

## Caso práctico

**Juan** y **María** debaten sobre qué lenguaje de marcas usar para realizar la web de la empresa. Por una parte, **Juan** está interesado en que la web sirva para distintas plataformas y que enlace bien con cualquier base de datos y, por otra parte, **María** recuerda que es importante el contenido en formato de video.

¿En qué lenguaje desarrollarán su web? ¿HTML 5 o XHTML?



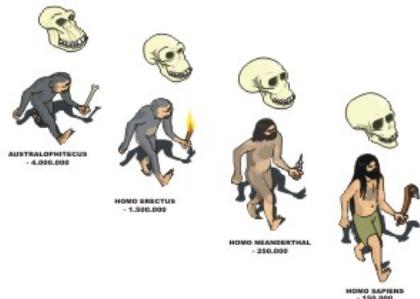
Materiales formativos de FP Online propiedad del Ministerio de Educación, Cultura y Deporte.  
[Aviso Legal](#)

# 1.- HTML y XHTML: evolución.

Los lenguajes de marcas más usados en los sitios web son el HTML y el XHTML, que son lenguajes de presentación. La misión de cualquiera de los dos es presentar información legible para las personas.

Ahora bien, los encargados de acordar las reglas de sintaxis que las rigen, deben tener en cuenta que serán usadas por navegadores y servidores web.

Por eso, las normas de estos lenguajes son definidas y publicadas por el **Consorcio W3C**, que las elabora a partir de la colaboración de diferentes desarrolladores que forman grupos de trabajo. Así se garantiza que sean públicas y estén disponibles para todos los que quieran usarlas, garantizando la compatibilidad de los sistemas web.



El origen de HTML fue un sistema de hipertexto para compartir documentos electrónicos en 1980. La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993. Aunque ninguna de las dos propuestas de estándar que se hicieron (HTML y HTML+) consiguieron convertirse en estándar oficial.

- **HTML 2.0** fue la primera versión oficial de HTML el IETE publicó el estándar en septiembre de 1995.
- **HTML 3.2** se publicó el 14 de Enero de 1997 por el W3C. Incorpora 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 presenta se encuentran las hojas de estilos CSS y la posibilidad de incluir pequeños programas en las páginas web.
- **HTML 4.01** es la última especificación oficial de HTML se publicó el 24 de diciembre de 1999. Es una 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 5.0:** A pesar de que ya se estaba usando desde mucho antes, la norma no fue publicada por el Consorcio W3C hasta el 28 de octubre de 2014.
- **HTML 5.1 2nd Edition**, publicada el 3 de octubre de 2017.
- **HTML 5.2** en vigor desde el 14 de diciembre de 2017.
- **HTML 5.3:** el grupo de trabajo del W3C está desarrollando esta actualización desde el 18 de octubre. En cuanto sea publicada podrá ser usada por todos.

El XHTML surge como una necesidad de poder usar las herramientas de XML en entornos HTML. Durante varios años XHTML fue el lenguaje de marcas más usado ya que HTML 4.01 se quedó obsoleto y HTML5 no era una prioridad para el consorcio W3C, que estaba inmerso en la explotación de las posibilidades de XML.

- **XHTML 1.0** fue la primera versión, se publicó el 26 de Enero de 2000. Es una adaptación de HTML 4.01 al lenguaje XML, por lo que mantiene sus características, y añade algunas restricciones y elementos de XML.
- La versión **XHTML 1.1** publicada el 23 de noviembre de 2010 se encargaba de modularizar XHTML. Ha sido reemplazada en marzo del 2018.
- El borrador de **XHTML 2.0** el grupo de trabajo empezó a trabajar esta actualización en 2010 y prometía grandes novedades. Se trataba de un lenguaje versátil que permitía incorporar la tecnología XML para el desarrollo de interfaces para dispositivos móviles. Sin embargo, el grupo de trabajo ha terminado su investigación sin llegar a publicar la norma.

## Autoevaluación

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

- Si  
 No

Vuelve a leer el apartado.

Correcto

## Solución

1. Incorrecto
2. Opción correcta

## Para saber más

En el 2014 se cumplió el 25 aniversario de la web. En el siguiente video podemos escuchar a su creador Tim Berners-Lee hablando de su evolución en "Una carta magna para la web" coincidiendo con dicho aniversario.

[https://embed.ted.com/talks/lang/es/tim\\_berners\\_lee\\_a\\_magna\\_carta\\_for\\_the\\_web](https://embed.ted.com/talks/lang/es/tim_berners_lee_a_magna_carta_for_the_web)

[Enlace en Ted Talks](#)

## 2.- La estructura de un documento HTML 5

### 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:** Sabemos del tema anterior que el prólogo es la declaración del tipo de documento donde se 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.

En HTML5 el prólogo es muy sencillo, debido a que empezó a usarse antes de que se publicase la norma con el vocabulario oficial del lenguaje:

```
<!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. Es la única etiqueta obligatoria para la `<head>`.
- *El cuerpo*, contiene la información que se va a presentar en la pantalla. Está limitado por las etiquetas `<body>` y `</body>`.

Luego el esqueleto de cualquier página web en html5 es siempre el mismo:

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

### Autoevaluación

¿Cuál de las siguientes afirmaciones es verdadera?

- Un documento html ha de tener `<title>` aunque no tenga `<head>`
- Un documento html puede tener varios `<body>` pero solo una `<head>`
- Un documento `<html>` puede prescindir de `<head>`
- Un documento html ha de tener `<head>` y `<title>`

Fallaste: La `<head>` es un elemento obligatorio

Fallaste: Solo hay un `body` un elemento `head`

Fallaste: `<head>` es un elemento obligatorio

¡Correcto!

## Solución

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

## 2.1.- Etiquetas y atributos en 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.

Las etiquetas permitidas están definidas en el vocabulario que publica el W3C.

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.



### Debes conocer

Las etiquetas válidas para HTML están normalizadas por el W3C, que las actualiza gracias a la colaboración de sus grupos de trabajo.

[Estado actual de las normas para HTML 5.](#)

El vocabulario, es decir, las etiquetas válidas para HTML 5.2 están definidas en la siguiente norma:

[Vocabulario o dtd para HTML 5.2.](#)

## 2.2.- Clasificación de los atributos más comunes.

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

Atributo	Descripción
name = "texto"	Permite asignar el nombre "texto" a un objeto HTML
title = "texto"	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: a, link, img, object, abbr y acronym
id = "texto"	Permite identificar al elemento HTML sobre el que se aplica de forma única mediante el identificador "texto". Sólo es útil cuando se trabaja con CSS y con Javascript. No pueden empezar por números y sólo puede contener letras, números, guiones medios y/o guiones bajos.
style = "texto"	Permite aplicar al elemento HTML el estilo "texto" directamente.
class = "texto"	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 = "codigo"	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 border="1"><thead><tr><th>Código</th><th>Idioma</th><th>Código</th><th>Idioma</th></tr></thead><tbody><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></tbody></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.

## 2.3.- Esqueleto html.

La estructura básica de un documento viene determinada por las siguientes etiquetas:

Elemento	Descripción
html	Documento HTML.
head	Cabecera del documento.
body	Cuerpo del documento. Permite definir formatos que se aplican a los elementos de la página de manera global, como son el color del fondo del texto, los márgenes, el color de los enlaces, ...

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

```
<!DOCTYPE html>
<html>
    <head>
        <title>Ejemplo de la estructura de un documento HTML</title>
    </head>

    <body>
        <!-- Esto es un comentario html -->
        <p>La información que se quiere visualizar en el navegador va en body.</p>
    </body>
</html>
```

Ahora que sabemos como es la estructura de un documento html y tenemos un ejemplo, es hora de empezar a practicar.

Las únicas herramientas que se necesitan son un editor de texto plano y un navegador web.

### Editores:

- Notepad ++: <https://notepad-plus-plus.org/download/v7.5.9.html>
- Sublime Text: <https://www.sublimetext.com>

**Navegadores:** La recomendación es probar nuestras páginas html con más de un navegador. Al menos, deberíamos probar en Firefox, Chrome e Internet Explorer.

## 3.- Elementos HTML5.

Un elemento HTML está formado por:

- ✓ Una etiqueta de apertura.
- ✓ Cero o más atributos.
- ✓ Texto encerrado por la etiqueta. Es opcional, no todas las etiquetas pueden encerrar texto.
- ✓ Una etiqueta de cierre.

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.

Existen elementos cuyo comportamiento puede ser en línea o de bloque según las circunstancias.

El siguiente ejemplo muestra la diferencia entre ambos comportamientos:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Diferencia entre los elementos en línea y los elementos de bloque</title>
  </head>

  <body>
    <h1>Los encabezados son elementos de bloque.</h1>
    <p>Y los párrafos también.</p>
    <a href="https://www.w3c.es/">Los enlaces son elementos de línea</a>
    <p>Incluso si esta definido dentro de un párrafo, <b>un texto en negrita</b>
    sigue siendo un elemento en línea. </p>
  </body>
</html>
```

[Código fuente de la imagen](#)

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



### 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.

Lo siento, prueba otra vez.

Lo siento, prueba otra vez.

Correcto, los de bloque ocupan siempre hasta el final de la pantalla.

Lo siento, prueba otra vez.

## Solución

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

## 3.1.- Elementos de la cabecera.

### ✓ Metadatos del documento:

Elemento	Descripción	
title	Título del documento. Solo puede contener texto y cualquier otra información no se interpreta	
base	Define la URL base para las URLs relativas en la página	
link	Sirve para enlazar con documentos externos, por ejemplo, CSS o JavaScript	
meta	Sirve para definir los metadatos, es decir, indentificar propiedades como autor, fecha, codificación, etc.	
	<meta charset="UTF-8">	Informa sobre la codificación del documento.
	<meta name="description" content="Descripción">	Describe el contenido de la web
	<meta name="title" content="Título de la WEB">	El título de la web para los buscadores.

### ✓ Elementos de script:

Elemento	Descripción	
script	Para definir un script a un lenguaje externo, por ejemplo, JavaScript	
noscript	Contenido alternativo si el navegador no soporta scripts	

## Recomendación

La web no entiende de fronteras, por eso es importante emplear codificaciones que no den problemas con los caracteres de idioma. Por ejemplo, en el idioma Español utilizamos caracteres propios como la ñ. Luego, lo más sencillo sería emplear la codificación **iso8951-1** que contempla dicho carácter. Sin embargo, puede tener problemas si los navegadores o servidores no soportan dicha codificación.

La solución es sencilla, independizarnos de los caracteres de idioma y guardar nuestros documentos con codificación internacional **utf-8**.

## 3.2.- Elementos contenedores.

Los elementos contenedores se usan para agrupar etiquetas con el mismo contenido semántico, o simplemente establecer secciones a las que aplicar estilos diferenciados.

Elemento	Descripción
<div>	Se usa para definir bloques o agruparlos. Ya existía en html 4
<article>	El contenido dentro de <article> tiene un valor semántico alto, y debe aportar información relevante de nuestro documento. Se usa para representar un contenido específico en nuestra web. Si sacamos el contenido de <article> de la web, por sí solo ya tendría entidad propia. Puede representar un artículo, un comentario, entrada de blog, etc.
<section>	Define la sección de una página o documento. Se usa para representar un bloque con valor semántico relacionado Semánticamente es menos importante que <article> pero más importante que <div>. Puedo tener un artículo con diferentes secciones. Puedo tener una sección de deportes con varios artículos.
<header>	Se usa para definir el encabezado de un texto o de una sección.
<footer>	Se usa para definir el pie de un texto o de una sección.
<nav>	Define una sección que solamente contiene enlaces de navegación
<aside>	Define algunos contenidos vagamente relacionados con el resto del contenido de la página. Si es removido, el contenido restante seguirá teniendo sentido
<main>	Define el contenido principal o importante en el documento. Solamente existe uno en el documento
<address>	Contiene información de contacto.
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>	Los elementos de cabecera implementan seis niveles de cabeceras de documentos; <h1> es la de mayor y <h6> es la de menor importancia. Un elemento de cabecera describe brevemente el tema de la sección que introduce.
<form>	Elemento que define un formulario y contiene todos los elementos con los controles deseados.
<iframe>	Representa un contexto de navegación anidada, el cual permite incorporar otra página HTML en la página actual

### Reflexiona

¿Cómo escribiríamos un artículo en html5 empleando etiquetas con valor semántico?

El artículo tiene una cabecera con el título, el cuerpo y el pie con el autor.

[Mostrar retroalimentación](#)

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Artículo</title>
  </head>
  <body>
    <article>
      <header>
        <h1>Noticia</h1>
      </header>
      <section>
        <h2>Opinan los lectores</h2>
      </section>
      <footer>
        <h3>Escrito por DDAW</h3>
      </footer>
    </article>
  </body>
</html>
```

[Descarga el código fuente del artículo](#)

## Reflexiona

Y, ¿cómo crearíamos una sección de opinión con diversos artículos?

[Mostrar retroalimentación](#)

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Artículo</title>
  </head>
  <body>
    <section>
      <h1>Hablando de HTML 5</h1>
      <article>
        <header><h2>Tituto articulo 1</h2></header>
        <nav>menú de navegación</nav>
        <p>EL cuerpo del articulo</p>
        <footer><h3>DDAW</h3></footer>
      </article>
    </section>
  </body>
</html>
```

[Descarga el código](#)

### 3.3.- Elementos para formatear texto.

Los distintos elementos que podemos utilizar para dar formato a nuestro texto son:

Elemento	Descripción
<p>	Delimita los párrafos
<b>	Indica que el texto que está en ese elemento se le pondrá en negrita.
<i>	Indica que el texto que está en ese elemento se le pondrá en itálica ó cursiva.
<u>	Indica que el texto que está en ese elemento se le pondrá subrayado.
<sup>	Indica que el texto que está en ese elemento es un supraíndice.
<sub>	Indica que el texto que está en ese elemento es un subíndice.
<strong>	Indica que el texto que está en ese elemento estará resaltado. Habitualmente los navegadores resaltan el texto poniéndolo en negrita aunque podría haber algún navegador que resaltase el texto poniéndolo en cursiva y en naranja.
<cite>	Inserta una cita o referencia. Se suele usar para hacer referencia a los autores de alguna cita.
<code>	Para insertar un fragmento de código de una aplicación.

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

```
<!DOCTYPE html>
<html>
  <head>
    <title>Formatear textos</title>
  </head>
  <body>
    <h1>Este es un encabezado de nivel 1.</h1>
    <h6>Pero este otro es de nivel 6</h6>
    <p>Ahora voy a definir un párrafo. Dentro del que pondremos texto en <b>negrilla</b>,
      <i>cursiva</i>, <u>subrayada</u> e incluso <b><u>negrilla y subrayada</u></b> simultáneamente y
      <b><i><u>negrilla, cursiva y subrayada.</u></i></b> mediante el anidamiento de etiquetas.
      Además vamos a mostrar el uso de <sub>subíndices</sub> y de <sup>supraíndices</sup>,
      así como <strong>texto resaltado, que no es negrita</strong>
    </p>
  </body>
</html>
```

[Código fuente de la imagen](#)



### Autoevaluación

Si queremos poner un texto en negrita, podemos usar la etiqueta <b>

- Verdadero  Falso

**Verdadero**

## 3.4.- Elementos de lista.

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
<li>	Indica cada uno de los elementos de una lista
<dl>	Delimita los elementos que forman una lista de definición
<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:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Listas</title>
  </head>
  <body>
    <h3>Lista desordenada: Módulos Profesionales:</h3>
    <ul>
      <li>Bases de datos</li>
      <li>Programación</li>
    </ul>
    <h3>Ejemplo de lista ordenada: Módulos Profesionales:</h3>
    <ol>
      <li>Bases de datos</li>
      <li>Programación</li>
    </ol>

    <h3>Ejemplo de lista de definición: Modulos profesionales:</h3>
    <dl>
      <dt>Bases de datos</dt>
      <dd>Se estudian las bases de datos relacionales.</dd>
      <dt>Programación</dt>
      <dd>Se aprende a programar en Java</dd>
    </dl>
  </body>
</html>
```

[Código fuente de la imagen](#)

The screenshot shows a web browser window displaying the rendered HTML from the previous code block. The browser interface includes standard navigation buttons (back, forward, home), a search bar, and a toolbar. The main content area shows three sections: 1) A list titled "Lista desordenada: Módulos Profesionales:" containing two items: "Bases de datos" and "Programación". 2) A list titled "Ejemplo de lista ordenada: Módulos Profesionales:" containing two items: "1. Bases de datos" and "2. Programación". 3) A definition list titled "Ejemplo de lista de definición: Modulos profesionales:" with two entries: "Bases de datos" followed by the definition "Se estudian las bases de datos relacionales.", and "Programación" followed by the definition "Se aprende a programar en Java".

## 3.5.- Elementos de tabla.

---

Los elementos para definir una tabla son los siguientes:

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.

Un ejemplo de un documento HTML que muestra el modo de utilizar algunos de estos elementos es:

```
<!DOCTYPE html>
<html>
    <head>
        <title>Tablas</title>
    </head>
    <body>
        <h3>Tabla: Horas semanales para LMSGI</h3>
        <table border = "1">
            <thead>
                <tr> <th>Dia</th> <th>Horas</th> </tr>
            </thead>
            <tfoot align="center"> <tr> <td>Total</td> <td>4</td> </tr> </tfoot>
            <tbody align="center">
                <tr> <td>Miércoles</td> <td>2</td> </tr>
                <tr> <td>Jueves</td> <td>2</td> </tr>
            </tbody>
        </table>
    </body>
</html>
```

[Código fuente de la imagen](#)

← → ⌂ ⓘ Archivo | file:///D:/Curso\_18.

**Tabla: Horas semanales para LMSGI**

Dia	Horas
Miércoles	2
Jueves	2
Total	4

## 3.6.- Elementos multimedia.

HTML5 muestra todo su poder con el formateo de los recursos multimedia: imágenes, gráficos, video, sonido...

Elemento	Descripción
<img>	<b>Imágenes</b> Permite insertar una imagen en una página web. Es obligatorio utilizar el atributo src para determinar el path del fichero de imagen que queremos insertar.
<audio>	<b>Sonido</b> Contenido de tipo sonido
<embed>	<b>Multimedia</b> Se utiliza para insertar elementos multimedia de audio y video.
<figure>	<b>Figuras.</b> Una imagen, una ilustración, un diagrama, un fragmento de código, o un esquema al que se hace referencia en el texto principal, pero que se puede mover a otra página o a un apéndice sin que afecte al flujo principal.
<map>	<b>Imagen – mapa.</b> Se usa para especificar o crear un mapa. Por ejemplo, para enlazar un fragmento de una imagen con otras.
<video>	<b>Video.</b> Para definir un video o película.

## Ejercicio Resuelto

Veamos como usar algunas etiquetas multimedia

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="UTF-8"/>
    <meta name="description" content="Ejemplo etiquetas HTML5"/>
    <title>Ejemplo HTML5</title>
  </head>
  <body>
    <article>
      <header>Multimedia html5.</header>
      <h1>Retales multimedia</h1>
      
      <section id="multimedia">
        <p>Etiqueta &lt;audio&gt; de HTML5</p>
        <audio src="https://radioteca.net/media/uploads/audios/25Y_25m/a20kind20of20magic20-20queen(20top2020).mp3" controls>
        </audio>
        <div id="videos">
          <h2>Sobre los videos</h2>
          <p>Ejemplo: Insertado con iframe</p>
          <iframe width="560" height="315" src="https://www.youtube.com/embed/yHv16o56AlM" allow="accelerometer; autoplay; encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
        </div>
      </section>
      <footer><h3>LMSGI - IES AGLinares</h3></footer>
    </article>
  </body>
</html>
```

[Descarga el código](#)

[En el navegador](#)

Multimedia html5.

## Retales multimedia



### Audio

Etiqueta <audio> de HTML5



### Sobre los videos

Ejemplo: Insertado con iframe

```
<video>
  <source src="video.ogg" type="video/ogg"/>
  <source src="video.mp4" type="video/mp4"/>
  <source src="video.webm" type="video/webm"/>
  <p>Tu navegador no soporta HTML5</p>
</video>
```

Desplazar texto entre las etiquetas <video> y </video> sera visualizado en navegadores que no soporten el elemento <video>, es decir, que no implementen HTML5.

LMSGI - IES AGLinares

## 3.7.- Otros elementos.

Otros elementos que podemos utilizar son:

Elemento	Descripción
a	Permite definir un enlace a una página web, un archivo o una dirección de correo.
abbr	Indica una forma abreviada.
blockquote	Contiene un bloque de texto con sangría.
br	Inserta una línea en blanco. No tiene etiqueta de cierre, sólo se abre.

## Ejercicio Resuelto

Ejemplo de uso de estos elementos:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Otros elementos</title>
  </head>
  <body>
    <h1>Otros elementos.</h1>
    <h2>La etiqueta &lt;a&gt;</h2>
    <p>
      <a href="https://es.wikipedia.org/wiki/HTML5">Enlace a wikipedia: HTML5</a>
    </p>
    <h2>Abreviatura:</h2>
    <p>La norma para <abbr title="HyperText Markup Language, versión 5">HTML5</abbr>
    se publicó en octubre de 2014.</p>
    <h2>Texto con sangría</h2>
    <blockquote><p>Este es un texto con sangria.</p></blockquote>
    <h2>Los saltos de linea o <i>break return</i> con &lt;br&gt;</h2>
    <br><br>
    <p>Párrafo tras un par de saltos de linea.</p>
  </body>
</html>
```

[Descarga el código](#)

[En el navegador](#)

## Otros elementos.

### La etiqueta <a>

[Enlace a wikipedia: HTML5](#)

### Abreviatura:

La norma para HTML5 se publicó en octubre de 2014.

### Texto con sangría

Este es un texto con sangría.

### Los saltos de linea o *break return* con <br>

Párrafo tras un par de saltos de línea.

# Autoevaluación

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.

Lo siento, no es la opción más correcta.

Lo siento, no es la opción más correcta.

Lo siento, no es la opción más correcta.

Correcto. Los elementos html pueden contener o no atributos.

## Solución

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

## 4.- Formularios HTML5.

Los formularios permiten interactuar al cliente con la página web: introduciendo texto, seleccionando opciones, confirmando información o enviándosela al servidor. HTML dispone de elementos que permiten todas estas acciones: cuadros de texto, casillas de verificación, botones de acción, etc.

HTML5 incorpora nuevos elementos y atributos a los que ya existían en la versión anterior para conseguir mayor flexibilidad al usuario y dotar de más valor semántico a los formularios.

### Debes conocer

El formulario **<form>** es un elemento contenedor de que incluye elementos para que el usuario pueda escribir datos o marcar opciones. Por lo tanto, llevan asociada una acción y un método.

- La acción sirve para decirle al servidor web lo que queremos hacer con los datos. Por ejemplo:
  - Ejecutar un script: `action="script.php"`.
  - Enviar los datos por correo a una dirección: `action="mailto:direccion@mail.es"`
- El método sirve para escoger la forma en que se envían los datos. Podemos escoger dos métodos diferentes:
  - Get: Si queremos que los datos viajen rápidamente, sin importarnos que aparezcan en la URL: `method="get"`.
  - Post: Si enviamos más información y queremos preservar su intimidad, sin que aparezca en la URL: `method="post"`.
- En HTML5 se ha incorporado un nuevo atributo `novalidate`, que evita que el formulario sea validado antes de su envío.

```
<!DOCTYPE html>
<html lang="es">
    <head>
        <title>Formulario</title>
    </head>
    <body>
        <form action="miescript.php" method="post">
            <!-- Controles de formulario-->
        </form>
        <form action="busca.php" method="get">
            <!-- Controles de formulario-->
        </form>
        <form action="mailto:direccion@mail.com" method="post">
            <!-- Controles de formulario-->
        </form>
    </body>
</html>
```

[Descargar el código](#)

## 4.1.- Elementos de formulario I.

Los elementos clásicos para un formulario son los siguientes:

Elemento	Descripción
form	Delimita el contenido del formulario.
input	Caja de texto para texto corto. Dependiendo del valor que tome el atributo type de este elemento podemos estar ante un texto sin más, un campo de texto donde al escribir no se visualice el contenido si no que escriba asteriscos, un botón de radio que el usuario podrá elegir, una opción que el usuario podrá activar, botones, ...
textarea	Caja de texto para texto largo.
select	Crea un menú desplegable que permite elegir de una lista de opciones que contiene el elemento.
option	Delimita cada una de las opciones de un menú desplegable que le contiene.
button	Permite definir un botón. Su principal ventaja frente a los botones hechos con input es que este elemento permite introducir en el botón cualquier otro elemento de HTML, como por ejemplo imágenes.
fieldset	Permite agrupar elementos de un formulario.
legend	Permite poner un título al fieldset.
label	Etiqueta de un campo del formulario.
datalist	representa la lista de elementos <option> como sugerencias cuando se llena un campo <input>
output	Representa el resultado de un cálculo.

### Autoevaluación

Para crear un botón utilizaremos la etiqueta:

- <button>
- <input>
- Ambas etiquetas son una opción válida.

¿Seguro que es la única opción?

¿Seguro que es la única opción?

Correcto!

### Solución

1. Incorrecto
2. Incorrecto
3. Opción correcta

## 4.2.- Elementos de formularios II.

### Ejercicio Resuelto

Ejemplo de formulario que permita recoger información personal sobre las preferencias de una persona.

The screenshot shows a web browser window with the following elements:

- Header bar: Back, Forward, Stop, Refresh, Archivo, file:///D:/Curso\_18\_19/Material%20Distancia/Distancia%20LMSG%20Actualizar%20r
- Ejemplo de formulario**
- Text input fields for **Nombre:** and **Apellidos:**
- Radio buttons for **Sexo:** M or H
- Text input fields for **Fecha de nacimiento:** dia, mes, año
- Text input field for **Marca tus aficiones:**
- Checkboxes for interests: Los deportes, La lectura, El cine y el teatro, La música
- Text input field for **Qué estás estudiando (ciclo formativo, universidad, idiomas, etc):**
- Buttons: **Enviar** and **Restablecer**

**Pulsa el botón de "enviar" para enviar los datos o el botón de "restablecer" para limpiar el formulario.**

[Mostrar el código](#)

Descarga el código html del formulario:

[Código HTML del ejemplo](#)

### Debes conocer

Amplia tus conocimientos sobre elementos y atributos para los formularios en HTML5 visitando los siguientes enlaces:

Mozilla - Formularios en HTML5:

[https://developer.mozilla.org/es/docs/HTML/HTML5/Forms\\_in\\_HTML5](https://developer.mozilla.org/es/docs/HTML/HTML5/Forms_in_HTML5)

W3Schools - Formularios HTML:

[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)

## 5.- XHTML otra opción frente a HTML.

### 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 es muy similar al lenguaje HTML. De hecho, no es más que una adaptación de HTML al lenguaje XML, el estándar XHTML 1.0 combina la estructuración de XML con la presentación de HTML.

Aunque las etiquetas en XHTML se escriben igual que en HTML, su vocabulario está definido en otro DTD y regulado por otras normas.

El lenguaje HTML tiene una sintaxis muy permisiva, por lo que es posible escribir sus etiquetas y atributos de muchas formas diferentes. Esta 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. Es un lenguaje más rígido y con menos opciones para la multimedia que HTML5.

### Reflexiona

Luego, ¿qué lenguaje usar para construir mi web...HTML5 o XHTML?

[Mostrar retroalimentación](#)

No hay un lenguaje mejor que otro. Me puede ayudar a decidir:

- Si necesito usar multimedia, HTML5 me lo pone más sencillo.
- Si necesito alta compatibilidad con otras tecnologías XML, entonces XHTML es mejor opción.

Actualmente, la mayor parte de las webs se diseñan empleando HTML5.

## 5.1.- XHTML reglas de sintaxis.

El esquema básico del documento, para considerarse conforme a la especificación deberá cumplir las siguientes condiciones:

- ✓ El elemento raíz del documento debe ser <html>.
- ✓ El elemento raíz del documento debe indicar el **espacio de nombres** XHTML usando el atributo xmlns. El espacio nominal para XHTML es **xmlns="http://www.w3.org/1999/xhtml"**

```
<html xmlns="http://www.w3.org/1999/xhtml">
```

- ✓ Debe haber una declaración DOCTYPE en el prólogo del documento. El identificador público incluido en la declaración DOCTYPE debe hacer referencia a alguna de las tres DTD definidas por el W3C usando el Identificador Formal Público correspondiente:

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

### Restricciones básicas que introduce XHTML respecto a HTML en la sintaxis de sus etiquetas:

- ✓ Las etiquetas se tienen que cerrar en orden inverso al que se abren, nunca pueden solaparse.
- ✓ 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.
- ✓ 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 (<br><br/>) se puede utilizar la sintaxis <br/> para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto.

### Otras restricciones:

Además de las cinco restricciones básicas, XHTML incluye otros cambios más avanzados respecto a HTML, entre ellas:

- ✓ Antes de acceder al valor de un atributo, se eliminan todos los espacios en blanco que se encuentran antes y después del valor. Además, se eliminan todos los espacios en blanco sobrantes dentro del valor de un atributo.
- ✓ El código JavaScript debe encerrarse entre unas etiquetas especiales (<![CDATA[ y ]]>) para evitar que el navegador interprete de forma errónea caracteres como & y <.
- ✓ Las páginas XHTML deben prescindir del atributo name en su lugar, siempre debe utilizarse el atributo id.
- ✓ XHTML es necesario separar el formato del contenido. Los párrafos deben separarse consistentemente y las cabeceras h1-h6 sólo deben usarse para destacar los diferentes apartados. Es recomendable dar el formato a los datos por medio del uso de las CSS.

## Debes conocer

El vocabulario de XHTML está definido en un DTD por el W3C. Es importante escoger el DTD adecuado a la versión de XHTML que estemos usando: XHTML 1.0 Strict, XHTML 1.0 Transitional o XHTML 1.1.

Para no equivocarte nunca, escoge el doctype asociado al lenguaje que vayas a usar de la lista oficial del W3C en el siguiente enlace:

[Lista de doctype en el W3C](#)

<https://www.w3.org/QA/2002/04/valid-dtd-list.html>

## Autoevaluación

¿Para qué sirve el espacio de nombres **xmlns**?

### Sugerencia

- Es el espacio de nombres que emplea XHTML para hacer referencia las etiquetas definidas en su DTD
- En realidad carece de utilidad
- Es porque es XML
- Es el espacio de nombres que ponemos siempre, solo que en XHTML es obligatorio y en HTML no.

Muy bien, has acertado.

Fallaste, si fuese cierto, ¿por qué es obligatorio expresarlo?

Fallaste, XHTML es un lenguaje de marcas propio, aunque derive de XML.

Fallaste, revisa el tema.

### Solución

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

## 5.2.- XHTML: ventajas e inconvenientes.

---

### ✓ Ventajas:

- ◆ Compatibilidad parcial con navegadores antiguos: la información se visualiza, aunque sin formato.
- ◆ Un mismo documento puede adoptar diseños radicalmente distintos en diferentes apartados.
- ◆ Sencillez a la hora de editar y mantener el código.
- ◆ Es compatible con los estándares que está desarrollando el W3C como recomendación para futuros agentes de usuario o navegadores.
- ◆ Los documentos escritos conforme a XHTML 1.0 presentan mejor rendimiento en las actuales herramientas web que aquellos escritos conforme a HTML.
- ◆ La separación de los contenidos y su presentación hace que los documentos XHTML se adapten mejor a las diferentes plataformas: pantallas de ordenador, pantallas de dispositivos móviles, ...
- ◆ Como es XML se pueden utilizar fácilmente herramientas creadas para procesar documentos XML genéricos (editores, XSLT, etc.).



### ✓ Inconvenientes:

- ◆ Algunos navegadores antiguos no son totalmente compatibles con los estándares, lo que hace que las páginas no siempre se muestren correctamente. Esto cada vez es menos problemático ya que estos navegadores van cayendo en desuso.
- ◆ Muchas herramientas de diseño web aún no generan código XHTML correcto.

## 6.- CSS: Hojas de estilo.

### Caso práctico

**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.

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

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.
- Se utiliza el lenguaje CSS para definir el formato de cada elemento.

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. 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**, continúa en desarrollo desde 1998. Es la versión que se utiliza actualmente.

#### Sintaxis CSS:

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, }.

**Ejemplo:** p{ color : blue; }

*En este caso el selector es “p”, la declaración es: “color : blue” y, dentro de ésta, podemos diferenciar 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

Las hojas de estilo permiten:

- Definir formatos que se aplican sobre varias páginas web de un sitio.
- Incluir elementos de formato en la estructura de una página web.

- Estructurar el contenido de la página web.
- Todas las opciones anteriores son ciertas.

Correcto.

Incorrecta

Incorrecta

Fallaste

## Solución

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

## 6.1.-Soporte CSS en los navegadores.

El objetivo de una web es que sea compatible con el mayor número posible de navegadores web, lo que significa que muestra un aspecto igual o casi igual en todos.



Sin embargo, este objetivo no es tan sencillo como parece. Podemos darnos por satisfechos si conseguimos que se vea bien para el 99% de los usuarios, lo que implica asegurar la compatibilidad con los navegadores más usados: Chrome, Mozilla, IE, Safari, Opera.

Consejos para asegurar la compatibilidad:

- Validar el código CSS en el validador del W3C para garantizar que cumple con las reglas de sintaxis.
- Validar el código HTML o XHTML en el validador del W3C.
- Usar varios navegadores para probar el código.
- Usar reglas CSS que soporten la mayoría de los navegadores.

Enlace de interés para comprobar la compatibilidad de las etiquetas CSS y los navegadores:

[W3Schools](https://www.w3schools.com/cssref/css3_browsersupport.asp)

[https://www.w3schools.com/cssref/css3\\_browsersupport.asp](https://www.w3schools.com/cssref/css3_browsersupport.asp)

### Debes conocer

Los validadores del W3C: La regla de oro que no debes olvidar: validar tu código CSS y HTML.



- Validador HTML y XHTML: [https://validator.w3.org/#validate\\_by\\_input](https://validator.w3.org/#validate_by_input)
- Validador de CSS: [https://jigsaw.w3.org/css-validator/#validate\\_by\\_input](https://jigsaw.w3.org/css-validator/#validate_by_input)

## 6.2.- Incluir CSS en nuestra web.

Existen tres opciones para incluir CSS en un documento HTML o XHTML:

### 1. Definir CSS en un archivo externo.

En este caso, todos los estilos CSS se incluyen en uno, o varios, archivos de texto plano, cuya extensión es .css, que las páginas HTML enlazan mediante el elemento <link> de la cabecera del fichero HTML.

Puesto que una página web puede tener asociados varios ficheros CSS es recomendable agrupar estos últimos en un directorio.

El navegador descarga los archivos CSS externos, además de la página web asociada a ellos, y aplica los estilos a los contenidos de la página antes de mostrar sus contenidos.

Esta es la forma de incluir CSS en las páginas HTML más utilizada. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Además, el mantenimiento del sitio web se simplifica al máximo, ya que el cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML asociadas.

Puede hacerse de dos modos diferentes:

1. Mediante enlaces.
2. Importando el fichero CSS.

### 2. Incluir CSS en el documento HTML.

Este método se emplea cuando se definen pocos estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos globales de todas las páginas del sitio web.

Tiene el inconveniente de que para modificar los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a cambiar.

### 3. Incluir CSS en los elementos HTML.

El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que para modificar un formato hay que cambiar todos los elementos que estén asociados a él.

Solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

## 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.

Incorrecto

No es la mejor forma, impide la reutilización de estilos

No es la mejor forma, solo me vale el css para un documento.

Correcto.



## **Solución**

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

## 6.3.- CSS externo: link e import.

### Enlazar archivo

Para realizar una página web usando un archivo CSS externo, se deben seguir los tres pasos siguientes:

- Se crea un archivo de texto plano con las definiciones de los formatos.
- Dicho archivo de texto se guarda con extensión .css
- Se enlaza el archivo CSS externo mediante la etiqueta <link> en la cabecera de la página web.

El elemento <link> puede tener definidos cuatro atributos cuando se enlaza un archivo CSS:

- **rel**, indica el tipo de relación que tiene el archivo enlazado y la página HTML. Para los archivos CSS, siempre se utiliza el valor `stylesheet`
- **type**, indica el tipo de recurso enlazado. Para los archivos CSS su valor siempre es `text/css`
- **href**, indica la URL del archivo CSS que contiene los estilos. Puede ser relativa o absoluta y puede referenciar a un recurso interno o externo al sitio web.
- **media**, indica el medio en el que se van a aplicar los estilos del archivo CSS.

### Ejercicio Resuelto

Vamos a conseguir esta apariencia con un sencillo ejemplo.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <title>Ejemplo web</title>
    <link rel="stylesheet" type="text/css" href="R63formato1.css"/>
  </head>
  <body>
    <h1>Título en color azul</h1>
    <p>El párrafo aparece en verde</p>
  </body>
</html>
```

`h1 {color: blue;}`  
`p {color: green;}`

[Descarga el archivo css](#)

[Descarga el código html](#)

La vista en el navegador con el estilo aplicado



The screenshot shows the W3C Markup Validation Service interface. The address bar says "file:///D:/Cu". The main content area displays the HTML code with styling applied. The title "Título en color azul" is in blue, and the paragraph "El párrafo aparece en verde" is in green.

## Título en color azul

El párrafo aparece en verde

### Importar archivo

Se puede obtener el mismo resultado anterior utilizando el elemento <style> en lugar de <link>.

En este caso, se usa una regla de tipo `@import` seguida de una cadena de texto encerrada con comillas simples o dobles que se corresponde con la URL del archivo CSS, o de `url()` conteniendo dicha cadena entre los paréntesis. Las siguientes reglas `@import` son equivalentes para un fichero `formatos.css` que está en el directorio `css`:

```
@import '/css/formatos.css';
@import url('/css/ formatos.css');
```

### Reflexiona

Prueba a modificar el ejemplo anterior, sustituyendo el archivo enlazado por el importado. Para ello, debes volver a escribir la cabecera del documento.

```
<head>
  <title>Ejemplo web</title>
  <style type="text/css">
    @import 'R63formato1.css';
  </style>
</head>
```

[Mostrar retroalimentación](#)

¿Lo has conseguido? Comprueba el resultado

[Descarga el documento html con el estilo importado.](#)

## 6.4.- CSS dentro del documento HTML.

### Incluir CSS en el documento HTML:

En este caso los formatos de los elementos se definen en la cabecera del documento HTML, dentro del elemento `<style>`. Por ejemplo:



```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>CSS en la cabecera del documento</title>
    <style type="text/css">
        h3 { color: green; font-family: Times; }
        p { color: orange; font-family: Verdana; }
    </style>
</head>
<body>
    <h3>Ejemplo de CSS en la cabecera del documento</h3>
    <p>En esta ocasión el párrafo de texto aparece de color naranja y el encabezado h3 aparece en verde, mientras que en los ejemplos vistos hasta ahora el color de la letra era negro.</p>
</body>
</html>
```

[Descargar el código](#)

### Incluir CSS en las etiquetas:

Es una práctica poco recomendada que debería limitarse al máximo. De hecho, la mayoría de las veces nos dará error en el validador.

Para dar estilo diferenciado a una etiqueta concreta, siempre podemos recurrir a las clase y los selectores.

El ejemplo anterior, con el CSS en las etiquetas sería:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <title>CSS en la cabecera del documento</title>
</head>
<body>
    <h3 style="color: green; font-family: Times;">
        Ejemplo de CSS en la cabecera del documento</h3>
    <p style="color: orange; font-family: Verdana;">
        En esta ocasión el párrafo de texto aparece de color naranja y el encabezado h3 aparece en verde, mientras que en los ejemplos vistos hasta ahora el color de la letra era negro.</p>
</body>
</html>
```

[Descargar el código](#)

## 6.5.- Atributos de color y fondo.

Los atributos 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: repeat-x, repeat-y o no-repeat.
background-attachment	Especifica si la imagen ha de permanecer fija o realizar un scroll. Los valores que pueden tomar son: scroll o fixed.
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 [top, center, bottom] [left, center, right]
background	Establece en un solo paso cualquiera de las propiedades de background anteriores. Los valores que puede tomar son: background-color, background-image, background-repeat, background-attachment, background-position.

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:

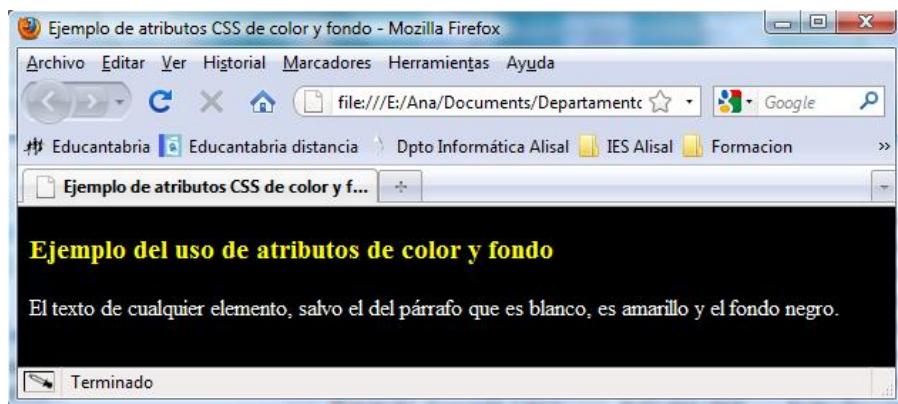
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de color y fondo</title>
        <style type="text/css">
            body { background-color: black; color:yellow; }
            p { color: #fffff; }
        </style>
    </head>

    <body>
        <h3>Ejemplo del uso de atributos de color y fondo</h3>
        <p>El texto de cualquier elemento, salvo el del p<aacute;>rrafo que es blanco, es amarillo y el fondo negro.</p>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



## 6.6.- Atributos de fuente.

En este apartado vamos a ver los distintos atributos que podemos utilizar referentes a las fuentes de nuestro documento y que son:

Elemento	Descripción
font-size	Indica el tamaño de la fuente. Puede ser un tamaño absoluto, relativo o en porcentaje. Toma valores de unidades de CSS
font-family	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.
font-weight	Define el grosor de los caracteres. Los valores que puede tomar son: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800 o 900
font-style	Determina si la fuente es normal o cursiva. El estilo oblique es similar al cursiva. Los valores posibles son: normal, italic, oblique.
font-variant	Determina si la fuente es normal o mayúsculas pequeñas. Los valores que puede tomar son: normal o small-caps
line-height	El alto de una línea y por tanto, el espacio entre líneas. Es una de esas características que no se podían modificar utilizando HTML.
font	Permite establecer todas las propiedades anteriores en el orden que se indica a continuación: font-style, font-variant, font-weight, font-size[line-height], font family. 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:

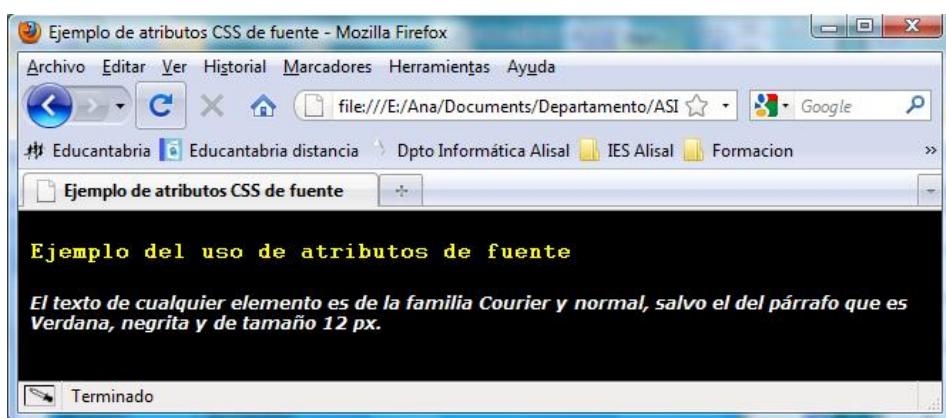
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de fuente</title>
        <style type="text/css">
            body { background-color: black; color:yellow; font-family: courier }
            p { color: #ffffff; font:italic 900 12px Verdana; }
        </style>
    </head>

    <body>
        <h3>Ejemplo del uso de atributos de fuente</h3>
        <p>El texto de cualquier elemento es de la familia Courier y normal, salvo el del párrafo que es Verdana, negrita y de tamaño 12 px.</p>
    </body>
</html>
```

### Código fuente de la imagen

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



## 6.7.- Atributos 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: none, underline, overline, line-through o blink
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: left, right, center o justify
text-indent	Determina la tabulación del texto. Los valores que toma son una longitud, en unidades CSS, 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: capitalize, uppercase, lowercase o none
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: baseline, sub, super, top, text-top, middle, bottom, text-bottom 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:

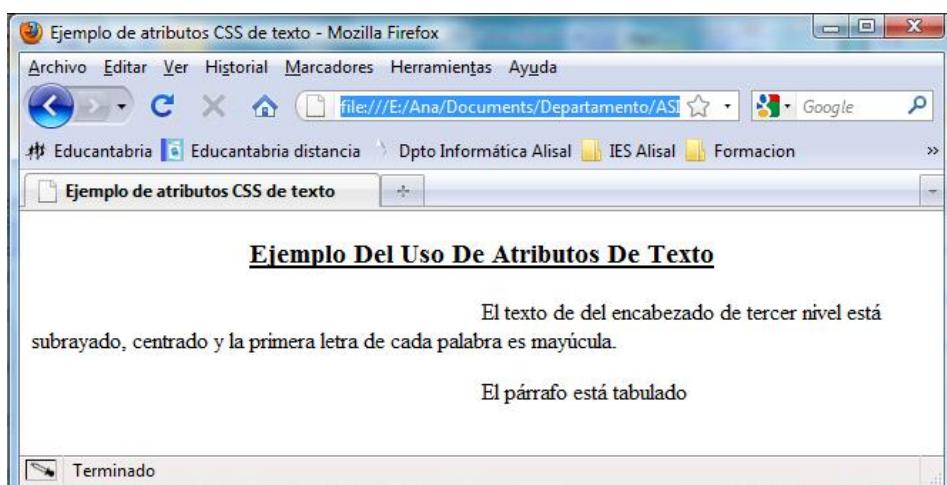
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de texto</title>
        <style type="text/css">
            h3 { text-decoration:underline; text-align: center; text-transform: capitalize }
            p { text-indent: 50%; }
        </style>
    </head>

    <body>
        <h3>Ejemplo del uso de atributos de texto</h3>
        <p>El texto de del encabezado de tercer nivel está subrayado, centrado y la primera letra de cada palabra es mayúcula.</p>
        <p>El párrafo está tabulado</p>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



## 6.8.- Atributos de caja.

Ahora vamos a ver otros atributos muy importantes y que utilizaremos muy a menudo y que no son ni más ni menos que los atributos de caja:

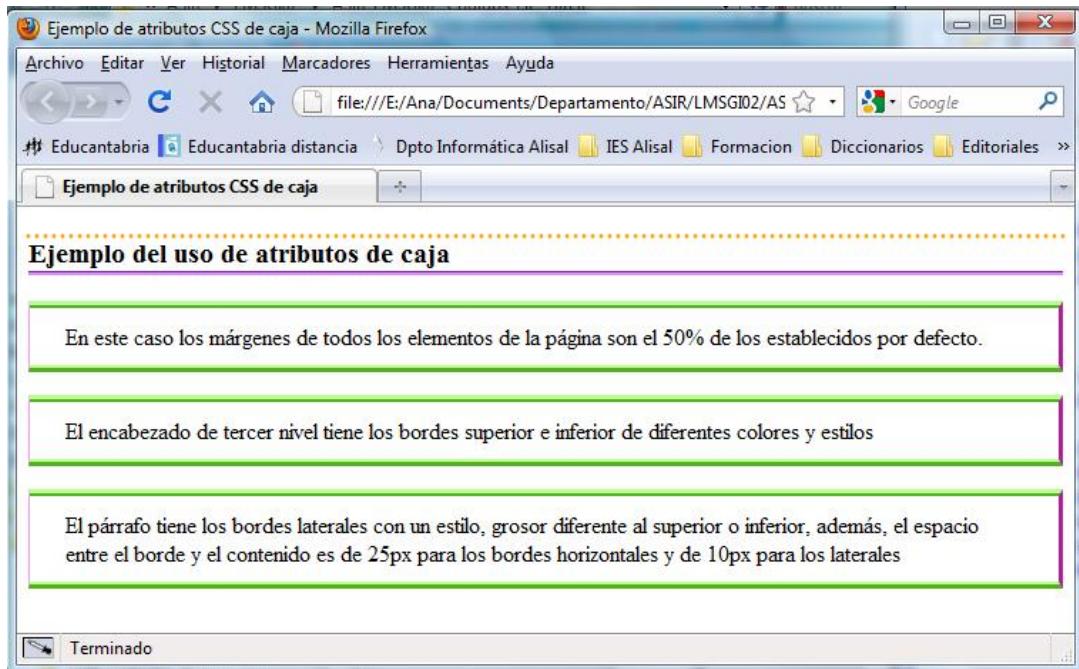
Elemento	Descripción
margin-left	Indica el tamaño del margen izquierdo. Puede usarse una longitud, en unidades CSS, o un porcentaje.
margin-right	Indica el tamaño del margen derecho. Puede usarse una longitud, en unidades CSS, o un porcentaje.
margin-top	Indica el tamaño del margen superior. Puede usarse una longitud, en unidades CSS, o un porcentaje.
margin-bottom	Indica el tamaño del margen inferior. Puede usarse una longitud, en unidades CSS, o un porcentaje.
margin	Permite establecer los márgenes de una vez. Hay que seguir el orden: superior, derecho, inferior e izquierdo.
padding-left	Indica el espacio izquierdo entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
padding-right	Indica el espacio derecho entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
padding-top	Indica el espacio superior entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
padding-bottom	Indica el espacio inferior entre el borde y el contenido. Puede usarse una longitud, en unidades CSS, o un porcentaje.
padding	Establece el espacio entre los bordes y el contenido de una sola vez. Hay que respetar el orden superior, derecho, inferior e izquierdo.
border-left-color	Establece el color del borde izquierdo del elemento. Su valor es un color RGB o el nombre del color.
border-right-color	Establece el color del borde derecho del elemento. Su valor es un color RGB o el nombre del color.
border-top-color	Establece el color del borde superior del elemento. Su valor es un color RGB o el nombre del color.
border-bottom-color	Establece el color del borde inferior del elemento. Su valor es un color RGB o el nombre del color.
border-color	Establece el color de los bordes del elemento de una sola vez. Hay que seguir el orden superior, derecho, inferior e izquierdo. Su valor es un color RGB o el nombre del color.
border-style	Establece el estilo del borde, los valores significan: none=ningun borde, dotted=punteado (no funciona siempre), solid=solido, double=doble borde, los valores groove, ridge, inset y outset son bordes con varios efectos 3D.
border-left-width	Grosor del borde izquierdo. Sus valores posibles son: thin, medium, thick o un tamaño.
border-right-width	Grosor del borde derecho. Sus valores posibles son: thin, medium, thick o un tamaño.
border-top-width	Grosor del borde superior. Sus valores posibles son: thin, medium, thick o un tamaño.
border-bottom-width	Grosor del borde inferior. Sus valores posibles son: thin, medium, thick o un tamaño.
border-width	Establece el tamaño de los bordes del elemento al que lo aplicamos. Hay que seguir el orden superior, derecho, inferior, izquierdo.
width	Establece el ancho del contenido del elemento. El valor es un porcentaje o un tamaño.

height	Establece la altura del contenido del elemento. El valor es un porcentaje o un tamaño.
float	Sirve para alinear un elemento a la izquierda o la derecha haciendo que el texto se agrupe alrededor de dicho elemento. Toma los valores none, left o right
clear	Establece si un elemento tiene a su altura imágenes u otros elementos alineados a la derecha o la izquierda. Sus valores posibles son: none, left, right o both.

Para que practiques todo lo aprendido te recomiendo que intentes hacer el ejercicio que se propone a continuación, antes de ver su solución. ¡Ánimo!

## Ejercicio Resuelto

¿Cuál sería un posible código XHTML para el documento de la imagen que se muestra a continuación?



[Mostrar retroalimentación](#)

### Descarga de código

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de caja</title>
        <style type="text/css">
            body{ margin: 50%, 50%, 50%, 50%; }
            h3 { border-top-color: orange; border-bottom-color: #b428ff; border-top-style: dotted; border-bottom-style: groove; }
                p { border-color: #73ff28 #ff28df #73ff28 #ff28df; border-style: ridge outset ridge outset; border-width: thick medium thick thin; padding: 10px 25px 10px 25px; }
        </style>
    </head>

    <body>
        <h3>Ejemplo del uso de atributos de caja</h3>
        <p>En este caso los márgenes de todos los elementos de la página son el 50% de los establecidos por defecto.</p>
        <p>El encabezado de tercer nivel tiene los bordes superior e inferior de diferentes colores y estilos </p>
        <p>El párrafo tiene los bordes laterales con un estilo, grosor diferente al superior o inferior, además, el espacio entre el borde y el contenido es de 25px para los bordes horizontales y de 10px para los laterales </p>
    </body>
</html>
```

## 6.9.- Atributos de clasificación.

En este apartado vamos a ver otros atributos que hemos etiquetado como atributos de clasificación y que son los siguientes:

Elemento	Descripción
display	Determina si el elemento es de bloque, línea, lista o ninguno de ellos. Los valores que puede tomar son: block, inline, list-item o none.
white-space	Indica el modo en que se ha de gestionar los espacios en blanco que hay en el elemento, es decir, si se mantienen todos los existentes tal y como estén en el documento o si se anulan a uno las secuencias de blancos, es el valor por defecto y el de la opción normal. Valores que puede tomar son: pre, nowrap, normal.
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.

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

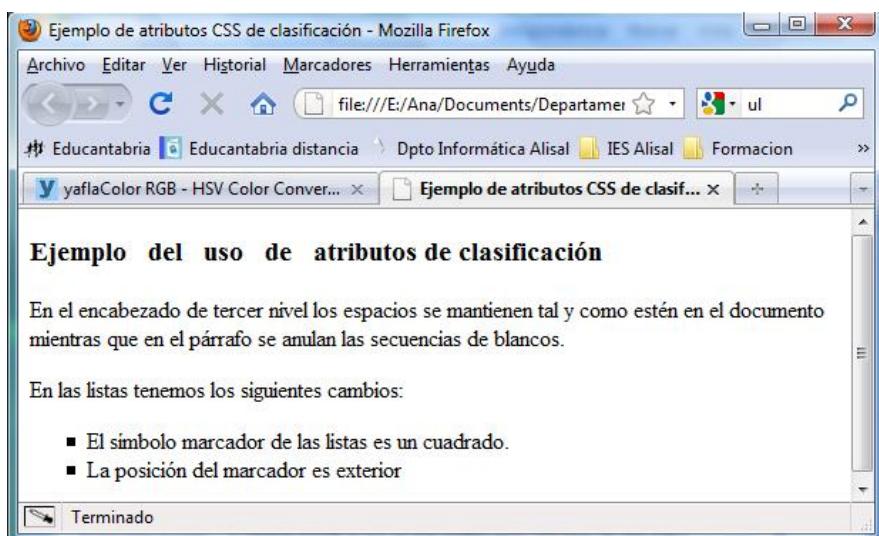
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de clasificaci&onacute;n</title>
        <style type="text/css">
            h3 { white-space:pre; }
            p { white-space:normal; }
            ul { list-style:square; list-style-position: outside; }
        </style>
    </head>

    <body>
        <h3>Ejemplo del uso de atributos de clasificaci&onacute;n</h3>
        <p>En el encabezado de tercer nivel los espacios se mantienen tal y como est&eacute;n en el documento mientras que en el p&aacute;rrafo se anulan las secuencias de blancos.</p>
        <p>En las listas tenemos los siguientes cambios:</p>
        <ul>
            <li>El s&iacute;mbolo marcador de las listas es un cuadrado.</li>
            <li>La posici&onacute;n del marcador es exterior</li>
        </ul>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



## 6.10.- Posicionamiento CSS.

Es un añadido a CSS que permite determinar el modo en que se ha de colocar un determinado elemento. Las propiedades definidas en CSS-P son las siguientes:

Elemento	Descripción
clip	Permite seleccionar una zona. Los valores que puede tomar son: shape o auto.
height	Permite establecer la altura de un elemento. Los valores que puede tomar son: auto o un tamaño.
width	Permite establecer la anchura de un elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
visibility	Indica si el elemento sobre el que actúa será visible o no. Los valores que puede tomar son:
left	Indica la posición del lado izquierdo del elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
top	Indica la posición del lado superior del elemento. Los valores que puede tomar son: auto o un tamaño o porcentaje.
overflow	Indica si el elemento será visible o no en caso de superar los límites del contenedor. Los valores que pueden tomar son: visible, hidden, scroll o auto.
position	Determinan si el posicionamiento de un elemento es absoluto, relativo o estático. Los valores que puede tomar son: absolute, relative o static.
z-index	Define la posición del elemento en el tercer eje de coordenadas, permitiendo superponer unos elementos sobre otros como si fueran capas.

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

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de posicionamiento</title>
        <style type="text/css">
            img{ position:absolute; left: 10px; top: 0px; z-index:-1; }
            p{overflow:scroll; }
        </style>
    </head>

    <body>
        <h3>Ejemplo del uso de atributos de posicionamiento</h3>
        </img>
        <p> La imagen tiene un valor del atributo z-index menor que el del encabezado de tercer nivel y que el del p&acute;srafo. Este es el motivo por el que la imagen aparece por detrás del texto. <br><br/>
            Si le texto del p&acute;srafo no cabe en el espacio que tiene asignado se ha habilitado un scroll.</p>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



## 6.11.- Clases y selectores CSS.

Cuando las reglas de estilos se asocian a un documento HTML utilizando un fichero externo o incluyéndolas en el contenido de la etiqueta STYLE en la cabecera del documento, pueden definirse estilos y asociarlos a determinados elementos del documento.

Para definir una clase hay que usar la sintaxis siguiente:

.clase\_azul{color:blue}

Para asociar un elemento HTML a una clase habrá que usar el atributo CLASS al usar dicho elemento en el documento HTML del siguiente modo:

<h3 class="clase\_azul">El encabezado de tercer nivel es ahora azul</h3>

Para restringir la clase a un determinado elemento basta poner el elemento delante del punto al definir la regla. Por ejemplo, para restringir el uso de la clase a párrafos tendremos:

p.clase\_azul{color:blue}

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

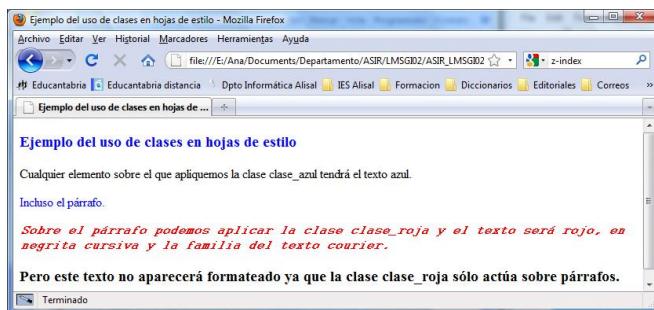
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo del uso de clases en hojas de estilo</title>
        <style type="text/css"> .clase_azul{color:blue} p.clase_roja{color:#ff0000; font-style:italic; font-weight:bolder; font-family:courier;} </style>
    </head>

    <body>
        <h3 class="clase_azul">Ejemplo del uso de clases en hojas de estilo</h3>
        <p>Cualquier elemento sobre el que apliquemos la clase clase_azul tendrá el texto azul.</p>
        <p class="clase_azul"> Incluso el párrafo.</p>
        <p class="clase_roja">Sobre el párrafo podemos aplicar la clase clase_roja y el texto será rojo, en negrita cursiva y la familia del texto courier.</p>
        <h3 class="clase_roja"> Pero este texto no aparecerá formateado ya que la clase clase_roja sólo actúa sobre párrafos.</h3>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



### Recomendación

El W3Schools nos resume como usar los selectores en el siguiente enlace: [W3Schools selectores](#)

En Libros Web también podemos encontrar un libro útil para trabajar con CSS: [Libros web: Introducción a CSS. Selectores](#).

Mozilla también nos enseña como usarlos: [Mozilla - selectores](#).

## 6.12.- Unidades de medida.

Las distintas unidades que podemos utilizar para indicar tamaños son las siguientes:



### ✓ Relativas

- ◆ *Element (em)*: Expresa el tamaño relativo al tamaño de la fuente utilizada.
- ◆ *X-height (ex)*: Expresa el tamaño relativo al de la letra "x".
- ◆ *Pixel (px)*: Expresa el tamaño relativo a la resolución del monitor.

### ✓ Absolutas

- ◆ *Milímetros (mm)*.
- ◆ *Centímetros (cm)*: Cada centímetro son 10 mm.
- ◆ *Pulgadas (in)*: Cada pulgada equivale a 2,54 cm.
- ◆ *Puntos (pt)*: Cada punto son 1/72 in.
- ◆ *Picas (pc)*: Cada pica son 12 pt.

### Autoevaluación

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

Verdadero  Falso

**Falso**  
Correcto