

Caso práctico

Después de asociarse, **María** y **Félix** esperan aumentar el número de clientes de su negocio teniendo presencia en Internet, para ello hablan con **Juan**, técnico superior en Administración de Sistemas Informáticos.

Juan les aconseja desarrollar una sencilla página web para la empresa, cuya funcionalidad puede ir aumentando a medida que aumenten sus necesidades. Por lo que el primer paso, será encargárselo a alguien que conozca el lenguaje HTML.

María le pregunta a **Juan** por su disponibilidad y capacidad para realizar dicha tarea. **Juan** le contesta que aprendió HTML en 1998. El boom de Internet le animó a interesarse por este campo y en este lenguaje realizó su primera página web. Desde entonces el lenguaje ha sufrido cambios en los que se ha ido actualizando.

Orientaciones para el alumnado

En esta unidad de trabajo estudiarás el lenguaje HTML y el uso de las hojas de estilos CSS. Al finalizar la unidad podrás elaborar páginas web sencillas.



[Ministerio de Educación y Formación Profesional](#). (Dominio público)

Materiales formativos de FP Online propiedad del Ministerio de Educación y Formación Profesional.

[Aviso Legal](#)

1.- Introducción a la utilización de lenguajes de marcas en entornos web.

Orientaciones para el alumnado

En este primer apartado de la UT2 aprenderás sobre la evolución del lenguaje HTML y las características del XHTML. Descubrirás cual es la estructura de un documento HTML y aprenderás a identificar las etiquetas y los atributos que forman el lenguaje. Por último conocerás los atributos comunes a todas las etiquetas HTML.

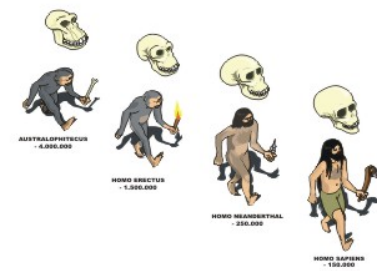
Para poder aprovechar todo el contenido de la unidad es recomendable que a la vez que lees los contenidos tengas preparado algún editor de HTML para ir probando los ejemplos que acompañan a las explicaciones.

1.1.- HTML: evolución y versiones.

HTML es el lenguaje utilizado para crear la mayor parte de las páginas web. Es un estándar reconocido en todos los navegadores, por lo tanto, todos ellos visualizan una página **HTML** de forma muy similar independientemente del sistema operativo sobre el que se ejecutan.

El origen de HTML fue un sistema de hipertexto para compartir documentos electrónicos en 1990. La primera propuesta oficial para convertir **HTML** en un estándar se realizó en 1993. Ninguna de las dos primeras propuestas de estándar que se hicieron (**HTML** y **HTML+**) consiguieron convertirse en estándares oficiales.

- ✓ **HTML 2.0** fue la primera versión oficial de HTML. El IETF publicó el estándar en septiembre de 1995.
- ✓ **HTML 3.2** se publicó el 14 de Enero de 1997, por el W3C. Incorporaba los applets de Java y texto alrededor de las imágenes.
- ✓ **HTML 4.0** se publicó el 24 de Abril de 1998. Entre las novedades que presentaba se encontraban las hojas de estilos CSS y la posibilidad de incluir pequeños programas en las páginas web.
- ✓ **HTML 4.01** se publicó el 24 de diciembre de 1999, como actualización de la versión anterior. En ese momento el W3C detuvo la actividad de estandarización de HTML hasta marzo de 2007, momento en que se retoma debido a la fuerza de las empresas que forman el grupo WHATWG y a la publicación de los borradores de **HTML 5.0**, que será la siguiente versión de este lenguaje.
- ✓ **HTML 5** es la versión más avanzada y la que se considera estándar actualmente, aunque ha ido evolucionando por diferentes especificaciones (HTML 5.1, HTML 5.2 y HTML 5.3). Es una versión viva en la que se sigue trabajando y su estándar actual (HTML 5.3) fue definido el 18 de octubre de 2018. Puedes encontrar las especificaciones de la última versión de HTML en el siguiente [enlace](#).



[ITE](#) (Uso Educativo NC)

Uno de los cambios más importantes en HTML 5 es que todo lo relativo a la presentación del documento se pasa del HTML a las hojas de estilo CSS. Es decir, cuestiones como colores, fondos, tamaño o posicionamiento se especifican mediante CSS. El HTML se encarga de la información que se quiere mostrar, su estructura y su semántica. Esto hace que muchos elementos y atributos de HTML 4 hayan quedado obsoletos (``, `<center>`, `align`,...).

1.2.- HTML y XHTML.

Caso práctico

Puesto que **Juan** ha aceptado realizar la página web se plantea el hacerla en HTML o en XHTML consulta con **Marina**, trabajadora de su empresa informática.

Marina opina que, desde un punto de vista formal, no hay diferencias sustanciales entre utilizar uno u otro lenguaje, y siguiendo la evolución lógica le parece que sería más apropiado utilizar XHTML y añade que este lenguaje tiene la ventaja de ser compatible con navegadores antiguos.

Juan opina que es una buena opción, pero que algunos navegadores, a pesar de ser compatibles con el lenguaje no interpretan los formatos.

El lenguaje **XHTML** (EXtensible HyperText Markup Language) es muy similar al lenguaje **HTML**. De hecho, no es más que una adaptación de **HTML** al lenguaje **XML**.

El estándar **XHTML** 1.0 sólo añade pequeñas mejoras y modificaciones menores al estándar **HTML 4.01**, por lo que este último está prácticamente incluido en el primero. Pasar del **HTML 4.01 Strict** a **XHTML** no requiere casi ningún cambio.

El lenguaje **HTML** tiene una sintaxis muy permisiva, por lo que es posible escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas, por ejemplo, pueden escribirse en mayúsculas, minúsculas e incluso combinando mayúsculas y minúsculas. El valor de los atributos de las etiquetas se pueden indicar con o sin comillas. Además, el orden en el que se abrían y cerraban las etiquetas no era importante.

La flexibilidad de **HTML** da lugar a páginas con un código desordenado, difícil de mantener y muy poco profesional.

XHTML soluciona estos problemas añadiendo ciertas normas en la forma de escribir las etiquetas y atributos. Son las normas que vimos en el capítulo 1 para que un fichero XML se considera bien formado. Así se consigue:

- ✓ Sencillez a la hora de editar y mantener el código.
- ✓ Al ser más regular, es más fácil escribir código que lo procese.
- ✓ Como es **XML** se pueden utilizar fácilmente herramientas creadas para procesar documentos **XML** genéricos (editores, **XSLT**, etc.).



Ana Polo Arozamena desde imágenes openclicart-0.18-full (Uso Educativo NC)

Diferencias sintácticas y estructurales con HTML

Para que el código HTML se pueda considerar XML bien formado, y por tanto, XHTML, tiene que cumplir:

- ✓ Toda la página debe estar contenida en un elemento raíz, `<html>`.
- ✓ Los nombres de las etiquetas y atributos siempre se escriben en minúsculas.
- ✓ El valor de los atributos, incluso los numéricos, siempre se encierra entre comillas.
- ✓ En los atributos en los que el nombre coincide con su valor, no puede darse el valor por entendido, es decir, no se pueden comprimir. Este tipo de atributos no son muy habituales.
- ✓ Todas las etiquetas deben cerrarse siempre. **XHTML** permite que en lugar de abrir y cerrar de forma consecutiva la etiqueta (`

`) se puede utilizar la sintaxis `
` para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto.

En general, los diseñadores web suelen trabajar con HTML. El XHTML es más apreciado por los desarrolladores, que aprecian la regularidad adicional. De cualquier manera, los tres primeros puntos de la anterior lista se consideran una buena práctica y se suelen cumplir siempre.

Por otro lado, hay que tener en cuenta que los navegadores no procesan HTML y XHTML exactamente igual. En concreto, en caso de errores para HTML el navegador intentará mostrar el mayor contenido posible. Pero si el documento se sirve como XHTML (con un tipo MIME XML), cualquier error de sintaxis causa que no se muestre el documento.

Para más información sobre HTML y XHTML, consulta [la sección correspondiente en la especificación](#) (inglés).

Autoevaluación

La aparición del lenguaje XHTML hace que deje de evolucionar el lenguaje HTML:

- ☐ Sí.
- ☐ No.

Respuesta errónea. Te recomiendo volver a leer el apartado.

Respuesta correcta.

Solución

1. Incorrecto
2. Opción correcta

La principal diferencia entre HTML y XHTML es:

- ☐ HTML permite usar las etiquetas en mayúsculas y XHTML no.
- ☐ XHTML da lugar a códigos ordenados.
- ☐ XHTML es compatible con XML.
- ☐ HTML es permisivo.

No es correcto.

No es la opción correcta.

Correcto.

Incorrecto.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

1.3.- Estructura de un documento HTML.

Caso práctico

Félix pregunta a **Juan** si existen grandes diferencias entre el XML y el HTML. **Juan** le explica que ambos lenguajes tienen origen en otro lenguaje que es el SGML y que sus diferencias son, principalmente, funcionales ya que la estructura del documento es semejante.

La estructura de una página HTML ha de ser coherente con la que hemos visto en el tema anterior para cualquier documento XML. Por ello tendrá un prólogo y un ejemplar.

Prólogo

Todo documento HTML ha de tener una declaración del tipo de documento donde se le indica al navegador el tipo de documento que se va a iniciar y la versión de HTML utilizada para la codificación del mismo y, además, le permite interpretarlo correctamente.

Para la versión HTML 4.0, hay tres prólogos distintos que definen tres tipos de documentos HTML:

- ✓ **HTML 4.0 Strict.** Es la DTD utilizada por defecto con HTML 4.0. En estos documentos no se permite el uso de los elementos declarados deprecated en otras versiones o Recomendaciones HTML. La declaración del tipo de documento correspondiente es:

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

- ✓ **HTML 4.0 Transitional.** Permite el uso de todos los elementos que permite el **HTML 4.0 Strict**, además de los elementos deprecated. La declaración del tipo de documento correspondiente es:

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

- ✓ **HTML 4.0 Frameset.** Es una variante de **HTML 4.0 Transitional** para documentos que usan frames. En estos documentos el elemento body hay que reemplazarlo por un elemento frameset. La declaración del tipo de documento correspondiente es:

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

En el caso de HTML5 solo existe una declaración de tipo de documento:

```
<!DOCTYPE html>
```

Ejemplar

En un documento HTML está delimitado por las etiquetas `<html>` y `</html>`. El ejemplar puede, a su vez dividirse en dos partes:

- ✓ **La cabecera**, delimitada por las etiquetas `<head>` y `</head>`. Contiene la información sobre el título de la página, el autor, palabras clave, etc. Dentro de esta sección es obligatorio definir el título del documento, para ello se usan las etiquetas `<title>` `</title>`. Esta información no se presentará en la ventana del navegador, salvo el título, que aparecerá en la barra de título de la parte superior.
- ✓ **El cuerpo**, contiene la información que se va a presentar en la pantalla. Está limitado por las etiquetas `<body>` y `</body>`, salvo en los documentos de tipo **HTML 4.0 Frameset** donde éstas se sustituyen por `<frameset>` y `</frameset>`.

Autoevaluación

¿Cuál de las siguientes afirmaciones es verdadera?

- ☐ Un documento HTML ha de tener título aunque no tenga cabecera.
- ☐ Un documento HTML ha de tener cabecera aunque no tenga título.
- ☐ Un documento HTML ha de tener título y cabecera.
- ☐ Un documento HTML ha de tener ejemplar aunque no tenga título.

Incorrecto.

No es la opción correcta.

Correcto.

No es correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

1.4.- Identificación de etiquetas y atributos HTML.

Caso práctico

María, tras escuchar que XML y HTML tienen el mismo origen, se interesa por si utilizan las mismas etiquetas.

Juan le contesta que en HTML las etiquetas y los atributos están definidos previamente, mientras que en XML los define el programador.

Un documento **HTML** está formado por etiquetas y atributos.

Al igual que en **XML** las etiquetas pueden ser de apertura, <etiqueta>, o de cierre, </etiqueta>. Una de las diferencias con **XML** es que la cantidad de etiquetas de **HTML** está limitada a aquellas que están definidas por el lenguaje.

Aunque **HTML** define una gran cantidad de etiquetas, estas no son suficientes para crear páginas complejas ya que la definición completa de ciertos elementos, como las imágenes y los enlaces, requiere información adicional. Como no es posible crear una etiqueta por cada elemento diferente, se añade la información adicional a las etiquetas mediante los atributos, dando lugar a los elementos.

Para cada uno de los atributos hay definido un conjunto de valores que se le puede asignar. Si el valor de un atributo no es válido, el navegador lo ignora.

Cada una de las etiquetas **HTML** define los atributos que puede utilizar, aunque algunos de ellos son comunes a muchas etiquetas.

Autoevaluación

Las etiquetas de HTML, al igual que las de XML, son ilimitadas:

- ☐ Verdadero.
- ☐ Falso.

No es correcto.

Correcto.

Solución

1. Incorrecto
2. Opción correcta

1.5.- Clasificación de los atributos comunes según su funcionalidad.

- ✓ **Atributos básicos o globales:** Se pueden usar en casi todas las etiquetas HTML.

Atributo	Descripción
<code>name = "texto"</code>	Permite asignar el nombre "texto" a un objeto HTML
<code>title = "texto"</code>	Asigna un título a un elemento HTML, mejorando así la accesibilidad. Dicho título es mostrado por los navegadores cuando el usuario pasa el ratón por encima del elemento. Es especialmente útil con los elementos: <code>a</code> , <code>link</code> , <code>img</code> , <code>object</code> , <code>abbr</code> y <code>acronym</code>
<code>id = "texto"</code>	Permite identificar al elemento HTML sobre el que se aplica de forma única mediante el identificador "texto". Sólo es útil cuando se trabaja con <u>CSS</u> y con <u>Javascript</u> . No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.
<code>style = "texto"</code>	Permite aplicar al elemento HTML el estilo "texto" directamente.
<code>class= "texto"</code>	Permite aplicar al elemento HTML el estilo "texto" definido en las CSS. No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.

- ✓ **Atributos para internacionalización:** Los utilizan las páginas que muestran sus contenidos en varios idiomas o aquellas que quieren indicar de forma explícita el idioma de sus contenidos

Atributo	Descripción																
dir	Indica la dirección del texto por lo que sólo puede tomar dos valores: ltr (left to right) de izquierda a derecha. Es el valor por defecto. rtl (right to left) de derecha a izquierda.																
lang = "código"	Especifica el idioma del elemento mediante un código predefinido. Los posibles valores de este atributo se encuentran en el documento RFC 1766 , algunos de los valores posibles son: <table><tr><th>Código</th><th>Idioma</th><th>Código</th><th>Idioma</th></tr><tr><td>en</td><td>Inglés (Gran Bretaña)</td><td>es</td><td>Español</td></tr><tr><td>en-US</td><td>Inglés americano</td><td>fr</td><td>Francés</td></tr><tr><td>ja</td><td>Japones</td><td>fr-CA</td><td>Francés de Canadá</td></tr></table>	Código	Idioma	Código	Idioma	en	Inglés (Gran Bretaña)	es	Español	en-US	Inglés americano	fr	Francés	ja	Japones	fr-CA	Francés de Canadá
Código	Idioma	Código	Idioma														
en	Inglés (Gran Bretaña)	es	Español														
en-US	Inglés americano	fr	Francés														
ja	Japones	fr-CA	Francés de Canadá														
xml:lang = "codigo"	Especifica el idioma del elemento mediante un código definido según la recomendación RFC 1766 .																

En las páginas **XHTML**, el atributo **xml:lang** tiene más prioridad que **lang** y es obligatorio incluirlo siempre que se incluye el atributo **lang**.

- ✓ **Atributos de eventos y atributos para los elementos que pueden obtener el foco:** Sólo se utilizan en las páginas web dinámicas creadas con JavaScript. Como no es nuestro objetivo no lo vamos a contemplar.

2.- Elementos HTML.

Introducción

Un elemento HTML está formado por:

- Una etiqueta de apertura.
- Cero o más atributos.
- Opcionalmente, un texto, encerrado por la etiqueta. No todas las etiquetas pueden encerrar texto.
- Una etiqueta de cierre. Para algunos elementos no hay etiqueta de cierre o es opcional.

Según el modo en que ocupan el espacio disponible en la página, los elementos pueden ser de dos tipos:

- **Elementos en línea.** Sólo ocupan el espacio necesario para mostrar sus contenidos. Su contenido puede ser texto u otros elementos en línea.
- **Elementos de bloque.** Los elementos de bloque siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea, aunque sus contenidos no lleguen hasta allí. Su contenido puede ser texto, elementos en línea u otros elementos de bloque.

El siguiente ejemplo muestra la diferencia entre ambos comportamientos

```
1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <title>Ejemplo de la diferencia entre los elementos en línea
6        y los elementos de bloque
7    </title>
8  </head>
9  <body>
10   <h1>Los encabezados son elementos de bloque.</h1>
11   <p>Y los párrafos también.</p>
12   <a href="#">Los enlaces son elementos de línea</a>
13   <p>Incluso si esta definido dentro de un párrafo, <strong>un texto resaltado</strong> sigue siendo un elemento en línea
14   </p>
15 </body>
16 </html>
```

Al publicarlo en un navegador quedaría así:

Los encabezados son elementos de bloque.

Y los párrafos también.

[Los enlaces son elementos de línea](#)

Incluso si esta definido dentro de un párrafo, **un texto resaltado** sigue siendo un elemento en línea.

Materiales educativos de la CAM (Uso educativo NC)

Autoevaluación

Los elementos de línea y de bloque se diferencian en:

- ☐ Los de línea sólo actúan sobre una línea de texto y los de bloque actúan sobre más de una línea.
- ☐ Los elementos de bloque pueden actuar como elementos de línea pero los de línea no pueden actuar como elementos de bloque.
- ☐ Los de línea ocupan el espacio imprescindible mientras que los de bloque no.
- ☐ Tanto los de línea como los de bloque ocupan el espacio imprescindible.

Incorrecto.

No es la respuesta correcta.

Correcto.

No es correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

Todos los elementos de HTML están formados por:

- ☐ Etiquetas de apertura y cierre.
- ☐ Etiquetas de apertura y cierre, un solo atributo y su valor correspondiente.
- ☐ Etiquetas de apertura, cierre, varios atributos y sus valores.
- ☐ Etiquetas de apertura, cierre, ninguno o varios atributos y sus valores.

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

2.1.- Elementos de la estructura básica de un documento.

Los elementos que definen la estructura básica de un documento son:

<html> - Define el inicio (y final con "</html>") de un documento HTML. Contiene a los elementos:

<head> - Define el inicio (y final con "</head>") de la cabecera del documento.

<body> - Define el inicio (y final con "</body>") del cuerpo del documento. Permite definir formatos que se aplican a los elementos de la página de manera global, como por ejemplo el color de fondo del texto o la anchura de los márgenes.

Un ejemplo de un documento HTML básico que utiliza estos elementos es:

```
1  <!DOCTYPE html>
2
3  <html>
4    <head>
5      <title>Ejemplo de la estructura de un documento HTML</title>
6    </head>
7    <body>
8      Aquí es donde se coloca la información que se quiere visualizar en el navegador.
9    </body>
10 </html>
```

Al publicarlo en un navegador, por ejemplo en Firefox, tendríamos:



Materiales educativos de la CAM (Uso educativo NC)

Autoevaluación

La etiqueta body encierra los elementos que van a determinar el formato del documento:

☐ Verdadero ☐ Falso

Verdadero

2.2.- Elementos de la sección de cabecera.

Se clasifican en dos tipos:

✓ Elementos contenedores:

Elemento	Descripción
title	Título del documento. Elemento obligatorio.
script	Script incrustado. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado.
style	Estilo aplicado al documento utilizando CSS. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado.

✓ Elementos no contenedores:

Elemento	Descripción
base	URI base del documento
isindex	Prompt de entrada de datos. (Eliminado de los estándares web aunque todavía algún navegador lo soporta)
link	Enlaces a documentos externos de librerías
meta	Metadatos sobre la página, como la codificación de caracteres, descripción o autores.

Ejemplo:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>El título es obligatorio</title>
6      </head>
7      <body>
8          ...
9      </body>
10 </html>
```

El título aparece en la barra del navegador y como nombre de la pestaña. El ejemplo anterior se vería así:



Para que los acentos se vean bien hay que guardar el ejemplo con la codificación indicado en el documento, UTF-8

2.3.- Encabezados y párrafos.

Para agrupar el texto en párrafos se usa el elemento `<p>`. Es un elemento de bloque.

```
<p>Texto del párrafo </p>
```

Para los encabezados, en HTML se definen 6 elementos:

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

Cuanto menor es el número, mayor es la importancia del encabezado. El texto marcado debe servir como encabezado a la sección en la que aparece. Se pueden utilizar para organizar jerárquicamente el contenido de la página.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Párrafos y encabezados</title>
6    </head>
7    <body>
8      <h1>Equipos</h1>
9      <p>Primer párrafo</p>
10     <p>Segundo párrafo</p>
11     <h2>Recién ascendidos</h2>
12     <p>...</p>
13     <h1>Jugadores</h1>
14     <p>...</p>
15   </body>
16 </html>
```

Se vería así:

Equipos

Primer párrafo

Segundo párrafo

Recién ascendidos

...

Jugadores

...

[Más ejemplos](#)

Si es necesario introducir un salto de línea se utiliza el elemento `
`. Para introducir múltiples espacios, se puede utilizar la entidad ` `, aunque también hay otras opciones con etiquetas HTML o propiedades CSS.

Esta página

Aquí hay espacios

Esto
introduce un salto de línea

2.3.2.- Comentarios.

Se pueden introducir comentario en los ficheros HTML así:

```
1 | <!-- comentario -->
```

También pueden ser multilínea:

```
1 | <!-- comentario  
2 |   de varias  
3 |   líneas -->
```

Los comentarios no son procesados por los navegadores, sirven para documentar el código.

2.4.- Semántica a nivel de texto.

Algunos elementos útiles son:

- ✓ `<a>`, para definir hipervínculos.
- ✓ ``, para representar que el texto marcado es importante.
- ✓ ``, para indicar énfasis.
- ✓ `
`, introduce un salto de línea.
- ✓ `<small>`, para comentarios accesorios, como lo que suelen aparecer en letra pequeña.

Elemento `<a>`

Este elemento hace que el texto encerrado entre las etiquetas sea un hipervínculo.

El atributo más importante es **href**, que indica la URL del vínculo. A veces se usa **href="#"** para referirse a la misma página en la que aparece el vínculo.

El atributo **target** permite elegir donde se abrirá el vínculo. Los valores más usados son:

- **target = "_blank"**, para que se abra en una ventana/pestaña nueva.
- **target = "_self"**, para que se abra en la misma. Es el valor por defecto.

Ejemplo:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Semántica a nivel de texto</title>
6    </head>
7    <body>
8      Texto marcado como <strong>importante</strong>.
9      <br>
10     Texto con <em>énfasis</em>
11     <br>
12     Texto marcado <small> con el elemento small </small>
13     <br>
14     Pulsa <a href="https://www.w3.org/">aquí</a> para ir a la página del W3C.
15   </body>
16 </html>
```

El resultado sería:

Texto marcado como **importante**.
Texto con *énfasis*
Texto marcado con el elemento small
Pulsa [aquí](https://www.w3.org/) para ir a la página del W3C.

Materiales educativos de la CAM (Uso educativo NC)

[Aquí](#) se puede consultar la lista completa de elementos para esta categoría. No todos tienen una representación visual en el navegador.

2.5.- Elementos de listas.

Hay tres tipos de listas: ordenadas, desordenadas y listas de definición.

Elemento	Descripción
ul	Delimita los elementos que forman una lista desordenada
ol	Delimita los elementos que forman una lista ordenada. Tiene varios atributos.
dl	Delimita los elementos que forman una lista de definición
li	Cada uno de los elementos de una lista ordenada o desordenada.
dt	Cada uno de los términos que se definen de una lista de definición.
dd	Cada una de las definiciones de una lista de definición.

Un ejemplo de un documento HTML que muestra la forma de utilizar estos elementos es:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Listas</title>
6    </head>
7    <body>
8      <h1>Ejemplo de lista desordenada: Módulos de 1º de ASIR</h1>
9      <ul>
10        <li>Fundamentos de Hardware</li>
11        <li>Gestión de Bases de Datos</li>
12      </ul>
13      <h1>Ejemplos de listas ordenadas: Módulos de 1º de ASIR</h1>
14      Comenzando en 1
15      <ol>
16        <li>Fundamentos de Hardware</li>
17        <li>Gestión de Bases de Datos</li>
18      </ol>
19      Comenzando en 4
20      <ol start = "4">
21        <li>Fundamentos de Hardware</li>
22        <li>Gestión de Bases de Datos</li>
23      </ol>
24      <h1>Ejemplo de lista de definición: Módulos de 1º de ASIR</h1>
25      <dl>
26        <dt>Fundamentos de Hardware</dt>
27        <dd>Componentes físicos de un ordenador</dd>
28        <dt>Gestión de Bases de Datos</dt>
29        <dd>Diseño y uso de bases de datos relacionales</dd>
30      </dl>
31    </body>
32  </html>
```

Que al publicarse en el navegador quedaría:

Ejemplo de lista desordenada: Módulos de 1º de ASIR

- Fundamentos de Hardware
- Gestión de Bases de Datos

Ejemplos de listas ordenadas: Módulos de 1º de ASIR

Comenzando en 1

- Fundamentos de Hardware
- Gestión de Bases de Datos

Comenzando en 4

- Fundamentos de Hardware
- Gestión de Bases de Datos

Ejemplo de lista de definición: Módulos de 1º de ASIR

Fundamentos de Hardware	Componentes físicos de un ordenador
Gestión de Bases de Datos	Diseño y uso de bases de datos relacionales

2.6.- Elementos de tablas.

Las tablas son una forma habitual de presentar información de manera compacta y fácil de entender para el que la lee. Aunque hay muchas posibilidades, en general las tablas muestran una serie de datos (columnas) comunes para varios elementos (filas). Como ejemplos, podemos pensar en un horario, o en una tabla comparativa de varios productos.

Los elementos para definir una tabla son los siguientes (no es necesario usar todos):

Elemento	Descripción
table	Delimita el contenido de una tabla.
tr	Delimita cada una de las líneas de la tabla.
td	Delimita el contenido de cada celda de la tabla.
colgroup	Permite agrupar columnas.
tbody	Permite agrupar líneas de la tabla.
thead	Define la línea cabecera de la tabla.
th	Delimita cada una de las celdas de la cabecera
tfoot	Define la fila pie de la tabla.
caption	Para añadir una leyenda a la tabla

En HTML las tablas están formadas por filas (elemento `<tr>`;) y estas a su vez por celdas. Las celdas pueden ser de datos (elemento `<td>`) o de cabecera (elemento `<th>`).

- El elemento `<table>` contiene al resto de elementos.
- Por cada fila de la tabla habrá un elemento `<tr>`. Este elemento contiene una serie de elementos `<td>` o `<th>` (uno por columna).
- Normalmente en la primera fila las celdas son elementos `<th>` y en el resto `<td>`.

Ejemplo de tabla básica

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>Tablas</title>
6      </head>
7      <body>
8          <table>
9              <caption>Tabla de socios</caption>
10             <tr>
11                 <th>Nombre</th>
12                 <th>Apellido</th>
13                 <th>Edad</th>
14             </tr>
15             <tr>
16                 <td>Juan</td>
17                 <td>Puertas</td>
18                 <td>54</td>
19             </tr>
20             <tr>
21                 <td>Eva</td>
22                 <td>Montes</td>
23                 <td>44</td>
24             </tr>
25         </table>
26     </body>
27 </html>
```

En el navegador se verá así:

Tabla de socios		
Nombre	Apellido	Edad
Juan	Puertas	54
Eva	Montes	44

Materiales educativos de la CAM (Uso educativo NC)

En [este tutorial](#) puedes ver ejemplos de tablas avanzadas.

NOTA IMPORTANTE: Aunque era habitual hace años, no es correcto usar tablas para la maquetación de la página.

2.7.- Elementos de formularios.

Los _____ formularios permiten recoger información que el usuario introduce en el navegador. Es importante validar estos datos, para detectar los errores en local (en el propio equipo), en caso de haberlos. De este modo se evita sobrecargar la red con datos erróneos y sobrecargar al servidor con tareas innecesarias. Normalmente se combinan los formularios con código **JavaScript**, que ayuda a realizar esas validaciones.

Ejercicio Resuelto

En la siguiente imagen puedes ver un ejemplo de un formulario con una muestra de algunos de los diferentes tipos de campos existentes.

Ejemplo de Formulario

Bienvenido al registro de usuarios de nuestro foro

Introduzca su nombre de usuario y contraseña

Usuario

Contraseña Repita Contraseña

Describe a continuación sus intereses:

Por último seleccione el sistema operativo de su ordenador:

Estoy conforme con la política de privacidad de la página ☒

Materiales educativos de la CAM (Uso educativo NC)

¿Cuál podría ser el código HTML asociado al documento que se muestra en la siguiente imagen?

Mostrar retroalimentación

Puedes ver el código que genera el formulario de la imagen en este [enlace](#) (Recuerda que, una vez se ha cargado una página en tu navegador, puedes visualizar el código que la generó. Basta con que pulses con el botón derecho sobre el fondo de la página y selecciones "Ver código fuente de la página").

2.7.1.- Declaración de formulario

La apertura y cierre de un formulario se hace mediante el elemento `form`.

La etiqueta permite especificar una serie de atributos para ajustar sus características. Los principales son:

- ✓ **name**— nombre del formulario
- ✓ **action**— Acción que se ejecuta cuando se pulsa el botón de enviar formulario
- ✓ **enctype**— Formato en el que se enviarán los valores del formulario, depende del contenido se enviarán de una u otra manera.
- ✓ **method**— Método de envío de la transmisión de datos, se puede elegir entre dos métodos **GET** y **POST**

```
<form name="miFormulario" action="/paginaQueEjecutaLaAccion.php" method="get">
```

La diferencia entre los métodos de envío **GET** y **POST** es la siguiente:

GET vs POST

GET	POST
<ul style="list-style-type: none">✓ Permite pasar valores en <u>ASCII</u> con un límite de 100 caracteres✓ Los valores de las variables que se transmiten se pueden ver en la <u>URL</u> y van concatenadas por el símbolo de AND (&) por ejemplo: <code>http://www.pagina.com/index.php?variable1=valor1&variable2=valor2&variable3=valor3</code>	<ul style="list-style-type: none">✓ Permite pasar valores de variables y otros elementos tales como archivos✓ Carece de restricciones de longitud como el método GET✓ Las variables y sus valores no son visibles en la <u>URL</u>

2.7.2.- Campos de formulario - input

Dentro de un formulario puede haber varios tipos de controles: campos de texto normales, campos para contraseñas, fechas o botones de radio, entre otros.

El elemento `<input>` (que no tiene etiqueta de cierre) se puede usar para varios tipos de control, según el valor que tome el atributo **type**. Por ejemplo, para **type="text"**, que es el valor por defecto, se obtiene un campo de texto. Con **type = "radio"**, un [botón de radio](#).

Además, de **type**, los atributos más importantes de la etiqueta son:

- ✓ **name** – nombre del campo
- ✓ **size** – número de caracteres visibles en el campo. Por defecto 20.
- ✓ **maxlength** – número máximo de caracteres que el usuario podrá introducir en el campo
- ✓ **value** – valor por defecto del campo de texto
- ✓ **placeholder** – valor sugerido, se presenta en color gris y desaparece al hacer foco con el cursor en el campo
- ✓ **readonly** – El valor del campo no puede ser modificado por el usuario
- ✓ **autofocus** – Sitúa el cursor del ratón en el campo una vez cargada la página
- ✓ **required** – No se podrá enviar el formulario hasta que el campo esté cumplimentado

Ejemplo:

- `<input type="text" name="usuario" size="30" maxlength="20" placeholder="Escriba aquí el nombre de usuario" required>`

Se vería en el navegador así:

Usuario

Materiales educativos de la CAM (Uso educativo NC)

2.7.3.- Campos de formulario - Área de texto

Permite recoger información abierta del usuario pero permitiendo un mayor número de caracteres. Se especifica mediante la etiqueta `<textarea>`

Para especificar un valor previo en el campo debe escribirse entre las etiquetas `<textarea>` de apertura y cierre

Atributos principales

- ✓ **name** - nombre del campo
- ✓ **rows** – número de filas. Reemplazable por la propiedad **CSS height**
- ✓ **cols** – número de columnas. Reemplazable por la propiedad **CSS width**

También puede utilizar algunos de los atributos especificados para input:

- ✓ **placeholder**
- ✓ **readonly**
- ✓ **autofocus**
- ✓ **maxlength**
- ✓ **required**

Por ejemplo, la siguiente línea de código **HTML**:

```
Describe a continuación sus intereses: <br> <textarea name="area"></textarea>
```

Quedaría en el navegador del siguiente modo:

Describe a continuación sus intereses:



2.7.4.- Campos de formularios - Lista desplegable

Permite que el usuario seleccione un valor entre diferentes opciones.

Etiquetas

- ✓ `<select>` para definir la lista
- ✓ `<option>` para cada una de las opciones

Atributos - `<select>`

- ✓ **name**– Nombre del elemento
- ✓ **size**– Número de elementos de la lista desplegable que se mostrarán
- ✓ **multiple**– Permite seleccionar varias opciones manteniendo pulsada la tecla **Ctrl** al seleccionarla

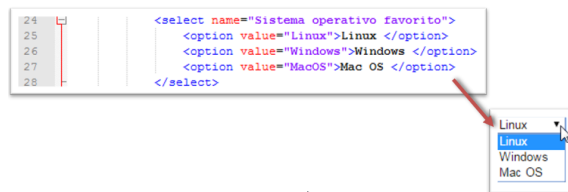
Atributos - `<option>`

- ✓ **selected**– campo que aparece seleccionado por defecto
- ✓ **value**– valor que se transmite cuando se envía el formulario

A continuación puedes encontrar un ejemplo de código HTML:

```
1 | Por último seleccione el sistema operativo de su ordenador:
2 | <select name="Sistema operativo favorito">
3 |   <option value="Linux">Linux </option>
4 |   <option value="Windows">Windows </option>
5 |   <option value="MacOS">Mac OS </option>
6 | </select>
```

En la siguiente imagen puedes observar cómo quedaría en un editor de **HTML** (en este caso **Notepad++**) y cuál es el resultado en el navegador.



2.7.5.- Campos de formulario - checkbox

Permite que el usuario indique si está conforme con una o varias opciones.

Etiqueta: `<input type="checkbox">`

Atributos

- ✓ **name**— obligatorio
- ✓ **checked**— el atributo aparece marcado
- ✓ **value**— valor que se transmitirá al enviar el formulario con ese campo seleccionado

Veamos un ejemplo de código HTML:

```
1 | Estoy conforme con la política de privacidad de la página
2 |
3 | <input type="checkbox" name="conforme" checked>
```

En el navegador se vería así:

Estoy conforme con la política de privacidad de la página ☒

Materiales educativos de la CAM (Uso educativo NC)

2.7.6.- Botones de formulario

2.7.6.1.- Botón de envío

Permite enviar el formulario a un destinatario, por ejemplo a una página determinada en un servidor web para que procese los datos o una dirección de correo electrónico. Todo dependiendo del valor del atributo **action** de la etiqueta de apertura de formulario.

Etiqueta: `<input type="submit">`

Atributos

- ✓ **value**— permite indicar el texto del botón

2.7.6.2.- Botón de anulación

Borra el contenido de los campos cumplimentados en el formulario

Etiqueta: `<input type="reset">`

Atributos

- ✓ **value**— permite indicar el texto del botón

Autoevaluación

Para crear un botón utilizaremos la etiqueta:

- ☐ `<button>`
- ☐ `<input>`
- ☐ Ambas.
- ☐ Ninguna.

Pero no sólo con esta.

Fíjate bien ya que hay una respuesta más correcta.

Correcto.

No es la opción correcta.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto



2.7.7.- Otros campos de formulario

Dependiendo de los valores que queramos recoger podemos utilizar definiciones de campos más apropiadas como las que se ven en la siguiente tabla.

Campo	Etiqueta	Notas
Oculto	<code><input type="hidden"></code>	
De contraseña	<code><input type="password"></code>	
Para envío de ficheros	<code><input type="file"></code>	Requiere formulario definido con <code>post</code> y <code>enctype="multipart/form-data"</code>
Para recoger correo electrónico	<code><input type="email"></code>	Valida formato automáticamente
Para recoger una URL	<code><input type="url"></code>	Valida formato automáticamente
Para recoger números enteros en un rango	<code><input type="number"></code>	Atributos: <code>max</code> – valor máximo <code>min</code> – valor mínimo <code>step</code> – incremento del contador

2.7.8.- Campos de formulario con formato de fecha

Dependiendo de la fecha que queramos recoger y sus características podemos utilizar diferentes variaciones en las etiquetas.

[`<input type="datetime-local">`](#)

Permite seleccionar día, mes, año y hora

[`<input type="date">`](#)

Permite seleccionar día, mes y año

[`<input type="month">`](#)

Permite seleccionar mes y año

[`<input type="week">`](#)

Permite seleccionar una semana

[`<input type="time">`](#)

Permite seleccionar una hora

Atributos

- ✓ **min**– Menor hora seleccionable
- ✓ **max**– Mayor hora seleccionable
- ✓ **step**– Incremento del contador del campo en segundos

2.7.9.- Campos de formulario - Rangos

Permite seleccionar un valor dentro de un rango.

Etiqueta: `<input type="range">`

Atributos

- ✓ **min**– valor mínimo del rango
- ✓ **max**– valor máximo del rango
- ✓ **step**– valor del incremento del contador
- ✓ **value**– valor inicial del contador

2.7.10.- Organización de formularios

Etiqueta <label>

Permite asociar a cada campo del formulario una etiqueta con su nombre. El texto mostrado entre las etiquetas **<label>** se muestra y constituye además una ayuda de usabilidad a personas invidentes.

```
1 | <label for="conforme">Acepto el acuerdo de licencia</label>
2 | <input type="checkbox" name="licencia" id="conforme" value="ok">
```

En el ejemplo la etiqueta **label** se asocia al campo tipo **input** mediante el atributo **for** que contiene el valor del identificador (**id**) de la etiqueta **input**. En el navegador se vería:

Acepto el acuerdo de licencia ☐

Materiales educativos de la CAM (Uso educativo NC)

Etiqueta <fieldset>

Agrupar los campos de formulario que estén entre la etiqueta de apertura y cierre **<fieldset>** y los rodea con un borde.

Etiqueta <legend>

Escrita inmediatamente a continuación de la etiqueta de apertura **<fieldset>** agrega un texto relacionado con los campos agrupados

2.7.11.- Un ejemplo de formulario

Ejercicio Resuelto

¿Podrías escribir el código HTML de este formulario?

Formulario

← → ↻ 🏠 https://fpdistancia.educa.madrid.org/pluginfile.php/95215/mod_scorm/content/1/LM1

Ejemplo de formulario: Matriculación en el ASIR

Nombre: Apellidos: Sexo: ☐ M ☐ H

Fecha de nacimiento: día mes año

Escoge los módulos en los que te matriculas:

<input type="checkbox"/> Fundamentos de Hardware	<input type="checkbox"/> Servicios de Red e Internet
<input type="checkbox"/> Gestión de Bases de Datos	<input type="checkbox"/> Administración de sistemas Operativos
<input type="checkbox"/> Implantación de Sistemas Operativos	<input type="checkbox"/> Implantación de Aplicaciones Web
<input type="checkbox"/> Planificación y Administración de Redes	<input type="checkbox"/> Administración de Sistemas Gestores de Bases de Datos
<input type="checkbox"/> Lenguajes de Marcas y Sistemas de Gestión de Información	<input type="checkbox"/> Seguridad y Alta Disponibilidad
<input type="checkbox"/> Formación y Orientación Laboral	<input type="checkbox"/> Empresa e Iniciativa Emprendedora
<input type="checkbox"/> Proyecto de Administración de Sistemas Informáticos en Red	<input type="checkbox"/> Formación en Centros de Trabajo

Estudios previos:

Pulsa el botón de "enviar" para formalizar la matrícula o el botón de "restablecer" para limpiar el formulario.

Materiales educativos de la CAM (Uso educativo NC)

Mostrar retroalimentación

En este [enlace](#) puedes ver un posible código que genera un formulario como el que aparece en la imagen inferior:

Caso práctico

Sin embargo verás que dicho código no sigue las reglas de organización de formularios que hemos indicado. ¿Podrías modificarlo? Si es así, entrégalo en esta [Actividad voluntaria](#).

2.8.- Multimedia.

Otros elementos habituales son:

Elemento	Descripción
img	Permite insertar una imagen en una página web.
audio	Para insertar audio
video	Para insertar vídeo
object	Para incrustar contenido multimedia (audio, vídeo, ficheros PDF, applets...)

Elemento

Este elemento se usa para insertar una imagen. No tiene etiqueta de cierre. Los atributos más importante son:

- ✓ **src**. Especifica la ruta de la imagen. Puede ser local, o una URL cualquiera. Los formatos soportados dependen de cada navegador, pero en general se pueden utilizar [JPG](#), [PNG](#) y [GIF](#), entre otros.
- ✓ **alt**. Texto alternativo. Se usa si no se puede cargar la imagen o para los [lectores de pantalla](#). Es obligatorio.
- ✓ **height**. Altura. Si no se especifica, se escoge el tamaño original. Si se especifica, se escala. Es de los pocos casos en HTML5 en que un atributo tiene información de representación.
- ✓ **width**. Anchura de la imagen. Como la anterior.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset='UTF-8'>
5     <title>Imágenes</title>
6   </head>
7   <body>
8     <p> Imagen en internet </p>
9     <img src='https://i.blogs.es/aa1b9a/luna-100mpx/450_1000.jpg' alt='la luna'>
10    <p>Imagen no encontrada</p>
11    <img src='http://rutaincorrecta.jpg' alt='la luna'>
12    <p>Imagen local</p>
13    <img src='index.jpeg' alt='logo'>
14  </body>
15 </html>
```

Se vería así:

Imagen en internet

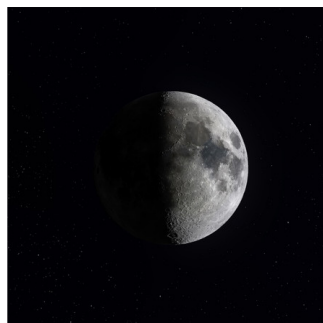


Imagen no encontrada

la luna

Imagen local



Para la segunda etiqueta ``, como no se encuentra la imagen, el navegador muestra el texto alternativo.

Para poder probar este ejemplo, descarga [esta imagen](#) y "guárdala como" *index.jpeg* en el mismo directorio que la el fichero html.

2.9.- Secciones y etiquetas semánticas.

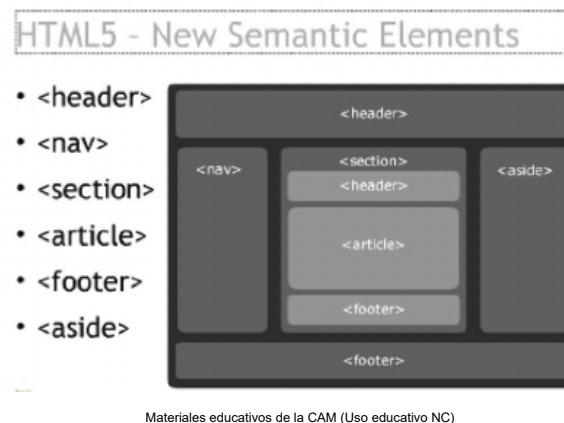
El elemento **<div>** se usa para agrupar otros elementos en secciones, tanto para organizar el contenido como para posicionarlo mediante hojas de estilo.

En **HTML5** aparecieron varias etiquetas semánticas para estructurar el contenido de la página, y por tanto solo se debería usar **<div>** cuando no haya una etiqueta más apropiada

Estas etiquetas son:

- ✓ **<header>** Contiene contenido introductorio para la sección de la página en que aparece. Es habitual que contenga los elementos de encabezado, h1,...,h6.
- ✓ **<aside>** Se utiliza para contenido parcialmente relacionado con el contenido principal. No tiene por qué mostrarse en un lateral.
- ✓ **<footer>** Contiene información sobre la sección correspondiente, como el autor. No tiene que estar necesariamente en la parte de inferior.
- ✓ **<section>** Una sección genérica dentro del documento.
- ✓ **<article>** Representa un elemento que se puede distribuir de manera independiente o reutilizable.
- ✓ **<nav>** Contiene vínculos, internos o externos. Suele usarse para la barra de navegación.

La siguiente imagen muestra una disposición habitual para estos elementos. Para obtenerla hay que usar CSS.



Para saber más

Lectura recomendada: [HTML semántico](#)

2.10.- Elemento iframe.

El elemento **iframe** permite integrar una página web dentro de otra. En ciertos aspectos, sustituye a la especificación de marcos (frameset) de HTML 4, ya obsoleta.

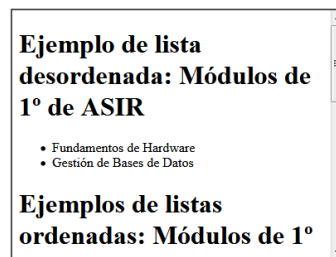
Atributos más importantes de **iframe**:

- ✓ **height**: Altura, por defecto en píxeles. Si no cabe todo el contenido, se usa una barra de desplazamiento. También se puede fijar con CSS.
- ✓ **width**: Anchura, como el anterior.
- ✓ **src**: La ruta al contenido inicial del **iframe**.
- ✓ **name**: Nombre, para referirse al **iframe** desde un vínculo.

A continuación, vamos a ver un ejemplo de uso habitual.

[Ejemplo de listas](#) [Ejemplo de tablas](#)

Aquí está el iframe



Materiales educativos de la CAM (Uso educativo NC)

El contenido del **iframe** cambia al seguir los vínculos. Para conseguirlo, en los vínculos hay que poner como valor del atributo **target** el valor del atributo **name** del **iframe** en que queramos que se cargue.

Es decir, para que se carguen en un **iframe** con **name = "contenido"**

se usará

```
<a href = ".." target = "contenido">...</a>
```

El código del ejemplo anterior sería el siguiente:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>iframe</title>
6    </head>
7    <body>
8      <a href = "listas.html" target = "contenido">Ejemplo de listas</a>
9      <a href = "tablas.html" target = "contenido">Ejemplo de tablas</a>
10     <p>Aquí está el iframe</p>
11     <iframe height = "300" width = "400" name = "contenido" src = "listas.html">
12   </body>
13 </html>
```

Inicialmente, el **iframe** carga "listas.html". Luego cambia entre "listas.html" y "tablas.html" al seguir los vínculos. Estos ficheros son los que se han utilizado en los ejemplos anteriores.

Para probar el ejemplo, tienes que tener los [tres ficheros](#) en el mismo directorio y abrir "iframe.html".

2.11.- Validación HTML.

Debes conocer

Es posible validar si nuestro código **HTML** cumple con la especificación de **HTML5**, para ello existen diferentes páginas en Internet que nos facilitan este servicio:

- ✓ **W3C** - <https://validator.w3.org/>
- ✓ **Validator.nu** - <https://html5.validator.nu/>

3.- Hojas de estilo o CSS.

Caso práctico

Juan muestra a **María** y **Félix** una primera versión de la página web.

María muestra su conformidad con la estructura y contenidos de la futura web corporativa que **Juan** ha realizado, aunque le gustaría probar otras gamas de colores y tipos de fuente.

Félix plantea que, quizás, resulte un trabajo demasiado laborioso ya que supondría modificar todos los ficheros de la web.

Juan, sonriendo, comenta que de hecho no es exactamente así. En realidad los datos están separados de sus formatos.

Introducción

CSS (Cascading Style Sheets) permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo.

Antes del uso de **CSS**, los diseñadores de páginas web debían definir el aspecto de cada elemento dentro de las etiquetas **HTML** de la página. El principal problema de esta forma de definir el aspecto de los elementos es que habría que definir el formato de cada uno de los elementos que formen la página, lo cual hace que sea muy difícil de actualizar.

CSS permite separar los contenidos de la página y su aspecto. Para ello se define en una zona reservada el formato de cada uno de los elementos de la web. Cualquier cambio en el estilo marcado para un elemento en la **CSS** afectará a todas las páginas vinculadas a ella en las que aparezca ese elemento. Las hojas de estilo están compuestas por una o más reglas de estilo aplicadas a un documento **HTML** o **XML**.

Al crear una página web, se utiliza en primer lugar el lenguaje **HTML/XHTML** para marcar los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, cabecera, texto destacado, etc. Una vez creados los contenidos, se utiliza el lenguaje **CSS** para definir el formato de cada elemento.

CSS obliga a crear documentos semánticos **HTML/XHTML**, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas **SGML**, alrededor del año 1970. Desde la creación de **SGML**, se observó la necesidad de definir un mecanismo que permitiera aplicar estilos a los documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos que tuvieran igual apariencia en distintos navegadores.

En 1995, el **W3C** añadió a su grupo de trabajo de **HTML** el desarrollo y estandarización de **CSS**.

CSS 1, se publicó en 1996, es la primera recomendación oficial.

CSS 2, publicada en 1998, es la segunda recomendación oficial.

CSS 3, empezó a desarrollarse en 1998, sus primeros borradores se publicaron en junio de 1999. A diferencia de las versiones anteriores la especificación se divide en varios documentos separados llamados módulos. Esto permite que diferentes módulos se encuentren en diferentes estados de desarrollo y no todos pasen a ser recomendaciones oficiales del **W3C**. Un módulo debe primero pasar por la fase de "candidato" antes de ser aprobado.

CSS 4, última versión sobre la que no existe una especificación integrada ya que al igual que **CSS 3** se encuentra dividido en diferentes módulos. A partir de la división en módulos que se hizo en **CSS 3** la evolución de cada uno de ellos es independiente lo que provoca que actualmente la mayoría de módulos estén en especificación de nivel 3 mientras que otros (por ejemplo [Selectores](#)) se encuentren en nivel 4.

En el siguiente enlace encontrarás el soporte que cada uno de los navegadores más actuales tienen sobre las etiquetas CSS.

[CSS Reference With Browser Support - W3shool](#)



[ITE](#) (Uso educativo NC)

Autoevaluación

Las hojas de estilo en cascada permiten:

- ☐ Definir formatos que se aplican sobre varias páginas web de un sitio.
- ☐ Separar el formato de la estructura de una página web.
- ☐ Estructurar el contenido de la página web.
- ☐ Ninguna respuesta es válida.

Correcto.

Incorrecto.

No es correcto.

No es la opción correcta.

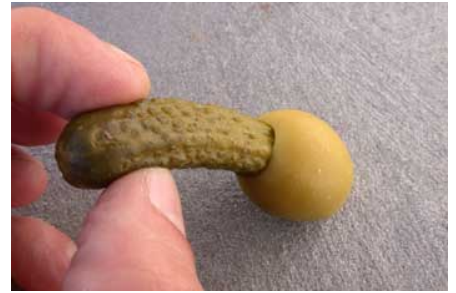
Solución

1. Opción correcta
2. Incorrecto
3. Incorrecto
4. Incorrecto

3.1.- Cómo incluir CSS en un documento HTML o XHTML.

Hay tres modos distintos de aplicar estilos CSS a los elementos de una página:

- ✓ **Declaración en línea:** se declara el estilo en la misma línea en que se va a aplicar. Esta opción está desaconsejada.
- ✓ **Declaración interna:** se declaran los estilos a emplear en la página, en el encabezado de dicha página, mediante la etiqueta **<style>**.
- ✓ **Declaración externa:** se declara la hoja de estilo que se va a emplear en la página, en el encabezado de dicha página, mediante la etiqueta **<link>**.



[ITE](#) (Uso educativo NC)

1.- Declaración en línea

Dentro de la propia etiqueta mediante el atributo **style**. Deben evitarse para preservar el principio de separación de contenidos y formato.

```
<p style="color: red;"> ... </p>
```

2.- Declaración interna

En el encabezado del documento dentro de las etiquetas **<style>**

```
<style type="text/css"> ... </style>
```

Ejemplo

El texto de los elementos **<p>** se mostrará en verde.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset='UTF-8'>
5      <title>CSS interna</title>
6      <style>
7        p {color:green}
8      </style>
9    </head>
10   <body>
11     <p>Hola</p>
12   </body>
13 </html>
```

2.3. Declaración en archivo externo

En el encabezado mediante la etiqueta **<link>** dentro del elemento **<head>**:

```
<link rel="stylesheet" type="text/css" href="rutaArchivo.css">
```

El archivo de estilos tendrá extensión **.css**. El código del archivo de estilos no tendrá etiqueta de declaración de estilo, por ejemplo:

```
h1 {background-color: #blue}
```

Otra forma de usar hojas de estilo externas es mediante la etiqueta **@import**. Es una directiva **CSS** no **HTML**.

```
1 | <style type="text/css">
2 |   @import url("formato1.css");
3 | </style>
```

Ejemplo

El siguiente [documento HTML](#) utiliza un archivo CSS externo:

```
1 | <!DOCTYPE html>
2 | <html>
3 |   <head>
4 |     <meta charset='UTF-8'>
5 |     <title>CSS externo</title>
6 |     <link rel="stylesheet" href="parrafo_verde.css">
7 |   </head>
8 |   <body>
9 |     <p>Hola</p>
10 |   </body>
11 | </html>
```

El contenido del [fichero CSS](#) es:

```
1 | p {
2 |   color: green;
3 | }
```

El texto de los elementos <p> se mostrará en verde.

Almacena los dos ficheros en la misma carpeta para probar el ejemplo. El fichero **CSS** se tiene que llamar "parrafo_verde.css".

Autoevaluación

El mejor modo de aplicar formatos a una página web es:

- ☐ Definiendo los formatos directamente a través de los atributos de los elementos HTML.
- ☐ Incluyendo el formato CSS en los elementos de HTML.
- ☐ Definiendo los estilos en la cabecera del documento HTML.
- ☐ Definiendo un fichero CSS externo.

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

3.1.1.- Elemento span.

Este elemento se utiliza habitualmente para dar estilo a texto no marcado.

Ejemplo

<p> Parte de este párrafo está en rojo

3.2.- Sintaxis de las reglas de estilo.

Cada uno de los estilos que componen una hoja de estilos **CSS** se denomina regla. Cada regla se forma por:

- ✔ ➤ **Selector:** indica el elemento o elementos **HTML** a los que se aplica la regla **CSS**
- **Llave de apertura, {**
- **Declaración:** especifica los estilos que se aplican a los elementos.
 - ** Propiedad:** permite modificar el aspecto de un atributo del elemento.
 - ** Valor:** indica el nuevo valor del atributo modificado en el elemento.
- **Llave de cierre, }.**

```
p { color: blue; }
```

En este caso el **selector** es "p" (párrafo), la **declaración** es "color: blue" donde encontramos la **propiedad** color y el **valor** blue.

Un archivo **CSS** puede contener infinitas reglas **CSS**, cada regla puede contener varios selectores y cada declaración puede estar formada por diferentes declaraciones.

Autoevaluación

El elemento HTML sobre el que se aplica un estilo se especifica:

- ☐ En etiquetas, es decir, entre < y >.
- ☐ Entre paréntesis.
- ☐ Entre llaves.
- ☐ No va encerrado entre signos.

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

3.3.- Cascada y herencia de estilos.

En ocasiones y dependiendo de cómo se haya hecho la definición de estilos es posible que el navegador se encuentre estilos contradictorios. Ante esta situación el navegador aplicará la siguiente precedencia

- 1.- Declaración en línea
- 2.- Declaración interna
- 3.- Declaración externa
- 4.- Propiedades por defecto del navegador

Por otro lado las hojas de estilo también permiten la herencia de propiedades, Si tenemos varios elementos **HTML** anidados los elementos más internos heredan los estilos de los externos en los que están anidados siempre y cuando ellos no los tengan definidos

3.4.- Selectores.

A la hora de aplicar estilos a nuestros elementos **HTML** necesitamos un mecanismo que permita identificar sobre cuál o cuáles de estos elementos queremos actuar. Para ello se utilizan los **selectores**, estos permiten identificar a qué elementos de nuestro código **HTML** vamos a aplicar el estilo definido. Existe diferentes selectores, a continuación veremos los más importantes.

Selector universal (*)

Sirve para seleccionar todos los elementos de la página.

```
* {margin:10px; padding: 5px};
```

Selectores de etiqueta

En este caso los estilos se aplican solo a la etiqueta.

```
p { text-align: center}
```

En este ejemplo, los párrafos serán alineados al centro.

Si queremos ajustar los mismos estilos a dos etiquetas diferentes podemos ponerlos separados por comas.

```
p, h1, h2 { text-align: center}
```

En este ejemplo, tanto los párrafos como los encabezados de tipo 1 y 2 serán alineados al centro.

3.4.1.- Selectores de clase.

En ocasiones no vamos a querer aplicar el mismo formato en todas las etiquetas del mismo tipo. Para ello podemos asignar a las etiquetas a una **clase** determinada, esto requiere que identifiquemos las etiquetas afectadas mediante el atributo **class** de la forma.

```
<p class="parrafoCentrado"> ... </p>
```

El selector se especificaría:

```
p.parrafoCentrado {text-align: center; }
```

De manera que este estilo se aplicaría a todas las etiquetas **p** que tengan el atributo **class** a valor parrafoCentrado.

Sin embargo si hacemos la misma definición pero omitimos el identificador de etiqueta:

```
.parrafoCentrado { text-align: center; }
```

Se aplicaría a todas las etiquetas que perteneciesen a la clase "parrafoCentrado" sean del tipo que sean. Por ejemplo a una que fuese:

```
<h1 class="parrafoCentrado">encabezado 1 centrado</h1>
```

Veamos a continuación un código de ejemplo:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset='UTF-8'>
5     <title>Ejemplo del uso de clases en hojas de estilo</title>
6     <style type="text/css">
7       .clase_azul{color:blue}
8       p.clase_roja{color:#ff0000; font-style:italic; font-weight:bold; font-family:courier;}
9     </style>
10  </head>
11  <body>
12    <h3 class="clase_azul">Ejemplo del uso de clases en hojas de estilo</h3>
13    <p>Cualquier elemento sobre el que apliquemos la clase clase_azul tendrá el texto azul.</p>
14    <p class="clase_azul"> Incluso el párrafo.</p>
15    <p class="clase_roja">Sobre el párrafo podemos aplicar la clase clase_roja y el texto será rojo, en negrita cursiva y la
16    <h3 class="clase_roja"> Pero este texto no aparecerá formateado ya que regla de la clase clase_roja solo actúa sobre párr
17  </body>
18 </html>
```

El resultado en el navegador sería:

Ejemplo del uso de clases en hojas de estilo

Cualquier elemento sobre el que apliquemos la clase clase_azul tendrá el texto azul.

Incluso el párrafo.

Sobre el párrafo podemos aplicar la clase clase_roja y el texto será rojo, en negrita cursiva y la familia del texto courier.

Pero este texto no aparecerá formateado ya que regla de la clase clase_roja solo actúa sobre párrafos.

3.4.2.- Selectores de ID.

Permite seleccionar un elemento de la página por medio de su atributo **id**. El uso de este selector se suele asociar a elementos de estilo que se van a aplicar de manera excepcional una única vez, por ello el valor del atributo **id** no debe repetirse en dos elementos diferentes de la misma página.

Por ejemplo:

```
<p id="unico">...</p>
```

Se referencia mediante una de las siguientes posibilidades:

```
#unico { color: blue;}
```

```
p#unico {color: blue;}
```

3.4.3.- Selectores descendentes.

Permite seleccionar elementos que se encuentran dentro de otros elementos, por ejemplo:

```
p h1 {color: red; }:
```

Se aplicará a todas las etiquetas **h1** que estén dentro de bloques **p**. Además hay que tener en cuenta que:

- ✓ No tiene porqué ser descendiente directo.
- ✓ El nivel de anidación puede tener varios niveles, por ejemplo:

```
p a b i {text-decoration: underline; }
```

Se aplica a los elementos en cursiva, dentro de etiquetas de negrita, anidados dentro de enlaces que se encuentren en párrafos.

Es importante tener en cuenta que si no tenemos cuidado podemos confundir el selector descendente con la aplicación del mismo selector a distintas etiquetas

```
p a b i {color: blue; }
```

¡NO ES LO MISMO QUE...!

```
p,a,b,i {color: blue;}
```

También podemos combinar el selector universal con selectores descendentes, por ejemplo:

```
p * b {color: #0000FF;}
```

Se aplica a todas las etiquetas de tipo **** que estén anidadas en cualquier otra etiqueta que a su vez esté dentro de una etiqueta de tipo **<p>**.

Sin embargo no se aplicará a las etiquetas de tipo **** que estén dentro de una etiqueta de tipo **<p>** directamente.

3.4.4.- Selector hijo.

Es parecido al anterior, pero solo afecta al primer nivel de anidamiento.

Ejemplo

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset='UTF-8'>
5      <title>Selector hijo</title>
6      <style>
7        section > p {color:red}
8      </style>
9    </head>
10   <body>
11     <p>Párrafo inicial</p>
12     <section>
13       <p>Párrafo hijo de sección</p>
14       <article>
15         <p>Párrafo nieto de sección</p>
16       </article>
17     </section>
18   </body>
19 </html>
```

Se vería así:

Párrafo inicial

Párrafo hijo de sección

Párrafo nieto de sección

3.5.- Propiedades principales.

En los siguientes subapartados vamos a ver las propiedades principales que se usan en CSS como son:

- ✓ Propiedades de color y fondo.
- ✓ Propiedades de fuente.
- ✓ Propiedades de texto.
- ✓ Propiedades de caja.
- ✓ Propiedades de clasificación.

Pasemos a verlos detenidamente.



[ITE](#) (Uso educativo NC)

3.5.1.- Propiedades de color y fondo.

Las propiedades de color y fondo son los que enumeramos a continuación:

Elemento	Descripción
color	Indica el color del texto. Lo admiten casi todas las etiquetas de HTML . El valor de este atributo es un color, con su nombre o su valor RGB .
background-color	Indica el color de fondo del elemento. El valor de este atributo es un color, con su nombre o su valor RGB.
background-image	Permite colocar una imagen de fondo del elemento. El valor que toma es el nombre de la imagen con su camino relativo o absoluto
background-repeat	Indica si ha de repetirse la imagen de fondo y, en ese caso, si debe ser horizontal o verticalmente. Los valores que puede tomar son: <code>repeat-x</code> , <code>repeat-y</code> o <code>no-repeat</code> .
background-attachment	Especifica si la imagen ha de permanecer fija o realizar un scroll. Los valores que pueden tomar son: <code>scroll</code> o <code>fixed</code> .
background-position	Es una medida, porcentaje o el posicionamiento vertical u horizontal con los valores establecidos que sirve para posicionar una imagen. Los valores que puede tomar son: porcentaje, tamaño, o [<code>top</code> , <code>center</code> , <code>bottom</code>] [<code>left</code> , <code>center</code> , <code>right</code>]
background	Establece en un solo paso cualquiera de las propiedades de <code>background</code> anteriores. Los valores que puede tomar son: <code>background-color</code> , <code>background-image</code> , <code>background-repeat</code> , <code>background-attachment</code> , <code>background-position</code> .

Dado que no todos los nombres de colores son admitidos en el estándar, es aconsejable utilizar el valor RGB.

Un ejemplo de un documento [XHTML](#) en el que se utiliza este método para incluir formatos es:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Ejemplo de atributos CSS de color y fondo</title>
5          <style type="text/css">
6              body { background-color: black; color:yellow; }
7              p { color: #ffffff;}
8          </style>
9      </head>
10     <body>
11         <h3>Ejemplo del uso de atributos de color y fondo</h3>
12         <p>El texto de cualquier elemento, salvo el del párrafo que es blanco, es amarillo y el fondo negro.</p>
13     </body>
14
15 </html>
```

En el navegador se ve del siguiente modo:

Ejemplo del uso de atributos de color y fondo

El texto de cualquier elemento, salvo el del párrafo que es blanco, es amarillo y el fondo negro.

Materiales educativos de la CAM (Uso Educativo No Comercial)

Para saber más

Accede al tutorial del uso de color de [w3schools.com](https://www.w3schools.com)

<https://www.w3schools.com/colors/default.asp>

Autoevaluación

Cuál de los siguientes valores del atributo color no es válido:

- ☐ `rgb(33,33,0).`
- ☐ `red.`
- ☐ `#f06.`
- ☐ `#ff06.`

No es la opción correcta. Comprueba que es una opción correcta RGB.

No es correcto. Comprueba que es un nombre de color válido.

Incorrecto. Equivale a la opción hexadecimal `#ff0066`. Es una forma comprimida de indicar el color.

Correcto. Ya que sólo se indican 4 símbolos hexadecimales cuando tendrían que ser 6 o si usamos la forma comprimida 3.

Solución

1. Incorrecto
2. Incorrecto
3. Incorrecto
4. Opción correcta

3.5.2.- Propiedades de fuente.

En este apartado vamos a ver las distintas propiedades que podemos utilizar referentes a las fuentes de nuestro documento y que son:

Elemento	Descripción
<u>font-size</u>	Indica el tamaño de la fuente. Puede ser un tamaño absoluto, relativo o en porcentaje. Toma valores de unidades de CSS
<u>font-family</u>	Establece la familia a la que pertenece la fuente. Si el nombre de una fuente tiene espacios se utilizan comillas para que se entienda bien. El valor es el nombre de la familia fuente.
<u>font-weight</u>	Define el grosor de los caracteres. Los valores que puede tomar son: <code>normal</code> , <code>bold</code> , <code>bolder</code> , <code>lighter</code> , <code>100</code> , <code>200</code> , <code>300</code> , <code>400</code> , <code>500</code> , <code>600</code> , <code>700</code> , <code>800</code> o <code>900</code>
<u>font-style</u>	Determina si la fuente es normal o cursiva. El estilo <code>oblique</code> es similar al cursiva. Los valores posibles son: <code>normal</code> , <code>italic</code> , <code>oblique</code> .
<u>font-variant</u>	Determina si la fuente es normal o mayúsculas pequeñas. Los valores que puede tomar son: <code>normal</code> , <code>small-caps</code>
<u>line-height</u>	El alto de una línea y por tanto, el espaciado entre líneas. Es una de esas características que no se podían modificar utilizando HTML.
<u>font</u>	Permite establecer todas las propiedades anteriores en el orden que se indica a continuación: <code>font-style</code> , <code>font-variant</code> , <code>font-weight</code> , <code>font-size[line-height]</code> , <code>font family</code> . Los valores han de estar separados por espacios. No es obligatorio el uso de todos los valores.

Un ejemplo de un documento XHTML en el que se utiliza este método para incluir formatos es:

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Ejemplo de atributos CSS de fuente</title>
5      <style type="text/css">
6        body { background-color: black; color:yellow; font-family: courier }
7        p { color: #ffffff; font:italic 900 12px Verdana; }
8      </style>
9    </head>
10   <body>
11     <h3>Ejemplo del uso de atributos de fuente</h3>
12     <p>El texto de cualquier elemento es de la familia Courier y amarillo, salvo el del párrafo que es Verdana, blanco y de tama
13   </body>
14 </html>
```

En el navegador el resultado es:

Ejemplo del uso de atributos de fuente

El texto de cualquier elemento es de la familia Courier y amarillo, salvo el del párrafo que es Verdana, blanco y de tamaño 12 px.

3.5.3.- Propiedades de texto.

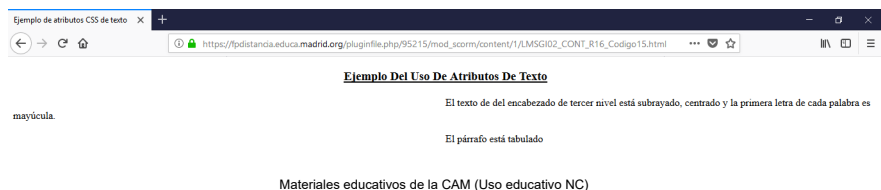
En el apartado anterior vimos los atributos relacionados con las fuentes y en este vamos a ver los relacionados con el texto en sí y son los siguientes:

Elemento	Descripción
text-decoration	Establece si el texto está subrayado, sobrerayado o tachado. los valores que puede tomar son: <code>none</code> , <code>underline</code> , <code>overline</code> , <code>line-through</code> o <code>blink</code>
text-align	Indica la alineación del texto. Aunque las hojas de estilo permiten el justificado de texto no funciona en todos los sistemas. Los valores que puede tomar son: <code>left</code> , <code>right</code> , <code>center</code> o <code>justify</code>
text-indent	Determina la tabulación del texto. Los valores que toma son una longitud, en unidades <code>CSS</code> , o un porcentaje de la establecida.
text-transform	Nos permite transformar el texto, haciendo que tenga la primera letra en mayúsculas de todas las palabras, todo en mayúsculas o minúsculas. Los valores que puede tomar son: <code>capitalize</code> , <code>uppercase</code> , <code>lowercase</code> o <code>none</code>
word-spacing	Determina el espaciado entre las palabras. Los valores que puede tomar es un tamaño.
letter-spacing	Determina el espaciado entre letras. Los valores que puede tomar es un tamaño.
vertical-align	Establece la alineación vertical del texto. Sus valores posibles son: <code>baseline</code> , <code>sub</code> , <code>super</code> , <code>top</code> , <code>text-top</code> , <code>middle</code> , <code>bottom</code> , <code>text-bottom</code> o un porcentaje.
line-height	Altura de la línea. Puede establecerse mediante un tamaño o un porcentaje

Un ejemplo de un documento `XHTML` en el que se utiliza este método para incluir formatos es:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Ejemplo de atributos CSS de texto</title>
5     <style type="text/css">
6       h3 { text-decoration:underline; text-align: center; text-transform: capitalize }
7       p { text-indent: 50%; }
8     </style>
9   </head>
10  <body>
11    <h3>Ejemplo del uso de atributos de texto</h3>
12    <p>El texto de del encabezado de tercer nivel est&aacute; subrayado, centrado y la primera letra de cada palabra es may&uacute;lcula.
13    <p>El p&aacute;rrafo est&aacute; tabulado</p>
14  </body>
15 </html>
```

Su aspecto en el navegador es el siguiente:



3.5.4.- Propiedades de lista.


Hay cuatro propiedades de estilo para listas.


Elemento	Descripción
list-style-type	Indica cual es el símbolo que se utiliza como marcador en las listas. Valores que puede tomar son: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none.
list-style-image	Permite utilizar el uso de una imagen como marcador en una lista. El valor que toma es la ruta del fichero imagen
list-style-position	Determinan la posición del marcador en una lista. Puede tomar los valores: outside o inside.
list-style	Permite establecer de una única vez todas las características de una lista. Hay que seguir el orden siguiente: list-style-type, list-style-position y list-style-image.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset='UTF-8'>
5     <title>Estilo para listas</title>
6     <style>
7       #flecha{ list-style-image: url("flecha.png") }
8       .circ{ list-style-type: circle }
9       #armenio{ list-style-type: armenian }
10    </style>
11  </head>
12  <body>
13    <p>Lista con imagen</p>
14    <ul id="flecha">
15      <li>Patatas</li>
16      <li>Peras</li>
17    </ul>
18    <p>Lista con círculo</p>
19    <ul class="circ">
20      <li>Patatas</li>
21      <li>Peras</li>
22    </ul>
23    <p>Alfabeto armenio</p>
24    <ol id="armenio" reversed>
25      <li>Peras</li>
26      <li>Manzanas</li>
27    </ol>
28  </body>
29 </html>
```


Que en el navegador se ve del siguiente modo:


Lista con imagen

 Patatas


 Peras


Lista con círculo

 Patatas

 Peras

Alfabeto armenio

 . Peras

 . Manzanas

3.5.5.- Propiedad display.

Esta propiedad se puede usar para:

- ✓ Hacer que un elemento sea de bloque o de línea.
- ✓ Ocultarlo o hacerlo visible. Normalmente, a través de JavaScript.

Ejemplo

En este ejemplo, hay varios vínculos, que son elementos de línea. Pero hay una regla CSS que hace que los que tienen clase "especial" sean de bloque.

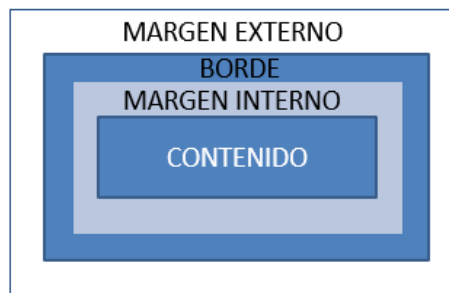
```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Ejemplo de atributos CSS de fuente</title>
5      <style>
6        .especial { display: block}
7      </style>
8    </head>
9    <body>
10     <a href="#">Primer vínculo (normal)</a>
11     <a href="#">Segundo vínculo (normal)</a>
12     <a href="#" class="especial">Tercer vínculo (especial)</a>
13     <a href="#" class="especial">Cuarto vínculo (especial)</a>
14   </body>
15 </html>
```

Se verá así:

[Primer vínculo \(normal\)](#) [Segundo vínculo \(normal\)](#)
[Tercer vínculo \(especial\)](#)
[Cuarto vínculo \(especial\)](#)

3.6.- Avanzado: Modelo de cajas.

La W3C define lo que se denomina "Modelo de caja", que no es más que una zona rectangular como la siguiente que rodea cada uno de los elementos de nuestra página web.



Materiales educativos de la CAM (Uso educativo NC)

Cada etiqueta **HTML** aplica ese modelo y por lo tanto tiene:

- ✓ **Contenido**. Texto de la etiqueta
- ✓ **Margen interior**. Distancia desde el contenido al borde del elemento. Propiedad **HTML** padding.
- ✓ **Borde**. El borde del elemento. Propiedad **HTML** border.
- ✓ **Margen exterior**. Distancia desde el borde del elemento a los elementos adyacentes. Propiedad **HTML** margin.

Cada uno de esos elementos puede definirse mediante propiedades **CSS** que veremos a continuación.

3.6.1.- Propiedades de caja.

A continuación veremos algunas de las propiedades que afectan a cada uno de los elementos del modelo de caja.

Contenido

FORMATO	PROPIEDAD	VALORES
anchura	width	auto Longitud en px o equivalente
altura	height	auto Longitud en px o equivalente

Tanto **width** como **height** especifican las dimensiones del contenido sin tener en cuenta borde y márgenes.

Margen interno

FORMATO	PROPIEDAD	VALORES
Relleno / margen interno	padding padding-top padding-bottom padding-right padding-left	auto valor de longitud valor de porcentaje

Si utilizamos **padding** con un solo valor se aplica a los cuatro lados, con dos valores el primero se aplica a superior e inferior y el segundo a laterales, con tres valores se aplica el primero al superior, el segundo a los laterales y el tercero al inferior, con cuatro valores se aplica a superior, derecho, inferior, izquierdo.

También es posible fijar el valor de cada una de los cuatro valores independientemente con la propiedad correspondiente.

borde

FORMATO	PROPIEDAD	VALORES
Color del borde	border-color border-top-color border-bottom-color border-right-color border-left-color	Color en alguna de las notaciones permitidas transparent
Grueso del borde	border-width border-top-width border-bottom-width border-right-width border-left-width	Valor de longitud thin médium thick
Estilo del borde	border-style border-top-style border-bottom-style border-right-style border-left-style	Solid dashed dotted double ridge Groove inset outset hidden none

Si utilizamos **border-color**, **border-width** o **border-style** con un solo valor se aplican a los cuatro lados, con dos valores el primero se aplica a superior e inferior y el segundo a laterales, con tres valores se aplica el primero al superior, el segundo a los laterales y el tercero al inferior, con cuatro valores se aplica a superior, derecho, inferior, izquierdo.

También es posible fijar el valor de cada una de los cuatro valores independientemente con la propiedad correspondiente.

Margen externo

FORMATO	PROPIEDAD	VALORES
Ancho del margen externo	margin margin-top margin-bottom margin-right margin-left	auto valor de longitud valor de porcentaje

Si utilizamos **margin** con un solo valor se aplica a los cuatro lados, con dos valores el primero se aplica a superior e inferior y el segundo a laterales, con tres valores se aplica el primero al superior, el segundo a los laterales y el tercero al inferior, con cuatro valores se aplica a superior, derecho, inferior, izquierdo.

También es posible fijar el valor de cada una de los cuatro valores independientemente con la propiedad correspondiente.

Autoevaluación

Para modificar el tamaño del borde izquierdo de una caja o utilizar el atributo:

- ☐ border-left-width
- ☐ border-width
- ☐ Ambas.
- ☐ Ninguna.

Incorrecto.

No es la opción correcta.

Correcto.

No es correcto.

Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta
4. Incorrecto

3.6.2.- Unidades de tamaño.

A la hora de especificar tamaños CSS nos permite usar diferentes tipos de unidades, estas además pueden ser absolutas o relativas:

- ✓ **Absolutas.** Cualquier longitud expresada en una de estas unidades siempre se mostrará del mismo tamaño.
- ✓ **Relativas.** Dependen del tamaño de otro elemento.

Normalmente se utilizan unidades absolutas para la secciones y la estructura de la página web porque se adaptan mejor a diferentes tamaños de pantalla. Las más habituales son los porcentajes y la unidad **em**.

Para las dimensiones de imágenes y vídeos se suelen utilizar píxeles. Es la única unidad absoluta que se utiliza habitualmente.

Se debe especificar la unidad usada:

- ✓ Por ejemplo si queremos especificar un tamaño de 13 píxeles utilizaremos: **13px**

Unidades absolutas	
cm	Centímetros
mm	Milímetros
in	Pulgadas
px *	Píxeles
pt	Puntos
pc	Picas

*: Los píxeles son relativos a la pantalla.

UNIDADES RELATIVAS	
em	Relativa al tamaño del tipo de letra por defecto
porcentajes (%)	Relativos a las dimensiones del elemento contenedor
ex	Relativa al valor de x-height de la fuente actual
ch	Relativa al ancho del cero "0"
rem	Relativa al tamaño de letra del elemento raíz

Ejemplo de porcentajes (1)

En esta página se crean dos secciones. La primera tiene fondo verde y su anchura la mitad que el elemento contenedor, **<body>**, que al ser un elemento de bloque ocupa toda la anchura disponible.

Las dos secciones también ocuparían toda la anchura si no fuera por las reglas de estilo.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Porcentajes</title>
6     <style>
7       #s1{
```



```

8         background-color:green;
9         width:50%;
10    }
11    #s2{
12        background-color:blue;
13        width:30%;
14    }
15    </style>
16    </head>
17    <body>
18        <section id = "s1">Contenido sección 1</section>
19        <section id = "s2">Contenido sección 2</section>
20    </body>
21 </html>

```

Se vería así:



Materiales educativos de la CAM (Uso educativo NC)

Ejemplo de porcentajes (2)

Con la altura (**height**) ocurre lo mismo, pero hay que tener en cuenta que depende del contenido que haya en la página, a no ser que se fije un valor para el elemento como en este ejemplo. El elemento **<body>** tiene una altura fijada en píxeles. Las secciones tienen la altura y anchura expresadas en porcentajes relativos a los del elemento **<body>**.

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="UTF-8">
5          <title>Porcentajes</title>
6          <style>
7              body{
8                  height:800px;
9              }
10             #s1{
11                 background-color:green;
12                 width:50%;
13                 height:40%;
14             }
15             #s2{
16                 background-color:blue;
17                 width:30%;
18                 height:20%;
19             }
20         </style>
21     </head>
22     <body>
23         <section id = "s1">Contenido sección 1</section>
24         <section id = "s2">Contenido sección 2</section>
25     </body>
26 </html>

```

Se vería así:



Materiales educativos de la CAM (Uso educativo NC)

Ejemplo de unidades em (3)

Para las fuentes es habitual utilizar la unidad **em**, que hace referencia al tamaño de la fuente actual.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Unidades em</title>
6      <style>
7        .grande {font-size: 2em;}
8        .muyGrande {font-size: 4em}
9      </style>
10   </head>
11   <body>
12     <p>Normal</p>
13     <p class = "grande">Grande</p>
14     <p class = "muyGrande">Muy grande</p>
15   </body>
16 </html>
```

Se vería así:

Normal

Grande

Muy grande

Materiales educativos de la CAM (Uso educativo NC)

Autoevaluación

Las unidades relativas especifican el tamaño en relación al tamaño de una letra determinada que escoge el programador:

☐ Verdadero.

☐ Falso.

Incorrecto.

Correcto.

Solución

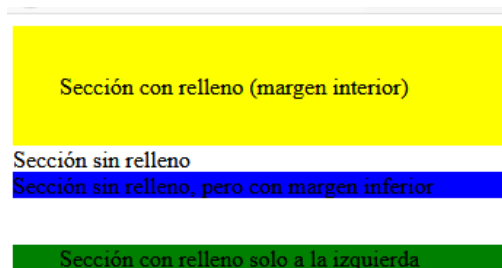
1. Incorrecto
2. Opción correcta

3.6.3.- Ejemplos del modelo de cajas.

Ejemplo de relleno y margen

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Relleno y margen</title>
6     <style>
7       #relleno{
8         background-color:yellow;
9         padding: 2em; /*se aplica a izquierda, derecha arriba y abajo*/
10      }
11      #rellenoIzq{
12        background-color:green;
13        padding-left: 2em; /*se aplica a izquierda, derecha arriba y abajo*/
14      }
15      #rellenoMargen{
16        background-color:blue;
17        margin-bottom: 2em; /*se aplica a izquierda, derecha arriba y abajo*/
18      }
19    </style>
20  </head>
21  <body>
22    <section id = "relleno">Sección con relleno (margen interior)</section>
23    <section>Sección sin relleno</section>
24    <section id = "rellenoMargen">Sección sin relleno, pero con margen inferior</section>
25    <section id = "rellenoIzq">Sección con relleno solo a la izquierda</section>
26  </body>
27 </html>
```

Se verá así:



Materiales educativos de la CAM (Uso educativo NC)

El espacio en blanco (color de fondo de **<body>**) entre las secciones y azul y verde se debe al margen inferior de la primera.

Ejemplo de **border**

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>Relleno y margen</title>
6     <style>
7       #borde1{
8         background-color:yellow;
9         border-color: green;
10        border-style: solid;
11      }
12      #borde2{
13        background-color:pink;
14        border-width: 16px;
15        border-left-color: green;
16        border-top-color: green;
17        border-left-style: dashed;
18        border-top-style: dashed;
19      }
```

```
20 |         </style>
21 |     </head>
22 |     <body>
23 |         <section id = "borde1">Sección con borde1</section>
24 |         <br>
25 |         <section id = "borde2">Sección con borde2</section>
26 |     </body>
27 | </html>
```

Se verá así:



Materiales educativos de la CAM (Uso educativo NC)

3.6.4.- Posicionamiento.

En los ejemplos vistos hasta ahora, el navegador representa los elementos en el orden en el que aparecen atendiendo a si son elementos de bloque o de línea (entre otras cosas). Utilizando CSS, es posible modificar el posicionamiento por defecto de los elementos. Las propiedades implicadas son **position** y **float**.

Con la propiedad **float**, los elementos "flotan" hacia la izquierda o derecha. Todos los elementos flotados se van situando uno junto a otro. Si no hay espacio disponible, se pasan a una nueva línea. Se adapta bastante bien a diferentes tamaños de pantalla.

En este ejemplo se utiliza la propiedad **float** para maquetar una página sencilla, junto a las etiquetas semánticas. Los elementos **<nav>** y **<section>** están flotados a la izquierda y se reparten la anchura disponible.

Es necesario fijar la altura porque en la página apenas hay contenido.

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="UTF-8">
5      <title>Etiquetas semánticas y float</title>
6      <style>
7        body{
8          background-color: pink;
9        }
10       header{ background-color: blue;}
11       nav {
12
13         background-color: red;
14         width: 10%;
15       }
16       section {
17         background-color: green;
18         width: 90%;
19       }
20       footer {
21         background-color: yellow;
22       }
23       nav, section {
24         height: 500px;
25         float: left;
26       }
27     </style>
28   </head>
29   <body>
30     <header>Encabezado</header>
31     <nav>Vínculos</nav>
32     <section>Contenido principal del página</section>
33     <footer>Página creada por...</footer>
34   </body>
35 </html>
```

Se vería así:



Para saber más

- ✓ Los métodos de posicionamiento se describen en detalle en este [vínculo](#) (en el capítulo 5).
- ✓ [Aquí](#) puedes ver más ejemplos de maquetación.

3.7.- Validador CSS.

Recomendación

[Validador CSS online](#)

Para saber más

Para saber más

Tutoriales de W3School:

- ✓ [HTML](#)
- ✓ [CSS](#)

Muy prácticos porque se pueden probar online.

Recursos para desarrolladores de Mozilla - [MDN web docs](#). En español.

[Página del W3C](#)