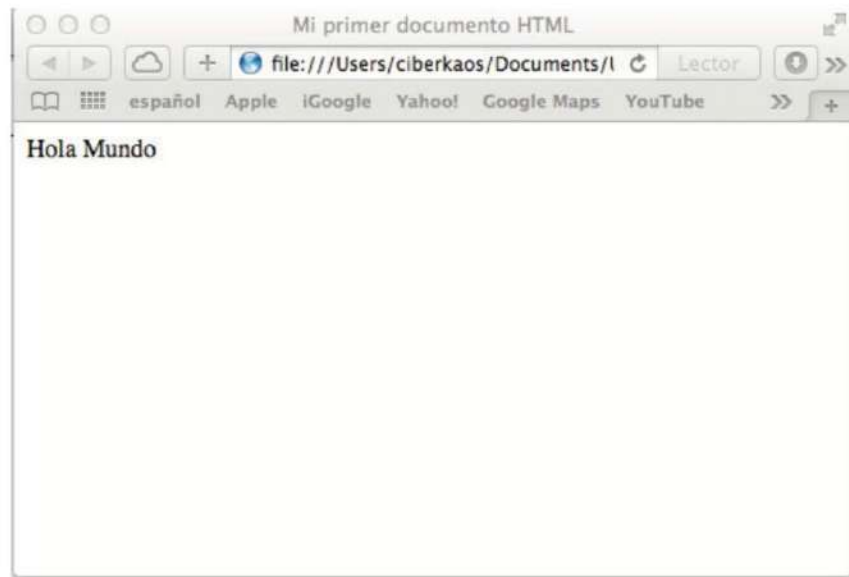
```
<html>

<head>
  <title>Mi primer documento HTML</title>
</head>

<body>
  <p>Hola Mundo</p>
</body>

</html>
```

Como podemos ver, tenemos etiquetas y contenido. En este caso estamos imprimiendo en pantalla el texto encerrado en un párrafo `<p> Hola Mundo`.



HTML viene de las siglas en inglés de *HyperText Markup Language*, en castellano lenguaje de marcas de hipertexto. Se trata de un lenguaje de marcado para la especificación del contenido de las páginas web, desarrollado y regulado por el consorcio W3C.

Las etiquetas o tags

Una **tag** o **etiqueta** es la marca que usa el lenguaje para delimitar a un elemento específico.



Las etiquetas o marcas están definidas por caracteres de escape que indican al software que lee el documento cuándo comienza y acaba el **tag** o etiqueta. En lenguajes de marcado como HTML existen etiquetas o marcas que usan el carácter “<”, el “\” o el “>”.

En HTML y en XML casi todas las marcas indican el comienzo y fin de un elemento.

<p>Este es el texto de párrafo</p>

Etiqueta de apertura

Etiqueta de cierre

Como podemos apreciar, en este tipo de etiquetas la de cierre es igual que la de apertura pero con la barra (/) delante.

Existen etiquetas que al no tener contenido propio, **no tienen su correspondiente cierre**. Como las etiquetas **
** (salto de línea) o **** (imagen).

Los atributos

Los atributos sirven para dotar al elemento de propiedades que tienen que ver con la marca en la que están contenidos.

La forma de representar dichas propiedades es a través de pares nombre/valor separados por un signo de igual (=), normalmente se escriben después de la etiqueta de comienzo de un elemento.

El valor que va asociado al atributo suele ir encerrado entre comillas dobles o simples.

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

Existe una clasificación para los atributos:

Atributos básicos

Se pueden utilizar prácticamente en todas las etiquetas HTML.

Atributos para internacionalización

Los utilizan las páginas que muestran sus contenidos en varios idiomas.

Atributos de eventos

Solo se utilizan en las páginas web dinámicas creadas con JavaScript.

Atributos de foco

Relacionados principalmente con la accesibilidad de los sitios web.

Atributos globales en HTML5

Existen en la actualidad atributos globales a todas las etiquetas en HTML5.

[VER](#)

Con el paso a HTML5 muchos de los atributos han sido eliminados, pues uno de sus objetivos es simplificar el código

Atributos básicos

Los atributos básicos se utilizan en la mayoría de etiquetas HTML y XHTML, aunque adquieren mayor sentido cuando se utilizan hojas de estilo en cascada (CSS):



id = "texto"

Establece un identificador único a cada elemento dentro de una página HTML.



class = "texto"

Establece la clase CSS que se aplica a los estilos de elemento.



style = "texto"

Establece de forma directa los estilos CSS de un elemento.



title = "texto"

Establece el título a un elemento; mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento.

Atributos para internacionalización

Estos atributos se utilizan en aquellas páginas que muestran sus contenidos en varios idiomas y en las que quieran indicar de forma explícita el idioma de sus contenidos:

☐

lang = "código de idioma"

Indica el idioma del elemento mediante un código predefinido.

☐

xml:lang = "código de idioma"

Indica el idioma del elemento mediante un código predefinido.

☐

dir = "ltr|rtl|auto"

Indica la dirección del texto; útil para los idiomas que escriben de derecha a izquierda.

Atributos de evento

Los atributos de evento son "escuchadores" que lanzan una acción cuando se detecta un comportamiento. Cuando hacemos clic en un botón o movemos el ratón, el navegador puede detectarlo y lanzar una función de JavaScript.

Existe un gran número de eventos; puedes ver una lista de estos con su descripción en la [lista de eventos de w3schools](#).

Atributos de foco

Se denomina foco o *focus* cuando un control o elemento del documento ha sido seleccionado. Cuando ese elemento deja de estar seleccionado "pierde el foco" y es el nuevo elemento seleccionado el que se dice que tiene "el foco".



accesskey = "letra"

Establece una tecla de acceso rápido a un elemento HTML.



tabIndex = "número"

Establece la posición del elemento en el orden de tabulación de la página. Su valor debe estar comprendido entre 0 y 32 767.



onfocus, onblur.

Controlan los eventos JavaScript que se ejecutan cuando el elemento obtiene o pierde el foco.

Los comentarios

Los **comentarios** sirven para incluir **información extra que no es interpretada por el navegador** y puede aparecer en cualquier parte de un documento.

Para definirlos en HTML se usa la siguiente etiqueta:

```
<!-- Esto es un comentario -->
```



Los comentarios de HTML no son interpretados por el navegador, por lo que **no aparecen en el documento que ve el usuario**. Pero sí están presentes en el código, de tal modo que el usuario puede visualizarlos si le pide al navegador que muestre el código fuente.

Los caracteres especiales

La **codificación de caracteres** consiste en la **conversión de un carácter** desde el lenguaje natural a un símbolo que pertenece a otro sistema de representación y se realiza mediante el uso de unas normas y reglas.

En este caso serviría para codificar caracteres a través del lenguaje de marcas y su contenido de lenguaje escrito en los distintos alfabetos existentes.

Una de las primeras codificaciones fue el código **ASCII**, que con 8 caracteres codificó 128 del alfabeto latino.

En la confección de documentos HTML o XML se indica en el encabezado el juego de caracteres que queremos que use el navegador para decodificar la página o documento web.

La codificación de caracteres más moderna se realiza en **Unicode** (**UTF-8**, **UTF-16** o **UTF-32**) que es un estándar definido por la **ISO10646**.

Por ejemplo, si queremos usar el UTF-8 en una página HTML podríamos incluir en la **cabecera de nuestro documento** la siguiente directiva:

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

Puedes ampliar información sobre HTML Document Representation.

IR

Cuando se quisieron codificar caracteres especiales o los caracteres de lenguajes de todo el mundo y representarlos en HTML, se crearon unas codificaciones que hacían que el navegador o el intérprete del documento los reconociera como caracteres especiales.

Códigos ASCII

Puedes consultar la tabla de códigos ASCII para caracteres y caracteres especiales.

IR

De este modo podemos insertar cualquier carácter de la tabla en un documento, solucionando problemas de incompatibilidad de comandos o de internacionalización de caracteres.

```
<!-- Por número -->  
<p>&#169; Copy</p>  
  
<!-- Por nombre -->  
<p>&copy; Copy</p>
```

Ejemplo de presentación del símbolo de copyright "©" en HTML.

El uso de estos caracteres especiales puede resultarnos útil en muchas ocasiones, pero sobre todo cuando queramos escribir caracteres reservados para el propio lenguaje de marcas, como "<" y ">", por lo que los más habituales son:

- `<` para <.
- `>` para >.
- `&` para &.
- `"` para “.

Estos códigos empiezan siempre con el signo "&" y acaban con “;”.

Otro caso en el que nos encontraremos habitualmente es el del idioma. Aunque ya existe el atributo *charset* de HTML, en ocasiones nos encontraremos con problemas de interpretación de caracteres, propios de un idioma, como en nuestro caso la letra ñ. Por eso es importante recordar:

- | | |
|-------------------------------|----------------------------------|
| • <code>&ntilde;</code> ñ | • <code>&uacute;</code> ú |
| • <code>&Ntilde;</code> Ñ | • <code>&Aacute;</code> Á |
| • <code>&aacute;</code> á | • <code>&Eacute;</code> É |
| • <code>&eacute;</code> é | • <code>&Iacute;</code> Í |
| • <code>&iacute;</code> í | • <code>&Oacute;</code> Ó |
| • <code>&oacute;</code> ó | • <code>&Uacute;</code> Ú |
| | • <code>&euro;</code> € euro |

```
<p>Este párrafo contiene caracteres acentuados</p>
```

Ejemplo.

Estructura del código de la página

La **estructura de un documento** permite identificar sus partes de manera que podamos ubicar en cada zona lo referente a esta y construyamos un documento válido.



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <title>No tittle</title>
    <meta name="generator" content="Amaya, see http://www.w3.org/Amaya/" />
  </head>
  <body>
    <h1>Mi primer encabezado</h1>
    <p>Mi primer párrafo</p>
  </body>
</html>
```

La etiqueta `<!DOCTYPE ...>` es de uso obligado, ya que informa al navegador del tipo de documento (si es HTML o XHTML) y cuál es la versión. Recuerda que existen varios tipos de documentos y diferentes versiones.

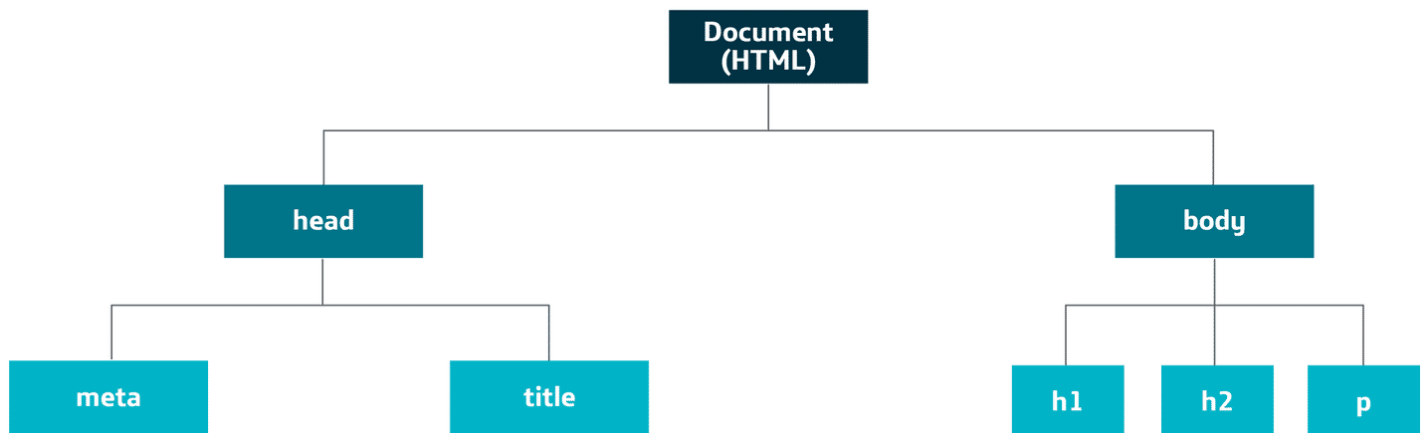
La etiqueta `<html> ... </html>` define el comienzo y el final del documento.

La cabecera `<head> ... </head>` es un elemento que contiene los encabezados de la página.

Recuerda, que en la cabecera se pueden incluir:

- Etiquetas *meta*.
- Título.
- Recursos enlazados.

La etiqueta `<body>...</body>` define el cuerpo del documento HTML, en esta se incluyen todas las etiquetas que veremos a continuación en el curso, y por supuesto los atributos y el texto o contenido de cada elemento.



EL DOCTYPE

La etiqueta `<!DOCTYPE ...>` es de uso obligado, ya que informa al navegador del tipo de documento (si es HTML o XHTML) y cuál es la versión. Recuerda que existen varios tipos de documentos y diferentes versiones.

Esto es necesario para traducir las etiquetas, ya que no son iguales en todas las versiones. En la etiqueta `<!DOCTYPE ...>` se incluye la URL de la DTD en la que se especifica la versión del lenguaje de etiquetas utilizado en el documento, además de la estructura, etiquetas y atributos permitidos para cada una de ellas.

En HTML 4.01 se indicará si el **doctype** es *strict*, *transitional* o *frameset*. Para HTML5 se usa `<!DOCTYPE html>`

Tipos de doctype

Strict

Es la versión más estricta y severa. Los sitios que incluyan este tipo de *doctype*, no pueden incluir atributos que estén relacionados con las características y el aspecto de los contenidos. Esto supone una separación del documento HTML de los estilos CSS. Es la recomendada por la W3C.

Transitional

Esta versión, a diferencia de la anterior, sí permite utilizar alguno de los atributos de aspecto para el contenido.

Frameset

Esta versión la utilizan las páginas o sitios que están formados por *frames* (marcos), algo que ya está muy en desuso y prácticamente obsoleto.

Actualmente, con HTML5 se ha simplificado el *Doctype* indicando solo el parámetro "html".

`<!DOCTYPE html>`

Ejemplos

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad

HTML 4.01 Transitional: esta DTD contiene todos los elementos HTML y atributos, incluyendo elementos de presentación y obsoletas (como ``). `<frameset>` no está permitida.

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

HTML 4.01 Frameset: igual que el anterior, pero se permiten los `<frameset>`.

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

XHTML 1.0 Strict: esta DTD contiene todos los elementos HTML y atributos, pero no incluye los elementos de presentación o en desuso (como ``). `<frameset>` no está permitido. El marcado también debe ser escrito como XML bien formado.

```
<!DOCTYPE html>
```

XHTML5

HTML5

HTML5 es el estándar más actual para realizar nuestras páginas. Se basa en la unión de HTML, CSS y JavaScript.

HTML5 es la última actualización de HTML. HTML5 también es un termino que **sirve para agrupar las tecnologías web HTML5, CSS3 y nuevas capacidades de JavaScript.**

La necesidad de esta nueva versión viene por que su antecesora HTML 4 carece de las características necesarias para la creación de aplicaciones modernas basadas en un navegador y los requerimientos de indexación (SEO) que demandan los buscadores de Internet como Google.

Otra de las razones es el declive de uso de Adobe Flash para dotar de dinamismo nuestras páginas. Así que es necesario que HTML disponga de herramientas para que nuestros navegadores puedan gestionar audio, vídeo, animaciones vectoriales, componentes de interfaz, entre muchas otras cosas. Ahora HTML5 es capaz de hacer esto sin necesidad de *plugins* y con una gran compatibilidad entre navegadores.





Cuando trabajes con documentación, fíjate bien que **esté en HTML5**. Muchas de las estructuras y atributos usados en HTML 4 ya no son validadas y nos perjudicarán en el posicionamiento natural de nuestras páginas (SEO).

Validación de marcado

Para saber si nuestro código cumple los estándares de HTML5 podemos utilizar las herramientas de validación disponibles en Internet.

<http://validator.w3.org/>.

Una de las características más relevantes de HTML5 no es la inclusión de nuevas etiquetas, sino la simplificación de etiquetas y el uso de atributos existentes en versiones anteriores.

La cabecera del documento <head>

La cabecera <head> ... </head> es un elemento que contiene los encabezados de la página. En HTML5 puede ser omitida.

En la cabecera se pueden incluir:

- **Etiquetas *meta*.** También llamadas "metatags" y sirven para integrar metadatos sobre la página. Normalmente se utilizan para dar información a los motores de búsquedas (por ejemplo Google) y mejorar la indexación de la página. Esto es importante a la hora del posicionamiento web.

Su sintaxis es muy sencilla:

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

- **Título.** El título de la página y el que aparece en la Barra de un navegador.
- **Recursos enlazados.**

```
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">  
  <title>Título de la página web</title>  
  <meta name="generator" content="Amaya, http://www.w3.org/Amaya/">  
  <link rel="stylesheet" href="ie7style.css" type="text/css" media="screen">  
</head>
```


Etiquetas que se pueden incluir en la etiqueta `<head>`

Charset

Valor: conjunto de caracteres UTF-8 Unicode.

Descripción: especifica la codificación de caracteres del HTML.

Content

Valor: texto.

Descripción: valor dado asociado a la etiqueta *http-equiv* o el valor de un atributo.

http-equiv

Valor: *application-name*, *author description*, *generator*, *keywords*.

Descripción: metadatos que definen los conceptos asociados.

Link

Valor: recursos enlazados, p. ej.: `<link rel="stylesheet" type="text/css" href="theme.css">`.

Descripción: sirve para incluir estilos enlazándolos con el fichero que se indica.

Script

Valor: p. ej.: `<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>`.

Descripción: en la cabecera se puede enlazar con librerías de JavaScript.

Base

Valor: indica la raíz de la página, p. ej.: `<base href="http://www.midominio.org/">`.

Descripción: establece la URL base del documento.

Style

Valor: p. ej.: `<style> h1{color:red;} p{color:blue} </style>`.

Descripción: define estilos de la página web en la cabecera.

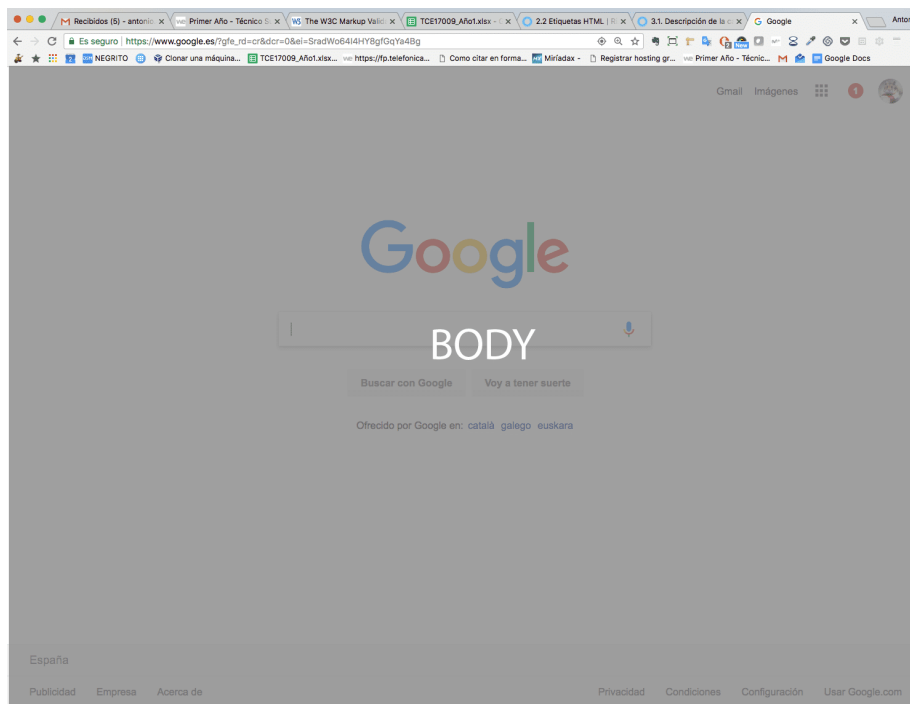
Puedes ampliar la información sobre estas etiquetas en este enlace

El cuerpo de la página <body>

La etiqueta **<body>...</body>** define el cuerpo del documento HTML.

En esta se incluyen todas las etiquetas que veremos a continuación en el curso, y por supuesto los atributos y el texto o contenido de cada elemento.

```
<body>
<h1>Mi primer encabezado</h1>
<p>Mi primer párrafo</p>
</body>
```



Entre las etiquetas *<body>* se encuentra el contenido visible de la página.

Los textos

Encabezados: la estructura de un documento web, en principio, debería ser parecida a un documento de texto preparado para la lectura y organizado según temas y subtemas. **Las etiquetas de los encabezados o "headings"** sirven para ello.

Hay hasta [seis encabezados](#), que difieren en el tamaño del texto que incluyen. Los encabezados se delimitan con el *tag* `<h1>` hasta el `<h6>`, `<h1>` es el título mayor y así van disminuyendo hasta el sexto, que sería el título menor.

La mayoría de los navegadores visuales muestran encabezados como texto grande y en negrita de forma predeterminada, aunque esto puede ser configurado si modificamos los estilos en la CSS asociada.

Los elementos de título no pueden ser usados para agrandar o poner en negrita texto normal, ya que su uso está asociado al título de una sección. Los encabezamientos o títulos existen en etiquetas HTML desde su estandarización en la versión HTML 2.0, y todavía se usan. Todos los navegadores soportan esta etiqueta.

Párrafos

Los documentos en HTML, como cualquier documento de texto, están divididos en párrafos. Un [párrafo](#) está definido por la etiqueta o elemento `<p>...</p>`.

Tenemos otras etiquetas que sirven para introducir un salto de línea, como el elemento `
`.

También existe la etiqueta `<pre>...</pre>` y sirve para que el texto se visualice tal como está en el código fuente, con lo que respetaría los saltos de línea o espacios en blanco extra y el texto se representa con un tipo de fuente de ancho fijo.

En HTML5 todas las modificaciones de apariencia y estilo se realizarán con CSS, por lo que se han eliminado los atributos para ese fin.

Tanto en párrafos como en otros contenedores de texto podemos querer poner partes en negrita, cursiva, subrayado, etc ([palabras resaltadas](#)). Para ello disponemos también de multitud de etiquetas de estilo lógico o físico.

Etiquetas de estilo físico

Especifican un cambio concreto en el tipo de letra, estas etiquetas están en desuso frente a las de estilo lógico o semántico, dado que favorecen el posicionamiento SEO de la página.

- texto en negrita: ` ... `
- texto en cursiva: `<I> ... </I>`
- texto subrayado: `<U> ... </U>`
- texto monoespaciado: `<TT> ... </TT>`
- texto tachado: `<STRIKE> ... </STRIKE>`
- aumentar el tamaño del texto: `<BIG> ... </BIG>`
- disminuir el tamaño del texto: `<SMALL> ... </SMALL>`
- texto en subíndice: `_{...}`
- texto en superíndice: `^{...}`

Etiquetas de estilo lógico

No solo especifican una zona de texto a resaltar, sino que le dan contenido semántico:

- texto destacado: ` ... `
- texto importante: ` ... `
- citas y referencias: `<CITE> ... </CITE>`
- información sobre el autor: `<ADDRESS> ... </ADDRESS>`
- definición de un término: `<DFN> ... </DFN>`
- fragmento de código de programa: `<CODE> ... </CODE>`
- Ejemplos de programa: `<SAMP> ... </SAMP>`
- Variables, argumentos de un programa: `<VAR> ... </VAR>`

Sangrado

`<blockquote>... </blockquote>` sirve para realizar un sangrado en el texto.

Esta etiqueta se usaba inicialmente para el marcado de citas, como los navegadores hacen un sangrado para la cita así, se convirtió en la etiqueta para el sangrado. Esta etiqueta tiene el atributo *cite*, que es del tipo URL.

Para aplicar un sangrado desde CSS usaremos el atributo *text-indent*, al que le indicaremos un valor numérico.

```
<blockquote cite="http://www.google.com"> ... </blockquote>
```

Listas

Las listas de conceptos nos ayudan a definir conceptos y valores.

- Elemento 1
- Elemento 2
 - 1. Subelemento1
 - 2. Subelemento1
 - 3. Subelemento1
 - 4. Subelemento1
- Elemento 3
- Elemento 4

Ejemplo lista anidada.

Lista desordenada

`` define una lista sin orden. Cada elemento de la lista comienza con la etiqueta ``.

```
<ul>
  <li>Café</li>
  <li>Leche</li>
</ul>
```

Lista ordenada

`` define una lista ordenada, cada elemento de la lista comienza con la etiqueta ``.

```
<ol>
  <li>Café</li>
  <li>Leche</li>
</ol>
```

Listas de definición

`<dl>` es una etiqueta que define una lista de descripciones, se utiliza en conjunción con:

- `<dt>` términos a definir.
- `<dd>` describe cada término.

Las listas tienen unos atributos específicos que definen la marca delante del elemento de la lista.

```
<dl>
  <dt>Café</dt>
  <dd>- bebida negra caliente</dd>
  <dt>Leche</dt>
  <dd>- bebida blanca fría</dd>
</dl>
```

Anidamiento de listas

Para anidar [listas](#) se usan las mismas estructuras que hemos estudiado, teniendo en cuenta que dentro de cada elemento definido por `` incluiremos elementos ``. Veamos un posible anidamiento.

```
<ol>
  <li><h3>Primer plato</h3>
    <ul>
      <li>Macarrones</li>
      <li>Gazpacho</li>
      <li>Judías pintas</li>
    </ul>
  </li>
  <li><h3>Segundo plato</h3>
    <ul>
      <li>Pollo al ajillo</li>
      <li>Merluza frita</li>
    </ul>
  </li>
</ol>
```


Los enlaces

URL "**Uniform Resource Locator**" se traduce como localizador de recursos uniforme.

Este recurso se diseña y se usa por primera vez por Tim Bernes-Lee para enlazar documentos a través de hiperenlaces en la red *World Wide Web*.

URL está formado por un formato específico que está estandarizado desde 1994 y forma parte del estándar URI "**Uniform Resource Identifier**", en español identificador uniforme de recurso.

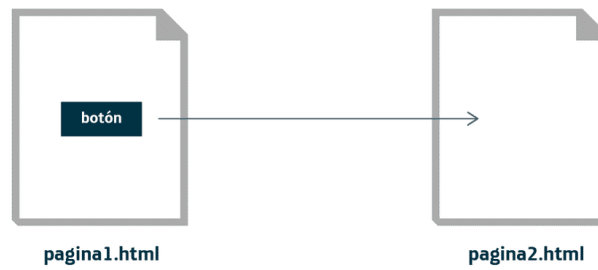
El recurso tiene una dirección única disponible en Internet. El formato general de una URL es:

esquema://máquina/directorio/archivo

<a> es la etiqueta usada para enlazar con recursos a través de una URL.

El enlace se puede asociar a un texto, una imagen o un elemento HTML. Una vez enlazado, pulsando en el elemento se navegará hacia el recurso definido. **El atributo más importante del elemento <a> es el atributo href**, que indica el destino del enlace, es decir, la URL del recurso al que queremos acceder.

Otros atributos como *download*, *hreflang*, *media*, *rel*, *target*, *type*... no tiene sentido usarlos mientras que no tengamos el atributo *href*.



` botón `

Atributo *target*

El atributo *target* está soportado por HTML5 y los navegadores más usados. Antiguamente se usaba para indicar en qué división (*frame*) de la página queríamos que actuase el enlace. Actualmente en HTML5 solo se usa la opción *_blank* para abrirlo en una ventana nueva.

```
<a href="direccion" target="_blank">Boton</a>
```

Enlaces absolutos y relativos

Normalmente las páginas que se desarrollan en un proyecto estarán alojadas en un servidor web, que además tendrá referencias del dominio que apunta a dichas páginas web. Esto implica que la URL base del dominio apunta al directorio raíz del sitio, por ejemplo `http://www.misitio.com`. El resto de recursos estarán, en la mayoría de los casos, alojados en directorios del sistema de ficheros del servidor que parten de la raíz del sitio o en directorios virtuales.

Dirección absoluta

Para hacer referencia a un sitio externo se usarán direcciones absolutas que indicarán la dirección raíz del sitio, por ejemplo *www.misitio.com*. Si es necesario acceder a un recurso como “fichero.html”, alojado en una carpeta o directorio hijo, por ejemplo “/documentos/”, se añadirá a la dirección global de esta forma:

```
<a href="www.misitio.com/documentos/pdf/fichero.html">
```

Dirección relativa

Para hacer referencia a un recurso interno que forma parte del sitio web que está por encima, por debajo o al mismo nivel en el árbol de directorio del sitio, usaremos direcciones relativas al directorio donde está la página que contiene el enlace al recurso.

Si el sitio web es *www.misitio.com* y quiero hacer referencia a un recurso que está en el directorio “/documentos/” y tengo que acceder a un recurso que cuelga de ese directorio, por ejemplo en “/documentos/pdf/fichero.html”, podré crear la dirección de esta forma:

```
<a href="/documentos/pdf/fichero.html">
```

Las tablas

<table> define una tabla en HTML. Cada **<tr>** define una fila y cada **<td>** define una celda, cuyo número deberá coincidir con el número de columnas definidos.

Listado de personas por edad

Nombre	Apellido	Edad
Juan	Escudero	50
Josefa	Valcarcel	94

Cada celda puede contener otros elementos como texto, tablas, listas imágenes, etc. El elemento **<th>** define un encabezado de la tabla.

<caption> sirve para definir el título de la [tabla](#), saldrá al principio de la tabla. HTML5 solo admite el atributo **"border"**, y su valor puede ser **"1"** o **""**.

Igualmente que el resto de elementos, **en HTML5** el aspecto visual (bordes, márgenes, líneas...), **se aplicarán únicamente con CSS**.

Agrupaciones

Para formatear una tabla se utilizan los estilos de CSS, pero también existen marcas para [agrupar](#) y aplicar estilos como estas:

- `<colgroup>`. Especifica un grupo de una o más columnas en una tabla para formatear.
- `<col>`. Especifica las propiedades de columna para cada columna dentro de un elemento `<colgroup>`.
- `<thead>`. Agrupa elementos de la cabecera de la tabla.
- `<tbody>`. Agrupa elementos del cuerpo de la tabla.
- `<tfoot>`. Agrupa elementos del pie de la tabla.



Las agrupaciones con los elementos *thead*, *tbody* y *tfoot* no afectan al diseño de la tabla por defecto, pero se podrían usar estilos para cambiarlos.

El anidamiento de tablas hace que el navegador tenga que procesar más instrucciones, por lo que hay que intentar evitarlo.

Colspan y rowspan

Países Europeos	Madrid
	Francia
	París

Países Europeos		
Madrid	Francia	París

En ocasiones queremos formar tablas donde [una celda abarque más que otras](#), para ello utilizaremos *colspan* y *rowspan*.

- **rowspan:** indica el número de celdas que ocupará horizontalmente en la fila.
- **colspan:** indica el número de columnas que ocupará la celda.

De esta forma, si ponemos `<td colspan="2">` quiere decir que la celda actual se extiende en el ancho de dos celdas.

Algo parecido ocurre si ponemos `<td rowspan="3">`, la celda ocupará el alto de 3 celdas normales.

Las imágenes

La etiqueta **** nos permite integrar imágenes en nuestro documento HTML.

Una imagen en formato digital contiene información sobre cada parte de la imagen, es decir, cada píxel. A través de una codificación se especifica información como el color y las capas.

Los formatos más utilizados son:

JPEG

Joint Photographic Experts Group; comprime la información de la imagen mediante un algoritmo. Eso supone pérdida de calidad.

PNG

Portable Network Graphics; comprime la información sin ninguna pérdida y no tiene patente. Almacena por lo tanto más datos que el formato anterior.

GIF

Graphics Interchange Format; este formato admite hasta 8 bits por píxel en cada imagen, y permite una sola imagen para hacer referencia a su propia paleta de hasta 256 colores diferentes, elegidos de los 24 bits de espacio de color RGB. También es compatible con animaciones y permite una paleta de hasta 256 colores para cada fotograma. Estas limitaciones en la paleta hacen que el formato GIF sea menos adecuado para la reproducción de fotografías en color y otras imágenes con color continuo, pero es muy adecuado para las imágenes más simples, tales como gráficos o logos con áreas solidas de color.

SVG

SVG (*Scalable Vector Graphics*) o Vectores Gráficos Escalables es la herramienta que nos proporciona HTML5 para crear gráficos o animaciones mediante vectores.

```

```

Características de las imágenes: tamaño, título, textos alternativos.

`` es la etiqueta para incluir imágenes en nuestra página web. `` no tiene cierre y requiere llevar dos atributos, que son `src` y `alt`. `` tiene atributos globales y de eventos y además unos específicos.

```

```

Ruta y nombre del archivo de imagen

Texto alternativo de la imagen,
es decir, si no se pudiera cargar la
imagen, sería el texto que
sustituiría dicha imagen.

Atributos

Los atributos de la etiqueta `` más habituales son:

- **src** - Ruta relativa o absoluta del archivo de la imagen.
- **alt** - Texto alternativo de la imagen.
- **title** - Título de la imagen.

Introducción a SVG

SVG (*Scalable Vector Graphics*) o Vectores Gráficos Escalables es la herramienta que nos proporciona HTML5 para crear gráficos o animaciones mediante vectores.

Con el uso de Flash desaprobado para realizar elementos como dibujos, banners, animaciones, etc. Las alternativas actuales son *Canvas* y SVG, con ambos podemos hacer las mismas cosas pero con diferente forma de trabajar.

Mientras que con *canvas* se basa en mapas de bits, con SVG los dibujos se crean mediante vectores. Esto significa que la creación de los elementos se realiza mediante parámetros y coordenadas en lugar de píxeles, lo que hace que la **calidad sea óptima** dado que se **adaptará a cualquier resolución**.

SVG se basa en el lenguaje XML. Mediante etiquetas construiremos las figuras teniendo la posibilidad de crear animaciones en conjunción con Javascript.



SVG se basa en el lenguaje XML. Mediante etiquetas construiremos las figuras teniendo la posibilidad de crear animaciones en conjunción con Javascript.

Veamos cómo se construyen algunas figuras básicas en SVG

Para integrar una imagen en SVG es necesario integrar las etiquetas `<svg> ... </svg>` e indicar las dimensiones totales del gráfico (*width* y *height*) como atributos. Dentro de esta etiqueta definiremos el tipo de figura a integrar y sus propiedades.

```
<svg width=200 height=200> ... gráfico ... </svg>
```

Puedes consultar la web de w3schools para ampliar la información sobre SVG.

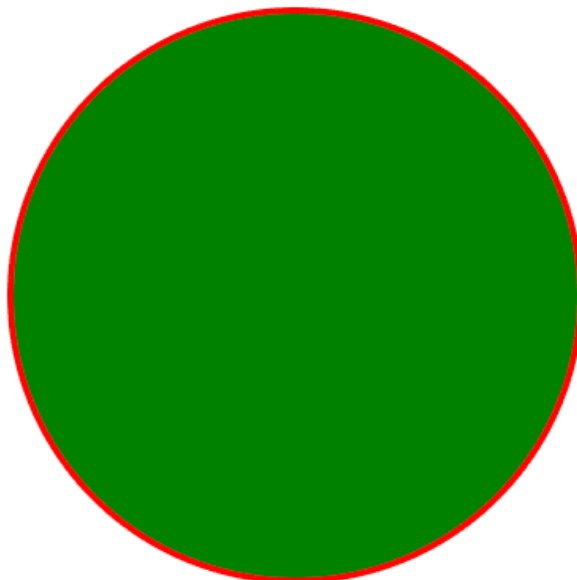
[SVG](#)

Puedes ampliar la información sobre los atributos de los SVG en este enlace.

[ATRIBUTOS SVG](#)

Crear un círculo

Para construir el elemento, tendremos que darle una serie de propiedades como sus dimensiones, borde, relleno, etc.



```
<svg width="200" height="200">  
<circle cx="100" cy="100" r="90" stroke="red" stroke-width="2" fill="green">  
</svg>
```

Propiedades:

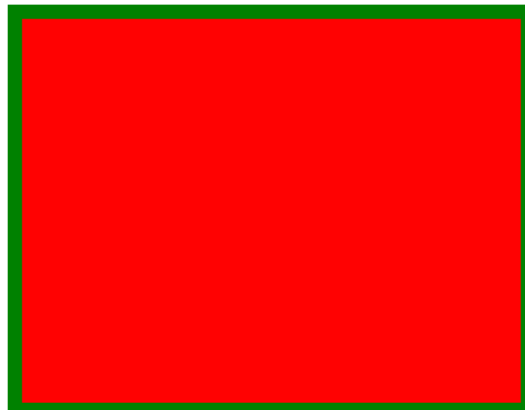
- **circle**: etiqueta correspondiente a la figura.
- **cx=n cy=n**: las coordenadas para la construcción del círculo. Indica en pixeles las distancias desde el centro tanto del eje x como del eje y.
- **r=n**: Longitud del radio de la circunferencia.

En el ejemplo se aplican otros atributos que podemos usar en todas las figuras:

- **stroke="color"**: Color del borde.
- **stroke-width=n**: Grosor de la línea exterior.
- **fill="color"**: Color del relleno de la figura.

Cuadrados y rectángulos

Como en el ejemplo anterior usaremos la etiqueta correspondiente a la figura, dando valores a los atributos que queramos aplicar.



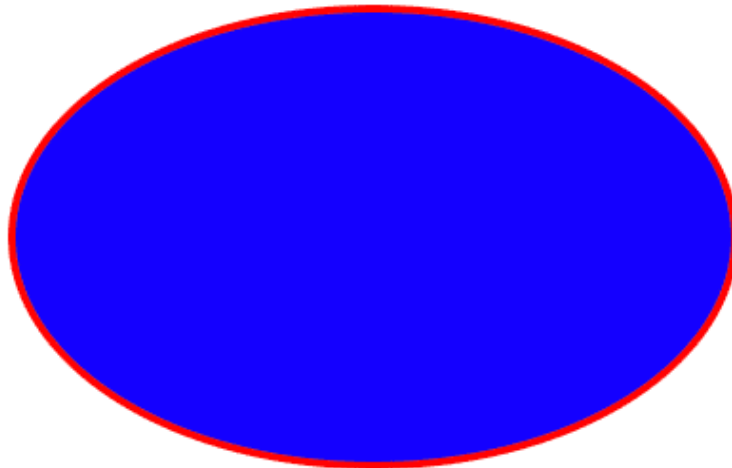
```
<svg width="200" height="200">
<rect x="10" y="30" width="180" height="140" stroke="green" stroke-width="5"
fill="red"/>
</svg>
```

Propiedades:

- **rect:** con esta etiqueta indicamos que vamos a dibujar un rectángulo.
- **x=n, y=n:** Esquina superior izquierda del rectángulo.
- **width=n, height=n:** Anchura y altura del rectángulo.

Elipse

Al igual que en las figuras anteriores utilizamos los atributos *stroke*, *stroke-width* y *fill*, para indicar el color del borde, el grosor del borde, y el color de relleno.



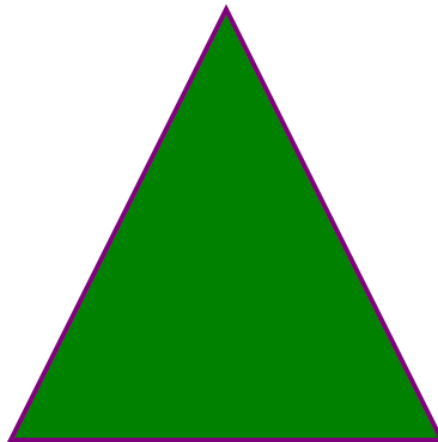
```
<svg height="200" width="200">
<ellipse cx="100" cy="100" rx="95" ry="60"
stroke="red" stroke-width="2" fill="blue" />
</svg>
```

Propiedades:

- *ellipse*: etiqueta para elipse.
- *cx=n, cy=n*: coordenadas del centro de la elipse.
- *rx=n*: radio horizontal.
- *ry=n*: radio vertical.

Polígonos

En el caso de los polígonos iremos indicando los puntos con el atributo *point* en una serie de parejas de números.



```
<svg height="200" width="200">  
  <polygon points="100,10 190,190 10,190"  
    stroke="purple" stroke-width="2" fill="green"/>  
</svg>
```

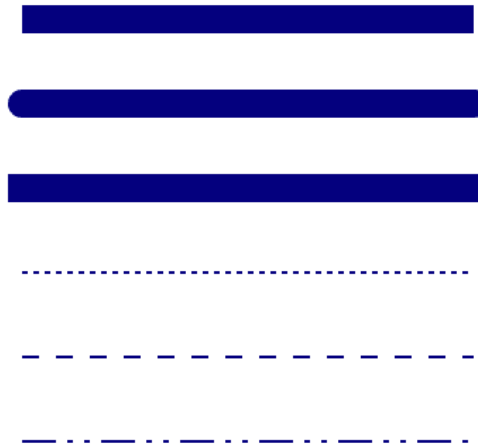
Propiedades:

- *polygon*: etiqueta para crear el polígono.
- *points*: cada pareja de números sirve para definir la coordenada x e y de cada vértice del polígono.

Líneas

Para dibujar líneas usaremos la etiqueta *line*, e indicando mediante coordenadas x e y tanto el inicio como el final de la línea.

```
<svg height="200" width="200">
  <line x1="5" y1="5" x2="190" y2="190" stroke="red" stroke-width="2"/>
</svg>
```



```
<svg width="200" height="200">
  <line x1="20" y1="20" x2="180" y2="20" stroke="navy" stroke-width="10"
  stroke-linecap="butt"/>
  <line x1="20" y1="50" x2="180" y2="50" stroke="navy" stroke-width="10"
  stroke-linecap="round"/>
  <line x1="20" y1="80" x2="180" y2="80" stroke="navy" stroke-width="10"
  stroke-linecap="square"/>
  <line x1="20" y1="110" x2="180" y2="110" stroke="navy" stroke-width="2"
  stroke-dasharray="2,2"/>
  <line x1="20" y1="140" x2="180" y2="140" stroke="navy" stroke-width="2"
  stroke-dasharray="6,6"/>
  <line x1="20" y1="170" x2="180" y2="170" stroke="navy" stroke-width="2"
  stroke-dasharray="12,4 2,4 2,4"/>
</svg>
```

- x_1, y_1 : coordenadas horizontal y vertical del principio de la línea.
- x_2, y_2 : coordenadas horizontal y vertical del final de la línea.
- ***stroke-linecap="butt"***: forma de inicio de la línea. Admite los siguientes valores:
 - *butt*
 - *round*
 - *square*
 - *inherit*
- ***stroke-dasharray***: líneas discontinuas.

Juntando SVG

[Aquí](#) tienes un ejemplo de integración de varios SVG.

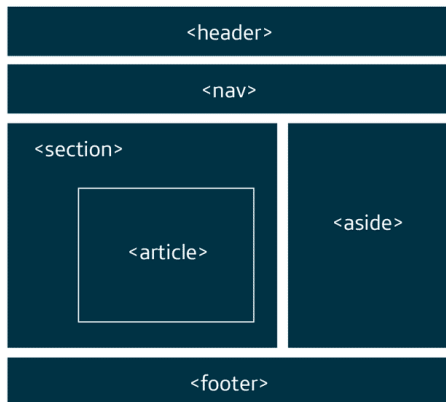
Etiquetas semánticas

HTML5 incorpora nuevas etiquetas denominadas etiquetas semánticas. Estas etiquetas no tienen contrapartida visual, actúan como capas igual que la etiqueta `<div>`. Lo que las diferencia de las etiquetas `<div>` es que sirven para informar sobre el tipo de contenido que soporta.

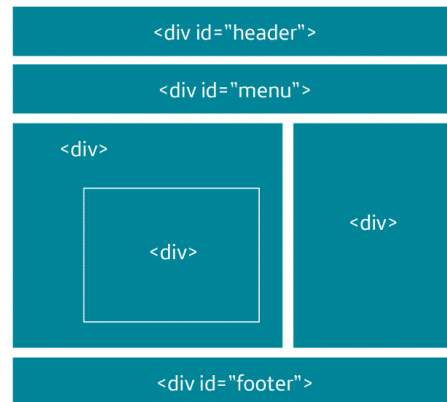
Nuevas etiquetas semánticas:

- `<header>`. Cabecera de la página. Normalmente contendrá el logotipo y título.
- `<nav>`. Área de navegación. Normalmente contendrá enlaces, menús o botones con vínculos.
- `<section>`. Una sección dentro de la página que podrá estar dividida en artículos.
- `<article>`. Un artículo.
- `<aside>`. Área de *banners* o publicidad.
- `<footer>`. Pie de página.

Maquetación HTML5



Maquetación versiones anteriores



Maquetación de una web con etiquetas semánticas HTML5 y maquetación a base de etiquetas `<div>` con las versiones anteriores.

El uso de las etiquetas semánticas nos ayuda a posicionar mejor nuestra web en los buscadores, ya que los motores de búsqueda sabrán el tipo de contenido que encierran y darán más peso a unas áreas que a otras. Por ejemplo, para los motores de búsqueda tiene más peso el texto incluido en el encabezado del área de navegación que el texto del resto de la página.

Un ejemplo de documento HTML maquetado con HTML5:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Página con formato HTML5</title>
<link href="estilos/estilo.css" rel="Stylesheet">
</head>
<body>
  <header> header </header>
  <nav>
    <a href="#">Opción 1</a>
```

```

        <a href="#">Opción 2</a>
        <a href="#">Opción 3</a>
    </nav>
    <div id="contenido">
        <section>
            <article>article 1</article>
            <article>article 2</article>
            <article>article 3</article>
            <article>article 4</article>
        </section>
        <aside>aside</aside>
    </div>
    <footer> footer </footer>
</body>
</html>

```

Página web con etiquetas semánticas HTML5.

Otras etiquetas

Contenedores <div>

La etiqueta <div> establece un bloque mediante una división o una sección de un documento web en HTML. Además, se usa para establecer y/o formatear a través de un CSS un bloque de texto.

```
<!DOCTYPE html>
<html>
  <body>
    <div style="color:#0000FF">
      <h3>Un título de texto en un bloque div que es azul</h3>
      <p>Texto normal en un elemento div que es azul.</p>
    </div>
    <p>Texto fuera del div.</p>
  </body>
</html>
```

Etiqueta

La etiqueta se usa para realizar agrupaciones en línea de los elementos de un documento. Esta etiqueta no cambia la visualización de un texto ella sola, necesitará de estilos de CSS para realizarlo. Además, facilita añadir una clase a una parte específica de un texto o de una parte de un documento.

```
<!DOCTYPE html>
<html>
  <body>
    <p>Mi hermano tiene los ojos <span style="color:blue; font-weight:bold;">azules</span>. </p>
  </body>
</html>
```

Creación de un formulario sencillo

Los formularios permiten recoger información del usuario para enviarla al servidor web. También pueden ser utilizados para mostrar información al usuario.

Vamos a comenzar por crear un formulario simple y a partir de su código iremos explicando la teoría.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Formulario</title>
<link href="estilos/estilo.css" rel="Stylesheet">
</head>
<body>
  <form action="procesa.jsp" method="get">
    <fieldset><legend>Datos personales:</legend>
      <label for="nombre">Nombre: </label>
      <input type="text" name="nombre" />
      <br />
      <label for="edad">Edad: </label>
      <input type="text" name="edad" />
      <br />
      <input type="submit" value="Enviar datos" />
      <input type="reset" value="Limpiar formulario" />
    </fieldset>
  </form>
</body>
</html>
```

Los datos del formulario se entregan al recurso de servidor *procesa.jsp* cuando el usuario pulsa el botón *submit* (enviar).

Vamos a estudiar las nuevas etiquetas:

form

`<form>`

Un formulario está situado dentro de la etiqueta `<form>` y encierra elementos de formulario que permiten al usuario interactuar. Estos elementos pueden ser: campos de texto, etiquetas, botones, listas desplegables, casillas de verificación, etc.

El **atributo *action*** determina el proceso al que se enviarán los datos introducidos por el usuario cuando se pulse el botón *submit*.

El **atributo *method*** determina la forma en que se enviarán los datos. Con el valor *get* los datos se envían en la cabecera de la petición HTTP, por lo que serán visibles en la URL. Este método no es recomendable si los datos son confidenciales. Con el valor *post* los datos se envían en el cuerpo de la petición HTTP, por lo que no serán visibles en la URL.

fieldset

`<fieldset>`

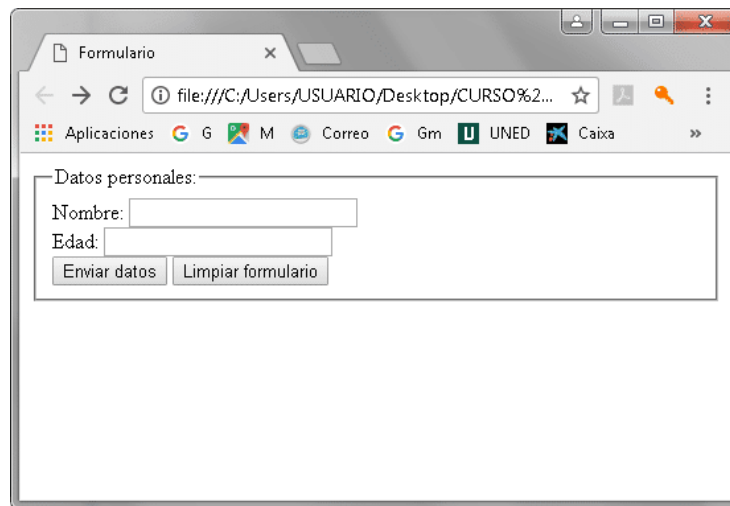
Genera un marco visible que encierra los datos del formulario.

legend

`<legend>`

Colocado inmediatamente después de una etiqueta `<fieldset>`, permite colocarle una etiqueta de texto (título).

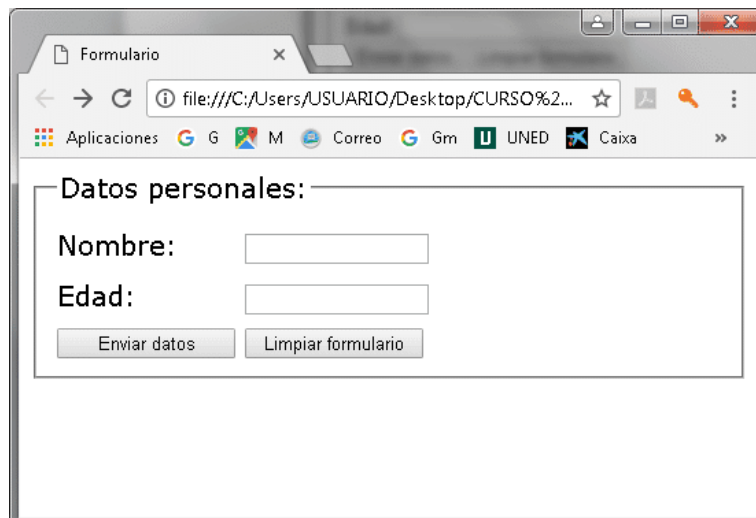
El resultado en el navegador, sin haber creado todavía el archivo CSS es el siguiente:



Con un poco de CSS podemos lograr que el estilo cambie bastante.

```
@CHARSET "ISO-8859-1";

body {
    font-family: Verdana;
    font-size: 20px;
}
input, label {
    margin-top: 10px;
    display: inline-block;
    width: 130px;
}
```



Campos de formulario

Un formulario de entrada de datos está compuesto por un conjunto de campos que pueden ser de distinto tipo: cajas de texto, casillas de verificación, listas de opciones, etc.

Vamos a descubrir poco a poco las etiquetas HTML que representan campos de formulario.

`<input>`

La etiqueta `<input>` define un elemento de campo de formulario. Estos son sus atributos más comunes:

- ***type***: tipo de campo. El valor más usual es *text* (caja de texto).
- ***value***: valor por defecto.
- ***readonly***: solo lectura. No puede ser modificado por el usuario.
- ***name***: el valor de este atributo será el que usará el recurso de servidor para recuperar el valor del campo.

```
<input type="text" name="edad" />
```

Los datos que introduzca el usuario en el formulario serán enviados a un programa de servidor, que utilizará el identificador *edad* para recoger el valor introducido por el usuario.

`<label>`

Define una etiqueta estática, que normalmente estará asociada con un elemento de campo de formulario que se identifica mediante el atributo *for*.

También se utiliza para presentar resultados de operaciones.

`<input type="submit">`

Se representa como un botón y cuando el usuario hace clic en él, se realiza la petición HTTP al programa de servidor especificado en el atributo *action*.

`<input type="reset">`

Se representa como un botón y cuando el usuario hace clic en él, se borra el contenido de todos los campos del formulario.

`<input type="hidden">`

Campo oculto. Puede ser útil para enviar una información al servidor que no queremos que sea vista por el usuario.

`<input type="password">`

Funciona como una caja de texto, pero en lugar de los caracteres que escribe el usuario se ven solo asteriscos o círculos.

`<input type="radio">`

Se representa como un botón de *radio*. Todos los botones de *radio* que tienen el mismo *name* pertenecen al mismo grupo y solamente uno de los elementos del grupo podrá ser seleccionado por el usuario.

```
<label for="ec">Estado civil: </label>
<br />
<input type="radio" name="ec" value="S" />Soltero<br />
<input type="radio" name="ec" value="C" />Casado<br />
<input type="radio" name="ec" value="V" />Viudo<br />
<input type="radio" name="ec" value="D" />Divorciado<br />
```

Este es el resultado:

Estado civil:

- ☐ Soltero
- ☒ Casado
- ☐ Viudo
- ☐ Divorciado

El usuario solamente podrá seleccionar uno de los cuatro valores. La variable que recogerá el programa *procesa.jsp* será *ec* y el valor enviado será *S*, *C*, *V* o *D* según el botón de *radio* seleccionado por el usuario.

`<input type="checkbox">`

Se representa como una casilla de verificación que el usuario podrá marcar y desmarcar.

```
<label for="ec">Estado civil: </label>
<br />
<input type="radio" name="ec" value="S" />Soltero<br />
<input type="radio" name="ec" value="C" />Casado<br />
<input type="radio" name="ec" value="V" />Viudo<br />
<input type="radio" name="ec" value="D" />Divorciado<br />
<br />
```

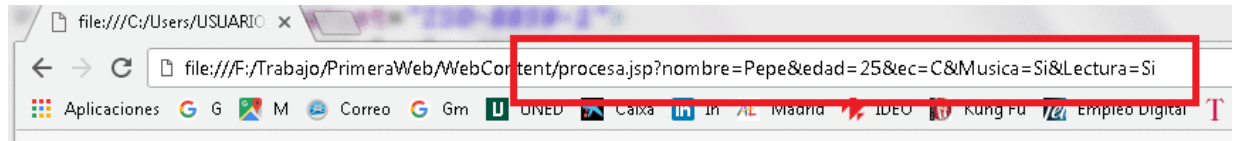
Si has ido añadiendo todos los ejemplos a tu página, el formulario se visualizará así:

The screenshot shows a web browser window with the title 'Formulario'. The address bar displays a local file path: `file:///C:/Users/USUARIO/Desktop`. The browser's toolbar includes icons for 'Aplicaciones', 'G', 'G', 'M', 'Correo', 'Gm', and 'UNED'. The form itself is titled 'Datos personales:' and contains the following elements:

- Nombre:** A text input field containing the value 'Pepe'.
- Edad:** A text input field containing the value '25'.
- Estado civil:** A group of four radio buttons with labels: 'Soltero', 'Casado' (which is selected), 'Viudo', and 'Divorciado'.
- Aficiones:** A group of four checkboxes with labels: 'Música' (checked), 'Lectura' (checked), 'Senderismo' (unchecked), and 'Deportes' (unchecked).

At the bottom of the form, there are two buttons: 'Enviar datos' and 'Limpiar formulario'.

Al pulsar el botón "Enviar datos" con los datos especificados en la imagen, la URL transportará los datos, que podrá recoger el programa de servidor *procesa.jsp*.



Parámetros enviados a través de la URL al recurso de servidor *procesa.jsp*. Como no existe el recurso, se producirá un error en la página, pero te sirve igualmente para comprender el mecanismo de las peticiones HTTP.

<textarea>

Se representa como una caja de texto multilínea. Tiene los siguientes atributos:

- **rows:** número de filas visibles del campo.
- **cols:** ancho del campo.
- **readonly:** solo lectura.

```
<label for="observaciones">Observaciones: </label>  
<br />  
<textarea rows="5" cols="80" name="observaciones"></textarea>
```

Resultado:

Observaciones:

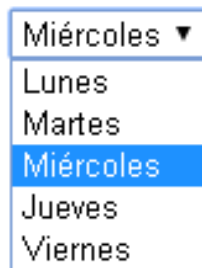
En un lugar de la Mancha, de cuyo nombre no quiero acordar-me, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda. El resto

`<select><option>`

La etiqueta `<select>` representa una lista desplegable de opciones definidas por medio de etiquetas `<option>`.

```
<select name="dia">
  <option value="LUN">Lunes</option>
  <option value="MAR">Martes</option>
  <option value="MIE" selected>Miércoles</option>
  <option value="JUE">Jueves</option>
  <option value="VIE">Viernes</option>
</select>
```

El texto contenido en cada `<option>` es el valor visible para el usuario, tal y como puedes apreciar en la imagen:



El valor que se enviará al servidor es el que contiene el atributo *value* de la opción seleccionada por el usuario (*LUN*, *MAR*, *MIE*, *JUE* o *VIE*).

Nuevos atributos para campos de formularios HTML5

Los formularios son uno de los ejemplos más claros donde pueden apreciarse las mejoras que ha aportado HTML5, con nuevos atributos y nuevos tipos de campos.

En este apartado vamos a ver los nuevos atributos que HTML5 ha aportado para la etiqueta `<input>`.

placeholder

Un texto que se muestra al hacer clic sobre una caja de texto para guiar al usuario y que desaparece cuando se sitúa el cursor.

Edad:

```
<input type="text" name="edad" placeholder="Escribe tu edad"/>
```

autofocus

El campo al que se le añade el atributo *autofocus* será el que esté seleccionado por defecto al cargar la página, es decir, tendrá el cursor.

```
<input type="text" name="edad" autofocus />
```

required

Hace que el campo al que se le aplica sea obligatorio. No se podrá enviar el formulario si rellenarlo.

n

En la imagen, si se intenta enviar el formulario sin rellenar la edad, sale un mensaje de advertencia y no permite enviar.

Nombre:

Edad:

Estado civil: ☒ Soltero ☐ Casado

 Completa este campo

```
<input type="text" name="edad" required />
```

Nuevos tipos de campos HTML5 con validación automática

En este apartado veremos nuevos valores que se pueden aplicar al atributo *type* de la etiqueta `<input>`. Se trata de valores nuevos en HTML5 y no funcionan con todos los navegadores, y aunque funcionen, la presentación será distinta entre unos y otros. El navegador validará la entrada de estos campos antes de enviar los datos al servidor.

`<input type="email">`

Campo que solo admite una dirección de correo electrónico (e-mail) correcta.

`<input type="tel">`

Solo admite un teléfono en el formato correcto.

`<input type="url">`

Solo admite una URL en el formato correcto.

`<input type="number" min=0 max=10 step=1>`

Admite un número del 0 al 10; esto queda determinado por los valores min y max. En varios navegadores además muestra un control deslizante o slider. Step es el valor que se suma y resta al utilizar los botones del slider. Si queremos una caja de texto que admita números decimales tendremos que poner un valor decimal en la propiedad step (por ejemplo: step=0.01). Si cambias el campo edad como tipo number quedará así:

Edad:

`<input type="range" min=0 max=10>`

Genera un número del 0 al 10, pero no se visualiza como una caja de texto, sino como una barra con un deslizador.



Estructura de un documento HTML

En el [siguiente vídeo](#) te mostramos cómo crear la estructura de una página web para realizar una aplicación de gestión de tareas.

Puedes consultar el listado de etiquetas HTML en el siguiente enlace.

ETIQUETAS HTML

Resumen

Has terminado la lección, veamos los puntos más importantes que hemos tratado.

En esta unidad hemos realizado un recorrido por las principales etiquetas de HTML5. Gracias a estas etiquetas podremos dotar de contenido nuestros documentos web, aplicándoles la estructura idónea.

Es importante entender que con HTML5 las etiquetas solo se ocupan de estructurar la información y darle sentido semántico al contenido, siendo CSS el encargado de darle el aspecto visual adecuado.

Una de las principales dificultades que nos encontramos cuando empezamos nuestro estudio de esta nueva versión de HTML5, es el encontrar documentación y ejemplos con otras versiones anteriores, que incluyen atributos y etiquetas que han sido eliminadas en esta nueva versión.

Con HTML5, aunque tenemos etiquetas nuevas, como las semánticas de contenidos `<header>` `<section>`, etc. el principal cambio es la simplificación.

Al unir las tecnologías HTML, CSS y JavaScript podemos realizar de forma más adecuada muchas de las acciones que antes hacíamos con HTML.

unir LA UNIVERSIDAD
EN INTERNET | FORMACIÓN
PROFESIONAL

PROEDUCA