

LMSGI

UNIDAD 2

Utilización de lenguajes de marcas en entornos web.

Javier Díaz Garrido

INDICE

Utilización de lenguajes de marcas en entornos web.

1	Del HTML al XHTML: evolución y versiones.	Pág.1
2	Estructura de un documento HTML.	Pág.2
3	Identificación de etiquetas y atributos de HTML.	Pág.3
3.1	Clasificación de los atributos comunes según su funcionalidad.	Pág.4
3.2	Elementos HTML.	Pág.5
3.2.1	Elementos de la estructura básica del documento.	Pág.6
3.2.2	Elementos de la sección de cabecera.	Pág.7
3.2.3	Elementos que dan formato al texto de un párrafo.	Pág.8
3.2.4	Elementos de listas.	Pág.10
3.2.5	Elementos de tablas.	Pág.11
3.2.6	Elementos de formularios.	Pág.12
3.2.7	Elementos de frames.	Pág.14
3.2.8	Otros elementos.	Pág.14
4	XHTML frente a HTML.	Pág.15
4.1	XHTML: diferencias sintácticas y estructurales con HTML.	Pág.16
4.2	Ventajas e inconvenientes de XHTML sobre HTML.	Pág.17
5	Herramientas de diseño web.	Pág.18
6	Hojas de estilo o CSS.	Pág.19
6.1	Soporte de CSS en los navegadores.	Pág.20
6.2	Cómo incluir CSS en un documento HTML o XHTML.	Pág.21
6.2.1	Definir CSS en un archivo externo enlazado.	Pág.22
6.2.2	Definir CSS en un archivo externo importado.	Pág.22
6.2.3	Definir CSS en el documento HTML.	Pág.23
6.2.4	Incluir CSS en los elementos HTML.	Pág.23
6.3	Sintaxis de las reglas de estilo.	Pág.23
6.4	Atributos principales.	Pág.24
6.4.1	Atributos de color y fondo.	Pág.24
6.4.2	Atributos de fuente.	Pág.25
6.4.3	Atributos de texto.	Pág.25
6.4.4	Atributos de caja.	Pág.26
6.4.5	Atributos de clasificación.	Pág.28
6.5	CSS de posicionamiento.	Pág.29
6.6	Unidades de tamaño.	Pág.29
6.7	Definición y uso de clases.	Pág.30

1.- Del HTML al XHTML: 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.

La evolución de sus versiones, desde su creación hasta la creación del XHTML podemos verla a continuación:

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.

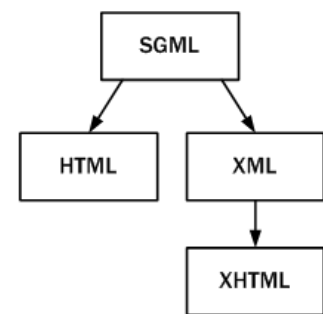
- **HTML 2.0** fue la primera versión oficial de HTML el IETF publicó el estándar en septiembre de 1995.
- **HTML 3.2** se publicó el 14 de Enero de 1997 por el W3C. Incorpora los applets de Java y texto alrededor de las imágenes.
- **HTML 4.0** se publicó el 24 de Abril de 1998. Entre las novedades que presenta se encuentran las hojas de estilos CSS y la posibilidad de incluir pequeños programas en las páginas web.
- **HTML 4.01** 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.

En 2004 Se funda WHATWG (Web Hypertext Application Technology Working Group) (grupo de trabajo formado por Apple, Mozilla y Opera, trabajan para continuar el desarrollo HTML manteniendo la compatibilidad con versiones anteriores). El W3C, tras interesarse en el desarrollo de HTML5 en el 2006, finalmente se une al grupo de trabajo para unificar el proyecto. En 2009 se produce la última llamada expedido para el proyecto de trabajo de HTML 5. El W3C no renueva XHTML 2.0. El borrador del W3C para HTML5 fue publicado en 2010 y HTML 5 se publicó en Octubre de 2014.

Tras la publicación del estándar HTML 4.01 se detecta su incompatibilidad con herramientas basadas en XML. Para evitar estos problemas se crea lenguaje XHTML que combina la sintaxis de HTML 4.0 con la 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** ya ha sido publicada en forma de borrador y pretende modularizar XHTML.
- En Octubre de 2010 se publica el XHTML 2.0.

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, XHTML no es más que una adaptación de HTML al lenguaje XML. Técnicamente, HTML es descendiente directo del lenguaje SGML, mientras que XHTML lo es del XML (que a su vez, también es descendiente de SGML). Las páginas y documentos creados con XHTML son muy similares a las páginas y documentos HTML.



Autoevaluación

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

- ☐ Sí.
- ☒ No.

2.- Estructura de un documento HTML.

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 5.0** existe una única DTD (**D**efinición de **T**ipo de **D**ocumento) en contra de versiones anteriores de HTML en las cuales existían varios tipos de definición.

Esta es `<!DOCTYPE html>` y a de colocarse antes de la etiqueta `<html>`, en la primera línea. Por otro lado indicar que no se considera como una etiqueta por lo cual no debe de cerrarse.

Al basarse en el lenguaje en SGML las versiones anteriores de HTML se debía especificar el DTD, pero en HTML5 no se basan en SGML, por lo que no existe necesidad de especificar el DTD , así que con un único DOCTYPE es suficiente.

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 4.0 //EN"
"http://www.w3.org/TR/REC-html40/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 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/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 4.0 Frameset//EN"
"http://www.w3.org/TR/REC-html40/frameset.dtd">
```

- **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>. En HTML 5.0 se deja de soportar la etiqueta <frameset> y </frameset> que podían sustituir a <body> y </body> respectivamente.

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.

3.- Identificación de etiquetas y atributos de HTML.

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 le 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.

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

- **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"	<div>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:</div> <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 Canada</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 Canada
Código	Idioma	Código	Idioma														
en	Inglés (Gran Bretaña)	es	Español														
en-US	Inglés americano	fr	Francés														
ja	Japones	fr-CA	Francés de Canada														
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.

3.2.- Elementos HTML.

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. Una vez que acaban, a continuación pueden aparecer más elementos en línea. Estos elementos están contenidos dentro de los elementos de bloque.
- **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 PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">

<html>
  <head>
    <title>Ejemplo de la 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="http://www.infoalisal.com">Los enlaces son elementos de
    línea</a>
    <p>Incluso si esta definido dentro de un párrafo, un texto en
    negrita</b>
    sigue siendo un elemento en línea. </p>
  </body>
</html>
```

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

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 en negrita sigue siendo un elemento en línea.

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.

3.2.1.- Elementos de la estructura básica del documento.

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, ...

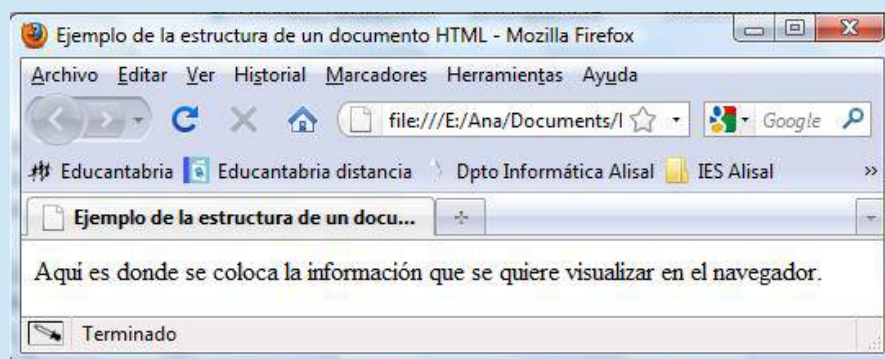
Un ejemplo de un documento HTML básico que utiliza estos elementos es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN" "http://www.w3.org/TR/REC-html40/loose.dtd">

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

  <body>
    Aquí es donde se coloca la información que se quiere visualizar en
    el navegador.
  </body>
</html>
```

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



Autoevaluación

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

☒ Verdadero. ☐ Falso.

3.2.2.- Elementos de la sección de cabecera.

Elementos contenedores:

Elemento	Descripción
title	Título del documento.
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.
link	Enlaces a documentos externos de librerías
meta	Información que agiliza la búsqueda del documento en buscadores.

3.2.3.- Elementos que dan formato al texto de un párrafo.

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

Elemento	Descripción
p	Delimita los párrafos
h <i>i</i>	Encabezado de nivel <i>i</i> , donde <i>i</i> es un número entero entre 1 y 6, ambos inclusive. El tamaño de la letra del encabezado es mayor cuanto menor sea el valor de <i>i</i> . No deben de usarse estas etiquetas para formatear texto. Sólo estarán bien usadas para designar títulos de párrafos.
hgroup	Una nueva forma de definir un encabezado para poder introducir elementos <i>h<i>i</i></i> y que no afecte al SEO.
b	Indica que el texto que está en ese elemento se le pondrá en negrita.
<i>i</i>	Indica que el texto que está en ese elemento se le pondrá en itálica ó cursiva.
<u>u</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.
strike	Indica que el texto que está en ese elemento está tachado.
small	Indica que el texto que está en ese elemento está más pequeño.
big	Indica que el texto que está en ese elemento está más grande.
em	Indica que el texto que está en ese elemento está en cursiva

Por ejemplo:

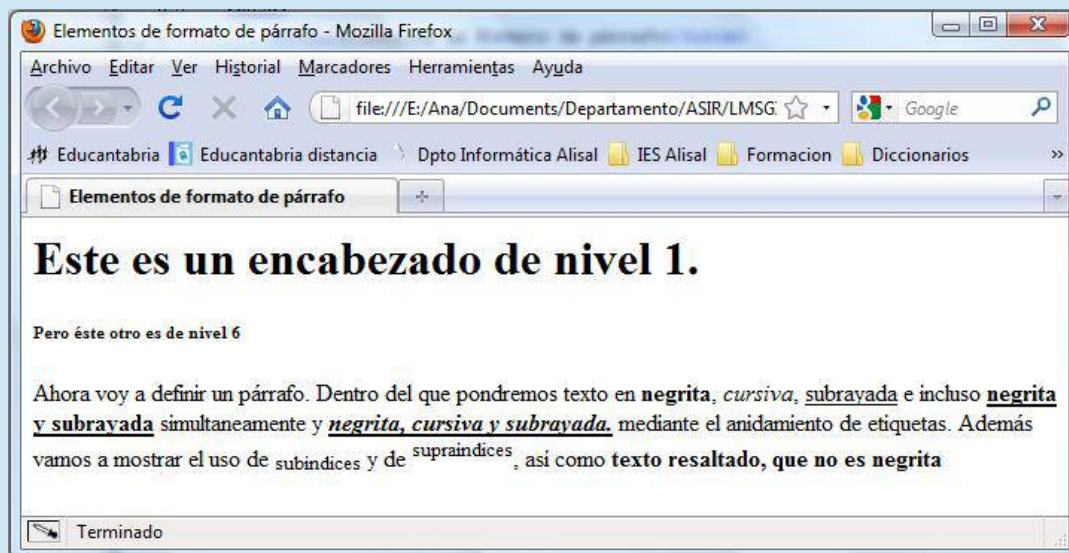
- Esta palabra la vamos a poner en `<i>cursiva</i>` y esta otra `también`
Esta palabra la vamos a poner en cursiva y esta otra también
- Esta palabra se va a escribir `<small>pequeñita</small>`, esta se va a escribir `<big>más grande</big>` y ésta `<big><big>más grande aún</big></big>`.
Esta palabra se va a escribir pequeñita, esta se va a escribir más grande y ésta más grande aún.
- Gracias a estas etiquetas podemos escribir cualquier expresión con símbolos matemáticos como esta: $H_{2}O$ o números elevados a potencias 7^{3} .
Gracias a estas etiquetas podemos escribir cualquier expresión con símbolos matemáticos como esta: $H_{2}O$ o números elevados a potencias 7^3 .
- Puedo proceder a tachar una palabra `<strike>así</strike>`, `<s>así</s>` o `así`
Puedo proceder a tachar una palabra así, así o así

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

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN" "http://www.w3.org/TR/REC-html40/loose.dtd">

<html>
  <head>
    <title>Elementos de formato de párrafo</title>
  </head>

  <body>
    <h1>Este es un encabezado de nivel 1.</h1>
    <h6>Pero éste otro es de nivel 6</h6>
    <p>Ahora voy a definir un párrafo. Dentro del que pondremos texto en <b>negrita</b>,
    <i>cursiva</i>, <u>subrayada</u> e incluso <b><u>negrita y subrayada</u></b> simultaneamente y
    <b><i><u>negrita, 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>
```




Autoevaluación

Para poner un párrafo en negrita utilizaremos la etiqueta:

- ☒ ****
- ☐ ****
- ☐ Ambas.
- ☐ **<n>**

3.2.4.- 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
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.
figure 	Representa una figura ilustrada como parte del documento.
div	Representa un contenedor genérico sin ningún significado especial.

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

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">

<html>
  <head>
    <title>Listas</title>
  </head>

  <body>
    <h3>Ejemplo de lista desordenada: Modulos de 1º de ASIR</h3>
    <ul> <li>Fundamentos de Hardware</li>
      <li>Gesti&ocute;n de Bases de Datos</li>
    </ul>

    <h3>Ejemplo de lista ordenada: Modulos de 1º de ASIR</h3>
    <ol> <li>Fundamentos de Hardware</li>
      <li>Gesti&ocute;n de Bases de Datos</li>
    </ol>

    <h3>Ejemplo de lista de definici&ocute;n: Modulos de 1º de ASIR</h3>
    <dl> <dt>Fundamentos de Hardware</dt>
      <dd>Componentes f&iacute;sicos de un ordenador</dd>
      <dt>Gesti&ocute;n de Bases de Datos</dt>
      <dd>Diseño y uso de bases de datos relacionales</dd>
    </dl>
  </body>
</html>
```

Ejemplo de lista desordenada: Modulos de 1º de ASIR

- Fundamentos de Hardware
- Gestión de Bases de Datos

Ejemplo de lista ordenada: Modulos de 1º de ASIR

1. Fundamentos de Hardware
2. Gestión de Bases de Datos

Ejemplo de lista de definición: Modulos 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

3.2.5.- Elementos de tablas.

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 PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<html>
  <head>
    <title>Tablas</title>
  </head>
  <body>
    <h3>Ejemplo de tabla: Notas del módulo de LMSGI de 1º de ASIR</h3>
    <table border = "1">
      <thead> <tr> <th>Eval</th> <th>LMSGI</th> </tr> </thead>
      <tfoot align="center"> <tr> <td>Media</td> <td>8</td> </tr> </tfoot>
      <tbody align="center">
        <tr> <td>Primera</td> <td>6</td> </tr>
        <tr> <td>Segunda</td> <td>7</td> </tr>
        <tr> <td>Tercera</td> <td>8</td> </tr>
      </tbody>
    </table>
  </body>
</html>
```

Ejemplo de tabla: Notas del módulo de LMSGI de 1º de ASIR

Eval	LMSGI
Primera	6
Segunda	7
Tercera	8
Media	8

3.2.6.- Elementos de formularios.

Lo elementos que puede contener 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 un formulario.
legend	Permite poner un título al fieldset.
label	Etiqueta de un campo del formulario.
Datetime	Permite elegir hora y fecha.
date	Para introducir la fecha.
week	Para introducir la semana.
month	Para introducir el mes.
time	Para introducir la hora.
number	Para introducir números.
range	Para introducir valores en un rango.
email	Para introducir email.
url	Para introducir url.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">
<html>
  <head>
    <title>Formulario</title>
  </head>
  <body>
    <h3>Ejemplo de formulario: Matriculación en el ASIR</h3>
    <form action="matricula.php" method="get">
      <b>Nombre:</b> <input type="text" name="Nombre"></input>
      <b>Apellidos:</b> <input type="text" name="Apellidos"></input>
      <br/><b>Sexo:</b> <input type="radio" name="Sexo"/>M
                       <input type="radio" name="Sexo"/>H
      <br/><br/>
      <b>Fecha de nacimiento:</b>
      <select>
        <option>dia</option><option>1</option><option>2</option>
        <option>3</option><option>4</option><option>5</option>
        ...
        <option>30</option><option>31</option>
      </select>
    </form>
  </body>
</html>
```

```

</select>
<select>
  <option>mes</option><option>Enero</option><option>Febrero</option>
  ...
  <option>Noviembre</option><option>Diciembre</option>
</select>
<select>
  <option>año</option><option>1992</option><option>1991</option>
</select>
<br/>
<p><b>Escoge los módulos en los que te matriculas: </b></p>
<table>
  <tr>
    <td>
      <input type="checkbox" name="Modulos"/>Hardware
    </td>
    <td>
      <input type="checkbox" name="Modulos"/>Red e Internet
    </td>
  </tr>
  <tr>
    <td>
      <input type="checkbox" name="Modulos"/>Bases de Datos
    </td>
    <td>
      <input type="checkbox" name="Modulos"/>Sistemas Operativos
    </td>
  </tr>
</table>
<br/><br/>
<b>Estudios previos:</b>
<br/>
<textarea rows="2" cols="50"></textarea>
<br/><br/>
<button type="button">Enviar</button>
<input type="reset" name="Reset"></input>
</form>
<p><b>Pulsa el botón de "enviar" para formalizar o "restablecer" para
limpiar.</b></p>
</body>
</html>

```

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:

☐ Hardware ☐ Red e Internet

☐ Bases de Datos ☐ Sistemas Operativos

Estudios previos:

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

Autoevaluación

Para crear un botón utilizaremos la etiqueta:

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

3.2.7.- Elementos de frames.

Los elementos utilizados para trabajar con frames son:

Elemento	Descripción
frameset	Define la partición de la ventana del navegador en marcos. Sólo puede partirse en filas o en columnas. Para partir la ventana del navegador en filas y columnas hay que anidar frames.
Frame	Define un marco que contiene información

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

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN" "http://www.w3.org/TR/REC-html40/frameset.dtd">
<html>
  <head> <title>Frames</title> </head>
  <frameset rows="20%,30%,*">
    <frame src="LMSGI02_Ejemplo07_1.html" /> <frame src="LMSGI02_Ejemplo03.html" />
    <frame src="LMSGI02_Ejemplo05.html" />
  </frameset>
</html>
```

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

Ejemplo de uso de frames

Permite visualizar simultáneamente varias páginas diferentes en el mismo navegador.

Este es un encabezado de nivel 1.

Pero éste otro es de nivel 6

Ahora voy a definir un párrafo. Dentro del que pondremos texto en **negrita**, *cursiva*, subrayada e incluso ***negrita y subrayada*** simultáneamente y ***negrita, cursiva y subrayada***, mediante el anidamiento de etiquetas. Además vamos a mostrar el uso de subíndices y de ^{supraindices}, así como texto resaltado, que no es **negrita**

Ejemplo de tabla: Notas del m&ocute;dulo de LMSGI de 1º de ASIR

3.2.8.- Otros elementos.

Otros elementos que podemos utilizar son:

Elemento	Descripción
a	Permite definir un enlace a una página web, un archivo o un correo.
img	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.
abbr	Indica una forma abreviada.
acronym	Indica un acrónimo.
blockquote	Contiene un bloque de texto con sangría.
q	Contiene una cita por lo que el navegador le añade las marcas de citación
br	Inserta una línea en blanco. No tiene etiqueta de cierre, sólo se abre.
article	Elemento que nos permite declarar un trozo del contenido como artículo. Ideal para blogs o periódicos.
aside	Representa un trozo de contenido que se relaciona muy levemente con el resto del contenido.
dialog	Elemento que permite reprensetar conversaciones.


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 transitional//EN"
"http://www.w3.org/TR/REC-html40/loose.dtd">

<html>
  <head>
    <title>Otros elementos</title>
  </head>

  <body>
    <h3>Ejemplo de uso de enlaces</h3>
    <a href="http://www.educantabria.es/">pínchame</a>

    <h3>Ejemplo de uso de imagenes</h3>
    </img>

    <h3>Ejemplo de uso de abreviaturas</h3>
    El <abbr title="Administración de Sistemas Informáticos y en
      Red">ASIR</abbr>
    comenzó a impartirse en Septiembre de 2010.

    <h3>Ejemplo de uso de acrónimos</h3>
    El protocolo web más usado es el <acronym title="Hypertext Transfer
      Protocol">HTTP</acronym>

    <h3>Ejemplo de uso de citas</h3>
    La frase <q>No hay nada repartido de modo más equitativo que la razón: todo
    el mundo está convencido de tener suficiente.</q> fué dicha por René
    Descartes (1596-1650) Filósofo y matemático francés.

    <h3>Ejemplo de uso de imágenes como enlaces</h3>
    Pulsa en la imagen <a href="http://www.edu.es/"></a>
  </body>
</html>
```

Autoevaluación

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.

4.- XHTML frente a HTML.

El lenguaje XHTML es muy similar al lenguaje HTML. De hecho, no es más que una adaptación de HTML al lenguaje XML. Las diferencias son básicamente sintácticas: la sintaxis HTML se basa en la norma SGML a pesar de no cumplirla estrictamente. Por otra parte, XHTML basa su sintaxis en la recomendación XML, y al igual que HTML no la cumple estrictamente.

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, podían escribirse en mayúsculas, en 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.

Autoevaluación

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.

4.1.- XHTML: diferencias sintácticas y estructurales con HTML.

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 nominal XHTML usando el atributo xmlns. El espacio nominal para XHTML es <http://www.w3.org/2002/06/xhtml12>
- Debe haber una declaración DOCTYPE (asocia un documento SGML o XML particular con una definición de tipo de documento) en el prólogo del documento. El identificador público incluido en la declaración DOCTYPE debe hacer referencia al siguiente esquema, siguiendo las referencias DTD en http://www.w3.org/TR/xhtml2/dtd_module_defs.html#a_dtd_module_defs

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 2.0//EN"
    "http://www.w3.org/Markup/DTD/xhtml12.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 (

) se puede utilizar la sintaxis
 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.

4.2.- Ventajas e inconvenientes de XHTML sobre HTML.

- **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.

Autoevaluación

El lenguaje XHTML permite:

- ☒ La información contenida en el documento se visualiza aunque el navegador no interprete el código de formato.
- ☒ Su funcionalidad presenta un rendimiento más alto que el de HTML.
- ☐ Al ser un lenguaje más novedoso que HTML es compatible con todas las herramientas de diseño que permiten trabajar con éste último.
- ☐ Ninguna respuesta es válida.

5.- Herramientas de diseño web.

Actualmente hay herramientas que permiten diseñar un sitio web sin necesidad de saber programar HTML. Las más populares:

- **Macromedia Dreamweaver.**
- **Microsoft Front Page.**
- **Adobe Go live.**
- **Wysiwyg web builder.**
- **NetObjects Fusion.**
- **Amaya** es una herramienta de libre distribución creada por el W3C que permite visualizar y editar páginas HTML y XHTML con hojas de estilo CSS, expresiones MathML y dibujos SVG; además de documentos XML. Actualmente se encuentra el proyecto detenido, y se detuvo en Amaya 11.4.4 (18 Enero 2012). **BlueGriffon** ha tomado el relevo de amaya

Además de las anteriores, para generar páginas web, es conveniente tener algunas de las siguientes herramientas:

- **Software de diseño**
 - **Google Web Designer**, para realizar animaciones, banners en HTML5, sustituto de **Macromedia Flash**.
 - **Macromedia Fireworks**, **InkScape** o **Adobe Illustrator** para diseñar botones, logos, imágenes, etc.
 - **Adobe Photoshop** o **Gimp**, para retocar fotografías y trabajar con imágenes.
- **Recursos: diseño web**
 - **My Fonts**, es un sitio web que vende fuentes, pueden ser utilizadas para la web.
 - **Color Voodoo**, tiene interesante información sobre el uso de los colores y su influencia en la web.
 - **moreCrayons**, una paleta de colores web seguros (Pues no todos los colores se ven iguales en distintos sistemas operativos).
 - **Yafla Color**, permite crear la gama a partir del color que se elija como principal además funciona como conversor entre los modelos de color RGB y HSV.
 - **ColorJack**, permite crear una gama a partir de un color y además calcula equivalencias de colores entre RGB, HSV y CMYK.
 - **Kuler**, es una aplicación online de Adobe Labs en la que se puede elegir una combinación de colores y compartirla con otros usuarios, que pueden votar sus favoritas. Requiere registro.

6.- Hojas de estilo o CSS.

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.

El organismo W3C propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML.

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, A diferencia de CSS2, que fue una gran especificación que definía varias funcionalidades, CSS3 está dividida en varios documentos separados, llamados "módulos". Actualmente, se siguen creando módulos para esta especificación. Algunos de estos módulos son, efectos de textos, bordes, animaciones, transformaciones 2D-3D, transiciones, etc.

El diseño web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.

HISTORIA DE CSS 3



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.

6.1.- Soporte de CSS en los navegadores.

El soporte de CSS de un navegador viene determinado por el motor del mismo ya que es éste el encargado de interpretar el CSS.

La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados en la actualidad:

Navegador	Motor	CSS 1	CSS 2.1	CSS 3
Safari	WebKit	Completo	Casi completo	Parcial
Opera	Presto	Completo	Casi completo	Parcial
Firefox	Gecko	Completo	Casi completo	Mayoría de Módulos.
Google Chrome	WebKit	Completo	Casi completo	Mayoría de Módulos.
Internet Explorer	Trident	Completo, desde la versión 6.0	Completo, desde la versión 8.0	Parcial.

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

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.

6.2.1.- Definir CSS en un archivo externo enlazado.

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.

Un ejemplo del uso de archivos externos CSS enlazados para la construcción de páginas web es:

```
<head>
    <title>Ejemplo de estilos CSS en fichero externo enlazado</title>
    <link rel="stylesheet" type="text/css" href="formatos.css" />
</head>
```

6.2.2.- Definir CSS en un archivo externo importado.

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

El ejemplo anterior quedaría:

```
<head>
    <title>Ejemplo de estilos CSS en fichero externo usando style</title>
    <style type="text/css">
        @import 'formatos.css';
    </style>
</head>
```


6.2.3.- Definir 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>`.

Un ejemplo en el que se utiliza este método de incluir formatos es:

```
<head>
  <title>Ejemplo de estilos CSS en la cabecera del propio documento</title>
  <style type="text/css">
    h3 { color: green; font-family: Times; }
    p { color: orange; font-family: Verdana; }
  </style>
</head>
```

6.2.4.- Incluir CSS en los elementos HTML.

Un ejemplo en el que se utiliza este método para incluir formatos es:

```
<body>
  <h3 style="color: green;">Ejemplo del uso de estilos CSS en el fichero</h3>
  <p style="color: orange; font-family: Verdana;">En esta ocasi&ocute;n el
  p&aacute;aacute;rroafo 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>
```

6.3.- 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, }.

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

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.

6.4.- Atributos principales.

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

- Atributos de color y fondo.
- Atributos de fuente.
- Atributos de texto.
- Atributos de caja.
- Atributos de clasificación.

Pasemos a verlos detenidamente.

6.4.1.- 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 en el que se utiliza este método para incluir formatos es:

```
<style type="text/css">
    body { background-color: black; color:yellow; }
    p { color: #ffffff;}
</style>
```

6.4.2.- 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 espaciado 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 en el que se utiliza este método para incluir formatos es:

```
<style type="text/css">
  body { background-color: black; color:yellow; font-family: courier }
  p { color: #ffffff; font:italic 900 12px Verdana; }
</style>
```

6.4.3.- 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 en el que se utiliza este método para incluir formatos es:

```
<style type="text/css">
  h3 { text-decoration:underline; text-align: center; text-transform: capitalize }
  p { text-indent: 50%; }
</style>
```

6.4.4.- 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-rigth-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, médium, thick o un tamaño.
border-rigth-width	Grosor del borde derecho. Sus valores posibles son: thin, médium, thick o un tamaño.
border-top-width	Grosor del borde superior. Sus valores posibles son: thin, médium, thick o un tamaño.
border-bottom-width	Grosor del borde inferior. Sus valores posibles son: thin, médium, 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.

```

<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>

```

Autoevaluación

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

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

6.4.5.- 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 en el que se utiliza este método para incluir formatos es:

```
<style type="text/css">
  h3 { white-space:pre;    }
  p { white-space:normal;  }
  ul { list-style:square; list-style-position: outside;}
</style>
```

6.5.- CSS de posicionamiento.

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 en el que se utiliza este método para incluir formatos es:

```
<style type="text/css">
  img{ position:absolute; left: 10px; top: 0px; z-index:-1;}
  p{overflow:scroll;}
</style>
```

6.6.- Unidades de tamaño.

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 de una letra del programador

☐ Verdadero. ☒ Falso.

6.7.- Definición y uso de clases.

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 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>
```

Autoevaluación

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

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