

Utilización de lenguajes de marcas en entornos web.

Caso práctico

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

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

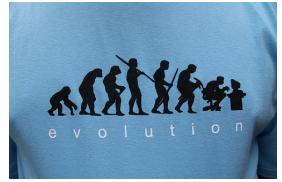
María le pregunta a **Juan** por su disponibilidad y capacidad para realizar dicha tarea. **Juan** le contesta que aprendió HTML en 1998. El boom de Internet le animó a interesarse por este campo y en este lenguaje realizó su primera página web. Desde entonces el lenguaje ha sufrido cambios en los que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



epifaniojk Juan trabajando. (Todos los derechos reservados)

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.



Germán Enrique Evolución. (Todos los derechos reservados)

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 y a la publicación de los borradores de HTML 5.0.

Tras la publicación del estándar HTML 4.01 se detecta su incompatibilidad con herramientas basadas en XML. Para evitar estos problemas se crea el 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.
- ✓ **XHTML 1.1** se publicó el 31 de mayo de 2001. Se basaba en un sistema de módulos y la definición de estos. Establecían una base para las futuras extensiones de XHTML.
- ✓ **XHTML 2.0** presentó su primer borrador el 5 de agosto de 2002. Hasta 2006 W3C siguió presentando borradores y no fue hasta diciembre de 2010 cuando se presentó versión definitiva.

En 2011 el W3C renunció al desarrollo del XHTML y se concentró en el HTML 5.

- ✓ **HTML 5.0** se publicó el 28 de octubre de 2014. Por su parte W3C siguió unos años lanzando sus versiones HTML 5.1 y HTML 5.2 como "Release Candidate". Pero al final se unió a WHATWG que desde 2011 abandonó el nombre de HTML 5 y pasó a denominarlo simplemente HTML, abandonando la idea de versiones en favor de una única versión continuamente modificada y mejorada.

En mayo de 2019, el W3C anunció que dejaba definitivamente el desarrollo del HTML en manos del WHATWG.

A continuación os indico un enlace muy interesante acerca de la evolución de la web, donde se observa gráficamente su linea temporal: [evolutionoftheweb](#)

Para saber más

Puedes revisar las últimas versiones de las especificaciones en los siguientes enlaces:

- ✓ [HTML 4.01](#)
- ✓ [HTML 5](#)
- ✓ [XHTML 1.0](#)
- ✓ [XHTML 2.0](#)

Autoevaluación

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

- Sí.
- No.

Respuesta errónea. Te recomiendo volver a leer el apartado.

Respuesta correcta.

Solución

1. Incorrecto
2. Opción correcta

1.1 HTML5



W3C. Logo de HTML5 (CC BY)

Como mención a parte, y a modo de iniciación en esta versión del lenguaje, HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del estándar creado en 1991 por Tim Berners-Lee y que contenía originalmente 22 elementos. La versión definitiva de HTML5 se publicó en octubre de 2014. importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML.

HTML5 ofrece nuevas características (elementos, atributos, manejadores de eventos, y APIs) para desarrollos de páginas web más simples y para formas más complejas de manejo. La especificación HTML5 se basa en HTML 4.01 Strict, pero a diferencia de especificaciones anteriores, HTML5 no utiliza una Definición de Tipo de Documento (DTD).

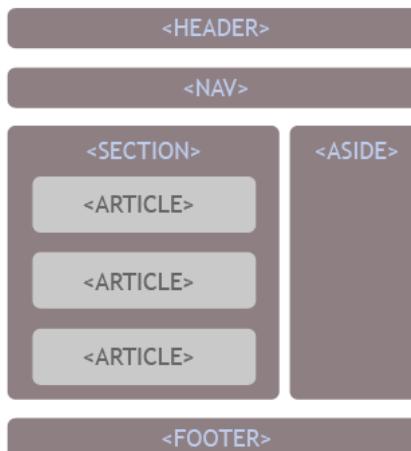
En su lugar, utiliza como base, el Modelo de Objetos del Documento (DOM, el "árbol" creado por la estructura del documento), en vez de un conjunto determinado de reglas sintácticas.

Se diferencia también de las especificaciones anteriores en que se incluyen instrucciones detalladas de cómo los navegadores deben utilizar el HTML mal especificado.

Hay muchas novedades en HTML5, pero las más destacables son las siguientes:

- ✓ **Archivos Multimedia:** HTML5 introduce soporte integrado para el contenido multimedia gracias a los elementos
- ✓ **Web Semántica:** Con la introducción de etiquetas como header, nav, section, aside, footer, article, hgroup y figure, entre otras, la W3C pretende hacer más descriptivo el lenguaje. De esta forma, los desarrolladores por un lado, pueden definir mejor la importancia o la finalidad de cada una de las partes de una página web y por otro lado, se pretende hacer el código más entendible para los buscadores que gracias a la nueva información podrán también detectar mejor que es que en una web.

Los formularios también han sido mejorados con validaciones sin necesidad de Javascript, varios atributos nuevos y se han añadido una buena cantidad de elementos semánticos para los inputs de los mismos.



W3C. Nuevas etiquetas HTML5 (CC BY)

A continuación puedes descargar una hoja-resumen con las etiquetas en HTML5 y su significado: (creada por @Manz - www.emezeta.com)

- ✓ [Versión completa \(Inglés\)](#)
- ✓ [Versión reducida \(Castellano\)](#)

Podéis obtener más información acerca de HTML5 en las siguientes direcciones:

- ✓ <http://es.wikipedia.org/wiki/HTML5>
- ✓ <https://lenguajehtml.com/p/html/>
- ✓ <http://www.edu4java.com/es/web/html5/introduccion.html>
- ✓ <http://www.maestrosdelweb.com/?s=html5>

Caso práctico

En este ejemplo vamos a ver cómo se maqueta una página en HTML5 junto con CSS, usando las nuevas etiquetas que nos proporciona HTML5 como son:



...

Código HTML5:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<title>Ejemplo en HTML5</title>
<link rel="stylesheet" href="css/estilo.css">
</head>
<body>
    <!-- Cabecera -->
    <header>
        
        <h1>Ejemplo de maquetación en HTML5</h1>
    </header>
    <!-- Contenido -->
    <section>
        <figure>
            
            <figcaption>Figura: descripción foto</figcaption>
        </figure>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean ornare sapien ac nibh vulputate feugiat. <p>Donec ligula arcu, sagittis lacinia nibh et, consequat gravida turpis. Aenean nec scelerisque felis. Inte</p>
    </section>
    <!-- Contenido relacionado-->
    <aside>
        <p>Contenido Relacionado</p>
    </aside>
    <!-- Pie de pagina -->
    <footer>
        <a href="http://www.ejemplocodigo.com">www.ejemplocodigo.com</a>
    </footer>
</body>
</html>
```

Código CSS:

```
html,body {
    height:100%;
}

header {
    display:block;
    background:#286af0;
    padding:10px 0px;
}

section {
    width: 79%;
    background: #ccc;
    float: left;
    overflow: auto;
    padding-bottom: 60px;
    padding-top:30px;
}

aside {
    float: right;
    border: 1px solid red;
    width: 19%;
    border: 1px solid red;
    /*Si quisieramos ocultar el contenido relacionado pondriamos visibility a hidden*/
    /*visibility: hidden;*/
}

footer {
    position: relative;
    margin-top: -50px;
    height: 40px;
    padding:5px 0px;
    clear: both;
    background: #286af0;
    text-align: center;
    color: #fff;
}
```

```
figure {  
    display: table; margin: 0 auto;  
}
```

2.- Estructura de un documento HTML.

Caso práctico

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

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

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

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

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

- ✓ Para la versión HTML 5 se simplifica enormemente el prólogo.

- ◆ **HTML 5.** Utiliza una única versión DTD. Además no hace falta indicar parámetros extra en la definición DOCTYPE.

- ✓ **Ejemplar:** En un documento HTML está delimitado por las etiquetas y . El ejemplar puede, a su vez dividirse en dos partes:

- ◆ *La cabecera*, delimitada por las etiquetas y . 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 . Esta información no se presentará en la ventana del navegador, salvo el título que aparecerá en la barra de título de la parte superior.
 - En la cabecera puede utilizarse también el elemento que puede forzar a que la página activa se cargue cada cierto tiempo (indicado en segundos mediante el atributo CONTENT): (El navegador recargará la página cada 10 segundos). Esto ha de ser utilizado con mucha precaución ya que podrá sobrecargar el servidor. Si el contenido de la página no va a cambiar es inútil hacerlo, de hecho, solo tendrá utilidad en casos muy especiales.
- ◆ *El cuerpo*, contiene la información que se va a presentar en la pantalla. Está limitado por las etiquetas y , salvo en los documentos de tipo HTML 4.0 Frameset donde éstas se sustituyen por y .

El siguiente ejemplo muestra la estructura de un documento HTML:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Aquí va el título del documento</title>
  </head>
  <body>
    <!-- Cuerpo del documento -->
  </body>
</html>
```



Alberto Jiménez. Estructura (Todos los derechos reservados)

Autoevaluación

¿Cuál de las siguientes afirmaciones es verdadera?

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

Incorrecto.

No es la opción correcta.

Correcto.

No es correcto.

Solución

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

3.- Identificación de etiquetas y atributos de 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.

```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
  <title>sample</title>
</head>
<body>
  <p>Voluptatem accusantium
    totam rem aperiam.</p>
</body>
</html>
```

HTML

Dreftymac. Documento HTML (CC BY-SA)

Un documento HTML está formado por etiquetas y atributos.

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

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

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

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

Autoevaluación

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

- Verdadero.
- Falso.

No es correcto.

Correcto.

Solución

1. Incorrecto
2. Opción correcta

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



Ana Isabel Velázquez. Atributos y elementos (Todos los derechos reservados)

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

Atributo	Descripción	Ejemplo
name = "texto"	Permite asignar el nombre "texto" a un objeto HTML	<input type="text" name="fname" id="fname">
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. Mejora la accesibilidad y los navegadores lo muestran cuando el usuario pasa el ratón por encima del elemento.	<p title="Unidad 2">Unidad 2</p>
id = "texto"	Permite identificar al elemento HTML sobre el que se aplica de forma única mediante el identificador "texto".	<h1 id="MiCiudad">Mi Ciudad</h1>
style = "texto"	Permite aplicar al elemento HTML el estilo "texto" directamente.	<body style="background-color:blue;"> <p style="color:red;">Esto es un párrafo rojo</p>
class = "texto"	Permite aplicar al elemento HTML el estilo "texto" definido en las CSS.	<h2 class="city">Londres</h2>

La mayoría de páginas web actuales utilizan los atributos id y class de forma masiva. Sin embargo, estos atributos sólo son realmente útiles cuando se trabaja con CSS y con Javascript.

Respecto al valor de los atributos id y class, sólo pueden contener guiones medios (-), guiones bajos (_), letras y/o números, pero no pueden empezar por números. Además, los navegadores distinguen mayúsculas de minúsculas y no se recomienda utilizar letras como ñ y acentos, ya que no es seguro que funcionen correctamente en todas las versiones de todos los navegadores.

- ✓ **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	Ejemplo
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.	<p dir="rtl">la dirección del texto es de derecha a izquierda</p>
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:	<html lang="es">

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.				<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">

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 los vamos a contemplar.

3.2.- Elementos HTML.



[Ana Isabel Velázquez](#), Elementos de linea y de bloque (Todos los derechos reservados)

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>Ejemplo de la diferencia entre los elementos en l&#xed;nea 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.infoalusal.com">Los enlaces son elementos de l&#xed;nea</a> <p>Incluso si esta definido dentro de un párrafo</p>
  </body>
</html>
```

[Visualización del código en un navegador](#)

Al publicarlo en un navegador, 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.

Captura de pantalla del navegador Firefox bajo Windows.. Elementos de linea y de bloque (Todos los derechos reservados)

Autoevaluación

Los elementos de línea y de bloque se diferencian en:

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

Incorrecto.

No es la respuesta correcta.

Correcto.

No es correcto.

Solución

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

3.2.1.- 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, ...

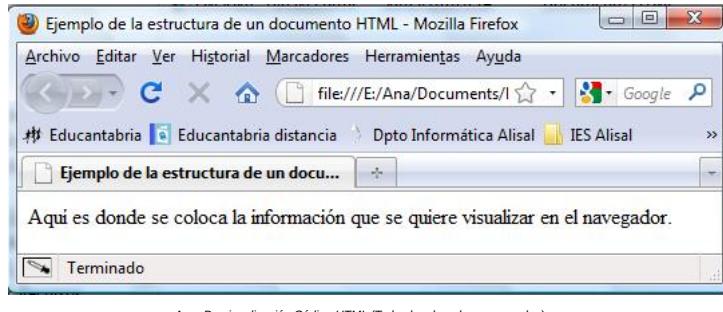
Un ejemplo de un documento HTML básico que utiliza estos elementos es:

```
<!DOCTYPE html>

<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.

Correcto.

Incorrecto.

Solución

1. Opción correcta
2. Incorrecto

3.2.2.- Sección de cabecera.



Aprendiendo programación Tatuaje HTML.
(Todos los derechos reservados)

La sección de cabecera puede contener los siguientes elementos:

✓ Elementos contenedores:

Elemento	Descripción	Ejemplo
title	Título del documento.	<pre><!DOCTYPE html> <html> <head> <title>Tituto de la web</title> </head> <body> <h1>Esto es un titular</h1> <p>Esto es un párrafo</p> </body> </html></pre>
script	Script referenciado o incrustado. Se puede agregar entre las etiquetas y entre las etiquetas	<pre><!-- Script referenciado en HTML4 y (x)HTML --> <script type="text/javascript" src="javascript.js"></script> <!--Script referenciado en HTML5 -->
<script src="javascript.js"></script> <!--Script incrustado que muestra la hora actual --> <html> <head> <title>La hora actual es: </title> </head> <body> <script> let d = new Date(); document.body.innerHTML = "<h1>Hora: " + d.getHours() + ":" + d.getMinutes() + ":" + d.getSeconds() + "</h1>"; </script> </body> </html></pre>
style	Estilo aplicado al documento utilizando CSS.	<pre><html> <head> <title>Mi primera página con estilo</title> <style type="text/css"> body { color: purple; background-color: #d8da3d } </style> </head> <body> Cuerpo de la web </body> </html></pre>

✓ Elementos no contenedores:

Elemento	Descripción

base	Especifica la dirección URL base que se utilizará para todas las direcciones URL relativas contenidas dentro de un documento. Sólo puede haber un elemento en un documento	<base href="http://www.ejemplo.es/inicio.html">
link	Brinda la posibilidad de añadir información externa afín al documento. Estos recursos externos podrían ser información de estilo (CSS), ayuda para facilitar la navegación, información en otro formato (RSS), información de contacto, etc.	<pre> <!-- Enlaces que mejorarán la navegación. Start (inicio) lleva al punto de inicio de la sección. Prev (anterior) lleva al punto anterior en la secuencia (Venus) Next (siguiente) lleva al punto siguiente en la secuencia (Marte) --> <link rel="Start" href="/sistema-solar/" /> <link rel="Prev" href="/sistema-solar/venus/" /> <link rel="Next" href="/sistema-solar/marte/" /> <!-- Referencia al índice de la sección --> <link rel="Contents" href="/sistema-solar/contenidos.html" /> <!-- ofrece la posibilidad de vincular una página que ayude a los visitantes a utilizar <link rel="Help" href="/ayuda-web.html" /> <!-- Una manera de contactar con el autor del sitio web --> <link href="mailto:webmaster@ejemplo.es" rev="made" /> <!-- Especifica la CSS a aplicar al sitio web. Por ejemplo, con este vínculo, especifica <link rel="stylesheet" type="text/css" media="screen" href="/estilo/astro.css" /> <!-- Da acceso a la traducción del documento en otro idioma, en este caso el Francés --> <link rel="alternate" href="/sistema-solar/tierra.fr" hreflang="fr" title="Traducción al Francés" /> </pre>
meta	Información que indican a los buscadores que indexan la página por qué términos debe ser encontrada, por lo que agiliza la búsqueda del documento en buscadores. Dependiendo de la utilización, caracterización y objetividad de dichos meta, más otros factores SEO se puede conseguir una excelente posición en el listado resultante de una búsqueda.	<pre> <meta charset="UTF-8"> <meta name="description" content="Free Web tutorials"> <meta name="keywords" content="HTML, CSS, JavaScript"> <meta name="author" content="John Doe"> <meta name="viewport" content="width=device-width, initial-scale=1.0">
 </pre>

3.2.3.- 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
hi	Encabezado de nivel i, donde 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. No deben de usarse estas etiquetas para formatear texto. Sólo estarán bien usadas para designar títulos de 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.

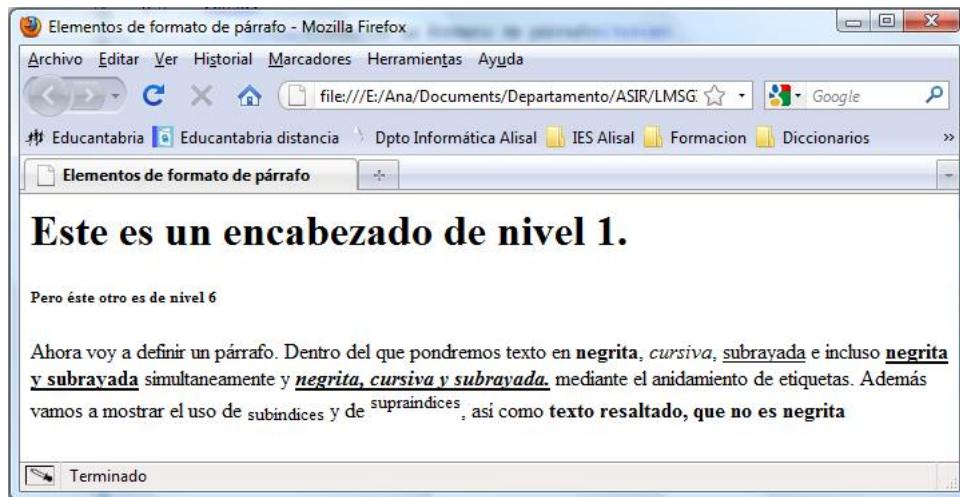
Un ejemplo de un documento HTML que utiliza estos elementos es:

```
<!DOCTYPE html>

<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> simultáneamente y <b><i><u>negrita, cursiva y subrayada.</u></i></b> mediante el anidamiento de etiquetas. Además vamos a mostrar el uso de <sub>sub&iacute;ndices</sub> y de <sup>supra&iacute;ndices</sup>, así como <strong>texto resaltado, que no es negrita</strong>
        </p>
    </body>
</html>
```

El resultado en el navegador sería:



Autoevaluación

Para poner un párrafo en negrita utilizaremos la etiqueta:

-
-
- Ambas.
- <n>

Correcto.

Incorrecto.

No es la respuesta correcta.

No es correcto.

Solución

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

3.2.4.- 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.

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

```
<!DOCTYPE html>

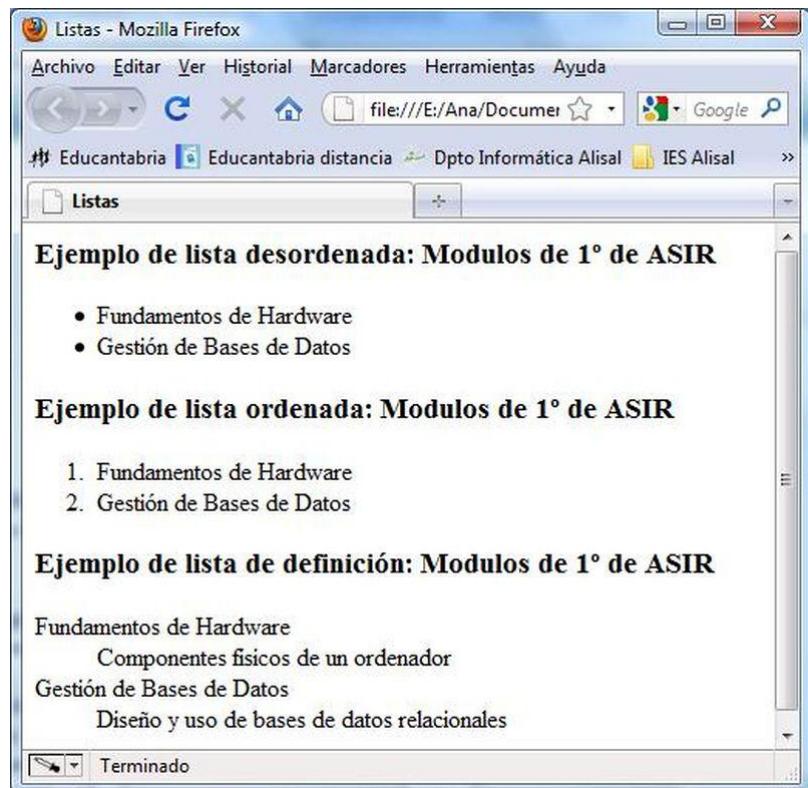
<html>
    <head>
        <title>Listas</title>
    </head>

    <body>
        <h3>Ejemplo de lista desordenada: Modulos de 1º de ASIR</h3>
        <ul> <li>Fundamentos de Hardware</li>
            <li>Gestión de Bases de Datos</li>
        </ul>

        <h3>Ejemplo de lista ordenada: Modulos de 1º de ASIR</h3>
        <ol> <li>Fundamentos de Hardware</li>
            <li>Gestión de Bases de Datos</li>
        </ol>

        <h3>Ejemplo de lista de definiciones: Modulos de 1º de ASIR</h3>
        <dl> <dt>Fundamentos de Hardware</dt>
            <dd>Componentes físicos de un ordenador</dd>
            <dt>Gestión de Bases de Datos</dt>
            <dd>Diseño y uso de bases de datos relacionales</dd>
        </dl>
    </body>
</html>
```

El resultado en el navegador sería:



Ana. Previsualización del código HTML (Todos los derechos reservados)

3.2.5.- 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>

<html>
  <head>
    <title>Tablas</title>
  </head>

  <body>
    <h3>Ejemplo de tabla: Notas del m&oacute;dulo 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>
```

El resultado en un navegador sería:

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

Eval	LMSGI
Primera	6
Segunda	7
Tercera	8
Media	8

Víctor Gómez. Previsualización del código HTML ([CC0](#))

3.2.6.- Formularios.

Los elementos que puede contener un formulario son los siguientes:

Elemento	Descripción
form	Delimita el contenido del formulario(es un elemento que permite a la página web solicitar información al usuario)
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.

Ejercicio resuelto

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

The screenshot shows a Mozilla Firefox browser window with the title "Formulario - Mozilla Firefox". The address bar shows the URL: file:///E:/Ana/Documents/Departamento/ASIR/LMSGI02/ASIR_LMSGI/. The page content is as follows:

Ejemplo de formulario: Matriculación en el ASIR

Nombre: Apellidos: Sexo: M H

Fecha de nacimiento: dia mes año

Escoge los módulos en los que te matriculas:

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

Estudios previos:

Enviar Restablecer

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

Terminado

[Mostrar retroalimentación](#)

El código HTML asociado a ese documento lo puedes visualizar a continuación:

```
<!DOCTYPE html>

<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>
            &nbsp;&nbsp;&nbsp;<b>Apellidos:</b> <input type="text" name="Apellidos"></input>
            &nbsp;&nbsp;&nbsp;<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>6</option>
                <option>7</option>
                <option>8</option>
                <option>9</option>
                <option>10</option>
                <option>11</option>
                <option>12</option>
                <option>13</option>
                <option>14</option>
                <option>15</option>
                <option>16</option>
                <option>17</option>
                <option>18</option>
                <option>19</option>
                <option>20</option>
                <option>21</option>
                <option>22</option>
                <option>23</option>
                <option>24</option>
                <option>25</option>
                <option>26</option>
                <option>27</option>
                <option>28</option>
                <option>29</option>
                <option>30</option>
                <option>31</option>
            </select>
            <select>
                <option>mes</option>
                <option>Enero</option>
                <option>Febrero</option>
                <option>Marzo</option>
                <option>Abril</option>
                <option>Mayo</option>
                <option>Junio</option>
                <option>Julio</option>
                <option>Agosto</option>
                <option>Septiembre</option>
                <option>Octubre</option>
                <option>Noviembre</option>
                <option>Diciembre</option>
            </select>
            <select>
                <option>año</option>
                <option>1992</option>
                <option>1991</option>
                <option>1990</option>
                <option>1989</option>
                <option>1988</option>
            </select>
        </form>
    </body>
</html>
```

```

        <option>1987</option>
        <option>1986</option>
    </select>
    <br/>
    <p><b>Escoge los módulos en los que te matriculas: </b></p>
    <table>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/> Fundamentos de Hardware
            </td>
            <td>
                <input type="checkbox" name="Modulos"/> Servicios de Red e Internet
            </td>
        </tr>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/>Gestión de Bases de Datos
            </td>
            <td>
                <input type="checkbox" name="Modulos"/>Administración de sistemas Operativos
            </td>
        </tr>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/>Implantación de Sistemas Operativos
            </td>
            <td>
                <input type="checkbox" name="Modulos"/>Implantación de Aplicaciones Web
            </td>
        </tr>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/>Planificación y Administración de Redes
            </td>
            <td>
                <input type="checkbox" name="Modulos"/>Administración de Sistemas Gestores de Bases de Dato
            </td>
        </tr>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/>Lenguajes de Marcas y Sistemas de Gestión de Informa
            </td>
            <td>
                <input type="checkbox" name="Modulos"/>Seguridad y Alta Disponibilidad
            </td>
        </tr>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/>Formación y Orientación Laboral
            </td>
            <td>
                <input type="checkbox" name="Modulos"/>Empresa e Iniciativa Emprendedora
            </td>
        </tr>
        <tr>
            <td>
                <input type="checkbox" name="Modulos"/>Proyecto de Administración de Sistemas Informáticos
            </td>
            <td>
                <input type="checkbox" name="Modulos"/>Formación en Centros de Trabajo
            </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 la matrícula o el botón de "restablecer" para limpiar el f
</body>
</html>

```

[Visualización en un navegador](#)

Autoevaluación

Para crear un botón utilizaremos la etiqueta:

-
-
- Ambas.
- Ninguna.

Pero no sólo con esta.

Fíjate bien ya que hay una respuesta más correcta.

Correcto.

No es la opción correcta.

Solución

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

3.2.7.- Frames.

Este apartado tan sólo lo nombraremos, ya que están en completo desuso actualmente.

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>
```

El contenido del fichero LMSGI02_Ejemplo07_1.html es:

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

<html>
  <head> <title>Frames_2</title> </head>

  <body>
    <h1>Ejemplo de uso de frames</h1>
    <p>Permite visualizar simultáneamente varias páginas diferentes en el mismo navegador.</p>
  </body>
</html>
```

Los ficheros LMSGI02_Ejemplo03.html y LMSGI02_Ejemplo05.html se corresponden con los códigos de ejemplos que hemos visto anteriormente para mostrar el uso de los formatos de texto y las tablas.

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

Frames - Mozilla Firefox

Archivo Editar Ver Historial Marcadores Herramientas Ayuda

file:///E:/Ana/Documents/LMSGI/Tema02/ASIR_LMSGI02_Codigos_Eje_Ti Google

Educantabria Educantabria distancia Dpto Informática Alisal IES Alisal Formación Diccionarios Editoriales Correos

Frames

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 ^{suprindice}s, así como **texto resaltado, que no es negrita**

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

Eval	LMSGI
Primera	6
Segunda	7
Tercera	8
Media	8

Terminado

Ana. Previsualización del código HTML (Todos los derechos reservados)

3.2.8.- Capas



Las **capas** se pueden definir como páginas que se pueden incrustar dentro de otras, es decir, que son bloques con contenido html que pueden situarse en la página de manera dinámica y organizables en 3D.

Los **atributos de una capa** (posición, visibilidad, etc..), como los de cualquier otro elemento HTML, pueden y deben definirse dentro de una hoja de estilo. Realmente tienen sentido cuando se les aplican estilos CSS, pues es cuando sus ventajas se pueden aprovechar al 100%. Es quizás la mejor ilustración de lo que significa separar contenido de presentación. Gracias a las capas o bloques

... es posible hacer que una página web se adapte al dispositivo desde el que se visualiza: móvil, tablet desktop.

Su **contenido**, en cambio, siempre deberá ser especificado dentro de la parte principal de la página. Como se puede ver, se está siguiendo la filosofía de separar el contenido y la forma de representarlo.

DatoWeb. Superponer capas (Todos los derechos reservados)
y modificadas desde un lenguaje de *script*. Desgraciadamente, las implementaciones de los distintos navegadores pueden ser incompatibles entre sí, por lo que resulta complicado escribir código que funcione de la misma forma en distintas plataformas, de ahí que un buen diseñador y programador web verifique su web en las distintas plataformas existentes y se asegure de que los resultados sean similares.

La única manera común de definir capas para distintos navegadores es mediante **hojas de estilo**. En realidad las capas no se definen completamente mediante el lenguaje html, necesitan del lenguaje de definición de estilos CSS. Entre ambos podemos colocar en nuestras páginas estos elementos móviles, ocultables y en general manipulables de forma dinámica.

La definición de una capa sigue la misma estructura que la que se usaba para decidir las características de una etiqueta con el atributo *id*, por ejemplo:

```
<style type="text/css">
.mi_capa {
    position: absolute;
    top: 100px;
    left: 20px;
    width: 300px;
    height: 200px;
</style>
```

Con esto hemos definido una clase denominada *mi_capa*, cuya altura (*height*) es de 200 px y anchura (*width*) 300 px. Además está situada a 100 px de la parte superior (*top*) y a 20 px del margen izquierdo de la página (*left*). Repito que hemos creado una clase para describir como es la capa, pero no hemos construido la capa. Para construirla usamos la etiqueta

y el atributo *class*.

```
<div class="mi_capa">
<h1>Titolo</h1>
<p>Continúa con más contenido</p>
...
</div>
```

Cualquier bloque

con *class="mi_capa"* estará en esa posición y con ese tamaño.

Esta capa puede colocarse en cualquier parte de la ventana del navegador, pues su posición es absoluta (*absolute*). Pero también podemos definir capas de posicionamiento relativo (*relative*), es decir, que más que definir las coordenadas de suposición respecto a la ventana, describimos su posición respecto al lugar donde aparecerá en el texto. En otras palabras: describimos el desplazamiento de la capa respecto de donde la escribimos. Se definen así:

```
<style type="text/css">
.relativa
{
    position: relative;
    left: 20px;
    top: 100px;
}
</style>
```

Este ejemplo te muestra dos capas, una absoluta y otra relativa, esta última está construida con una etiqueta , para evitar el salto de línea propio de los bloques.

Elemento	Descripción
----------	-------------

div	Permite crear una sección dentro del documento XHTML.
id	Permite asignar un identificador único al elemento <i>div</i> .
style	Permite dar un estilo al contenido del elemento empleando las hojas de estilo.
class	Permite dar un estilo al contenido del elemento empleando las hojas de estilo.

Para saber más

Podéis ampliar información acerca de las capas en estos enlaces:

- ✓ [Qué son las capas \(desarrolloweb\)](#)
- ✓ [La estructuración con Capas o Divs](#)

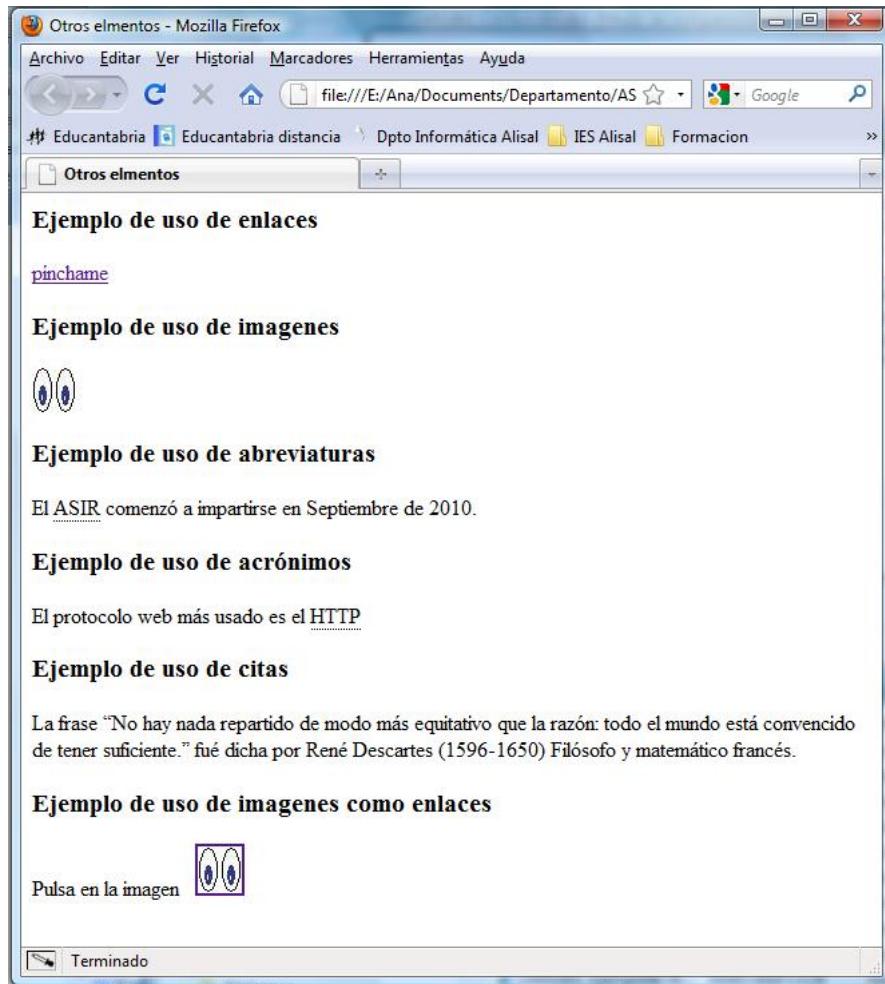
3.2.9.- Otros elementos.

Otros elementos que podemos utilizar son:

Elemento	Descripción
a	Permite definir un enlace a una página web, un archivo o una dirección de correo.
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.

Ejercicio resuelto

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



[Mostrar retroalimentación](#)

El código HTML asociado a ese documento podría ser el siguiente:

```

<!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/">p&iacute;nchame</a>

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

    <h3>Ejemplo de uso de abreviaturas</h3>
    El <abbr title="Administraci&acute;n de Sistemas Inform&acute;aticos y en Red">ASIR</abbr>
    comenz&acute; a impartirse en Septiembre de 2010.

    <h3>Ejemplo de uso de acr&acute;fonimos</h3>
    El protocolo web m&acute;s 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 m&acute;as equitativo que la raz&acute;n: todo el mundo
    est&acute; convencido de tener suficiente.</q> fu&eacute; dicha por Ren&eacute; Descartes (1596-1650)
    Fil&acute;sofo y matem&acute;tico franc&eacute;s.

    <h3>Ejemplo de uso de im&acute;genes como enlaces</h3>
    Pulsa en la imagen &nbsp;&nbsp;<a href="http://www.educantabria.es/"></a>
  </body>
</html>

```

[Código fuente de la imagen](#)

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.

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

Solución

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

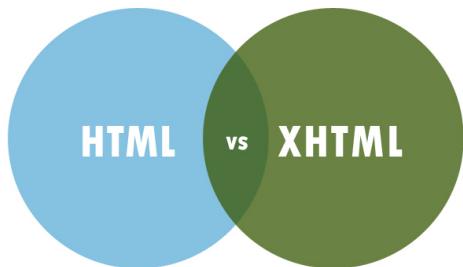
4.- XHTML frente a HTML.

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.



Víctor Gómez. HTML o XHTML (CC0)

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

El lenguaje HTML tiene una sintaxis muy permisiva, por lo que es posible escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas, por ejemplo, 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.

No es correcto.

No es la opción correcta.

Correcto.

Incorrecto.

Solución

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

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 .
- ✓ 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/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:



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.

Correcto:

```
<strong><em>Hola</em></strong>
```

Incorrecto:

```
<em><strong>Hola</em></strong>
```

- ✓ 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 y deben tener algún valor.

Correcto:

```
<option selected="selected">Ejemplo correcto</option>
```

Incorrecto:

```
<option selected>Ejemplo incorrecto</option>
```

- ✓ 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.

Correcto:

```
<img> </img> o bien <img/> o bien <img />
```

Incorrecto:

```
<br>
```

Otras restricciones:

Además de las cuatro 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 () para evitar que el navegador interprete de forma errónea caracteres como & y <.

```
<script type="text/javascript">
<![CDATA[
//contenido del script sin que XML tenga que preocuparse por lo que haya dentro...
]]>
</script>
```

- ✓ Las páginas XHTML deben prescindir del atributo name en su lugar, siempre debe utilizarse el atributo id.
- ✓ En 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.

Para saber más

Para ampliar conocimientos tenéis el enlace de abajo:

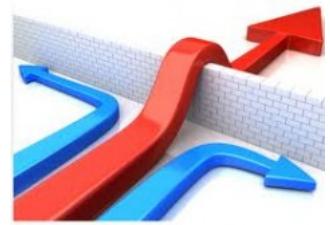
[Introducción a XHTML](#)

En este sitio web tenéis también a vuestra disposición numerosos ejemplos y ejercicios resueltos.

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.).



[InfoCrowdsourcing](#), Ventajas e Inconvenientes
(Todos los derechos reservados)

✓ 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.

[Mostrar retroalimentación](#)

Solución

1. Correcto
2. Correcto
3. Incorrecto
4. Incorrecto

5.- Herramientas de diseño web.

Caso práctico

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

Tras verla Félix tiene la curiosidad de saber si existen herramientas o recursos para facilitar el trabajo de creación y diseño de la web.

Juan le cuenta que puede bastar una aplicación de desarrollo de código, como Sublime Text, pero que existen muchas herramientas que facilitan la tarea de diseño.

El universo de las herramientas de diseño web es cada vez más amplio. Sobre todo por los avances tecnológicos que propician el surgimiento de nuevas plataformas con innovadoras funciones. Hay algunas que destacan por las facilidades, optimización y el ahorro de tiempo que brindan.

Sin embargo, por esa misma razón, a veces es complicado saber con exactitud cuál es la mejor herramienta de diseño web, o por lo menos, cuál es la que necesitas en función de tu nivel de conocimientos y la página que deseas crear.

A continuación mostramos **herramientas de edición de código, software de diseño web y recursos** que son muy útiles en la actualidad. Prueba las opciones y decide cuál satisface mejor tus necesidades; algunas son totalmente gratuitas, mientras que otras tienen versiones libres con características suficientes.

- ✓ [Adobe Color CC](#): La aplicación de Adobe Color CC mejora el color de tus fotografías, ya que analiza el tono de las ilustraciones y busca soluciones armónicas. Así, ayuda a la combinación de paletas cromáticas de forma armoniosa para cada uno de tus proyectos. No importa qué tipo de sitio web diseñas: Adobe Color CC logrará hacerlo ver atractivo con imágenes perfectamente equilibradas.

Esta herramienta te permite realizar ajustes de color, tono y saturación, curvas y máscaras de luminosidad. Si no tienes claro cómo combinar los colores, aquí tendrás una gama de tonalidades a partir de una base.

Es gratuita y fácil de usar. Está disponible en dos aplicaciones de escritorio de Creative Cloud: Adobe InDesign y Adobe After Effects; también puedes acceder a través de Photoshop. Si requieres una aplicación móvil, la encuentras como Adobe Capture CC.

- ✓ [Google Fonts](#): Una página web completa, además de ser agradable y estar optimizada, debe contar con diversos elementos que la hagan resaltar. Uno de los más importantes es la tipografía. Google Fonts te garantiza que tu fuente será original y adecuada para la imagen corporativa o el estilo que imaginas, en tanto que aquí encuentras un catálogo gratuito con más de 800 opciones. Y si necesitas un poco de inspiración, también puedes visitar su sección [Featured](#) donde conocerás muchas colecciones temáticas creadas por la misma herramienta.

Es muy intuitiva y visual. Dentro de su aplicación encontrarás filtros de búsquedas que puedes determinar por categorías y opciones de prueba para elegir la que mejor se adapte a tu proyecto.

- ✓ [Whatfontis](#): ¿Encontraste una tipografía que llamó tu atención, pero nunca supiste cuál era? Muchas fuentes pueden ser muy conocidas, pero entre tantas que se crean día con día (hay unas 500 mil registradas en el mundo), cada vez es más complicado identificarlas. Por fortuna, en la actualidad ya no tienes que quedarte con la duda sobre cuál fuente viste, ya que con a WhatFontis podrás saberlo.

Esta herramienta es la más potente del mercado gracias a su sencillez, rapidez y capacidad de reconocimiento. Su sistema es muy simple, ya que solo basta con que subas una imagen o URL a su sitio. Si la imagen que guardaste no tiene una resolución baja, podrás editarla rápidamente y obtener mejores resultados.

Reconoce hasta 460.000 tipografías diferentes. Además, dentro de su sitio encontrarás una comunidad de usuarios a la que podrás acudir cuando quieras.

- ✓ [Mockflow](#): La herramienta Mockflow es perfecta para construir esquemas de diseño de forma más rápida. Lo único que tienes que hacer es arrastrar y soltar para colocar los elementos con iconos preconfigurados. Te apoyará en la creación de una página web funcional y en la planificación de interfaces de usuario.

Cuenta con una tienda de plantillas para que encuentres inspiración por medio del trabajo de otros diseñadores. Puedes utilizarla también desde la aplicación móvil.

- ✓ [Tinypng](#): Tinypng es la mejor forma de optimizar tus imágenes: su función es esencial pues comprime tus contenidos gráficos. Esto es necesario para que, al momento de navegar en una página web, las imágenes no hagan más lento el tiempo de carga.

Su uso es fácil y optimiza las imágenes de forma automática.

- ✓ [Notepad++](#): Si prefieres diseñar sin un sistema de gestión de contenidos o CMS, entonces esta se convertirá en tu herramienta de cabecera. Con Notepad++ podrás despedirte de tu bloc de notas, ya que en esta aplicación editarás los códigos fuente. Identifica con su ayuda las expresiones de lenguaje utilizado y resalta las combinaciones de colores para facilitar la lectura. Optimiza las rutinas y mantiene un uso sencillo. Por supuesto, también tenemos a [Sublime Text](#), entre otros.



Freepik. Maquetación web (CC0)

- ✓ **Pixabay:** Pixabay es un banco de imágenes que puedes utilizar de forma gratuita. Actualmente cuenta con más de 900.000 archivos gratuitos que van desde fotografías artísticas hasta comerciales. ¡No te quedes sin las imágenes que atraerán a los usuarios! Copia, modifica y distribuye las imágenes que prefieras. También puedes descargar videos de varias categorías y en diferentes resoluciones. Otro ejemplo en esta categoría es [Freepik](#). Si lo que quieres es descargar iconos, [Iconfinder](#) es perfecto.
- ✓ **Ceros:** Ceros es de las más versátiles herramientas de diseño web, ideada especialmente para crear diferentes piezas de contenidos. Así que en este potente software podrás fácilmente diseñar diferentes formatos interactivos, tales como infografías, ebooks, banners, microsites, revistas, y mucho más.
- ✓ **Canva:** Sus funciones son más sencillas en comparación a otras herramientas de diseño web, pero sin duda, igual de efectivas. Canva es la plataforma estrella en la creación de imágenes, cabeceras para blogs y páginas web. Considerando que es lo primero que verán tus usuarios, se ha vuelto una herramienta imprescindible a la hora de armonizar el diseño de tu web. Además, también se ha vuelto una plataforma aliada a la hora de crear publicaciones para las redes sociales. Y el plus de esta herramienta es que no tienes que ser un experto en diseño gráfico, su interfaz es muy sencilla, gratuita y online.

Autoevaluación

Todas las herramientas web permiten hacer animaciones y retocar fotografías:

- Verdadero.
- Falso.

Incorrecto.

Correcto.

Solución

1. Incorrecto
2. Opción correcta

6.- Hojas de estilo o CSS.

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.

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 diciembre de 1995 fue la primera recomendación oficial.

CSS 2, publicada en 1998 fue la segunda recomendación oficial. Hubo una actualización CSS 2.1 el 7 de junio de 2011.

CSS 3, A diferencia de CSS 2, la tercera versión no es una gran especificación que integra todas las funcionalidades sino que está dividida en varios módulos separados. Cada módulo añade características de manera que se va actualizando por partes. Los trabajos en CSS 3 comenzaron a la vez que se publicó la recomendación oficial de CSS 2, y los primeros borradores de CSS 3 fueron liberados en junio de 1999.

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.



Cristóbal Alvarado. Hojas de estilo en cascada (CC BY)

Para saber más

A continuación algunos enlaces para ampliar los conocimientos que se verán en este apartado:

- ✓ [Especificaciones CSS 3](#)
- ✓ [Especificación CSS 2.1](#)
- ✓ [Introducción a CSS](#)
- ✓ [CSS avanzado](#)

Autoevaluación

Las hojas de estilo en cascada permiten:

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

Correcto.

Incorrecto.

No es correcto.

No es la opción correcta.

Solución

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

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 desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Opera	Presto	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Firefox	Gecko	Completo desde la versión 1.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Google Chrome	WebKit	Completo desde la versión 85 del motor	Completo	Todos los selectores, pseudo-clases y muchas propiedades
Internet Explorer	Trident	Completo desde la versión 7.0 del navegador	Completo	Todos los selectores, pseudo-clases y muchas propiedades a partir de la versión 10.0 del navegador



[gordiano](#) Navegando (Todos los derechos reservados)

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 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.

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

Solución

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

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 en la cabecera de la página web.

El elemento 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:

El archivo formatos.css contiene:

```
h3 { color: green; }
p { color: orange; font-family: Verdana; }
```

[Código fuente de la imagen](#)

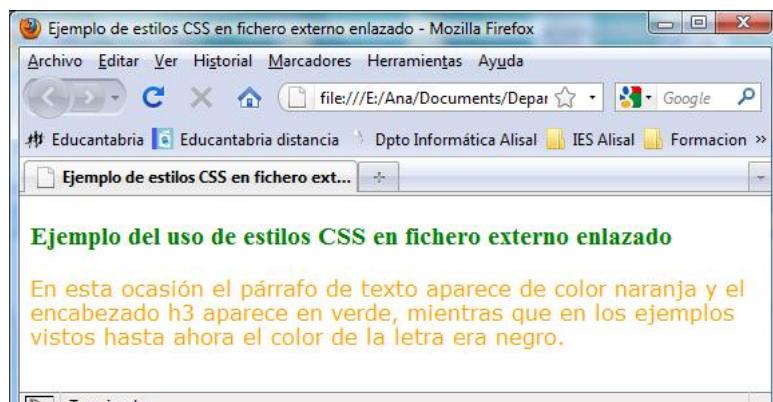
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

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

    <body>
        <h3>Ejemplo del uso de estilos CSS en fichero externo enlazado</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>
```

[Código fuente de la imagen](#)

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



Ana. Previsualización del código HTML (Todos los derechos reservados)

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() contenido 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:

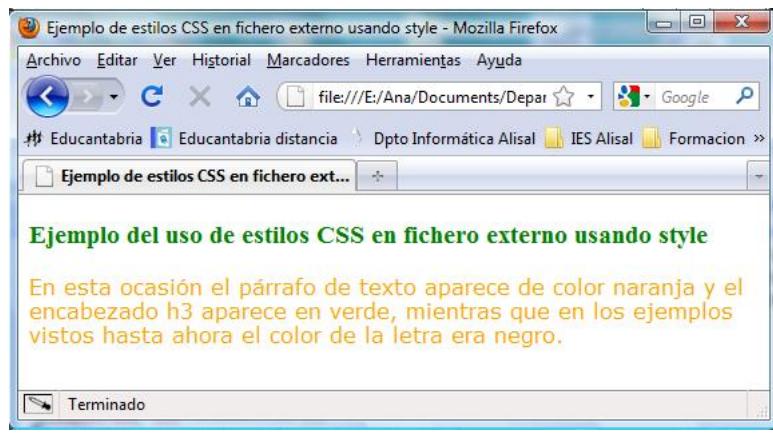
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

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

    <body>
        <h3>Ejemplo del uso de estilos CSS en fichero externo usando style</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>
```

[Código fuente de la imagen](#)

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



Ana. Previsualización del código HTML (Todos los derechos reservados)

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

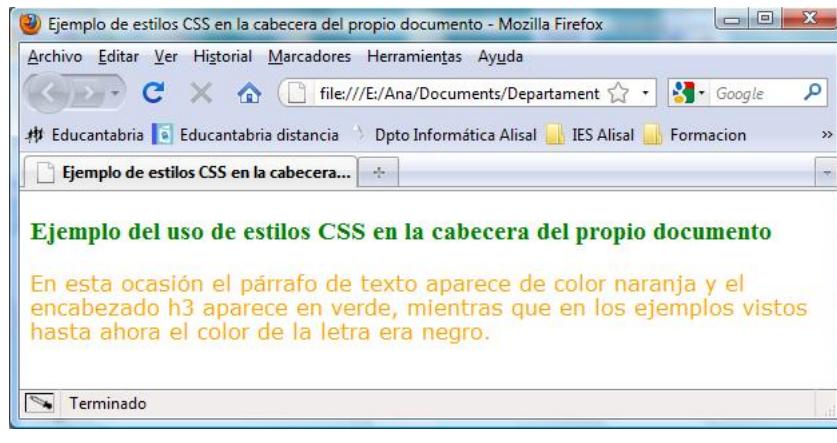
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de 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>

    <body>
        <h3>Ejemplo del uso de estilos CSS en la cabecera del propio 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>
```

[Código fuente de la imagen](#)

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



6.2.4.- Incluir CSS en los elementos HTML.

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

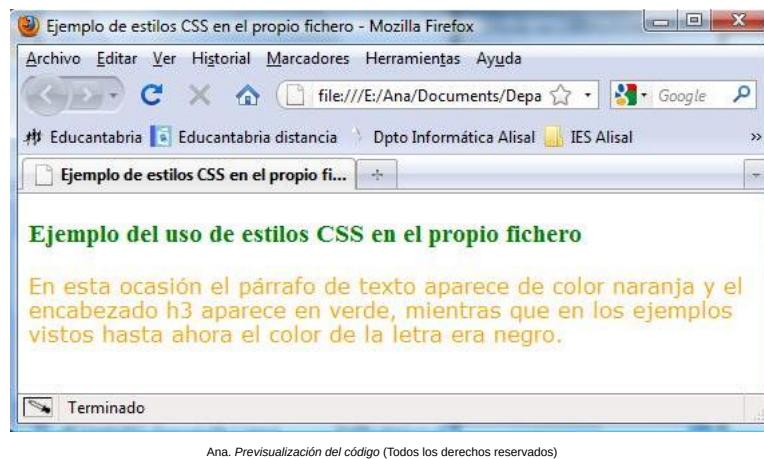
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
    <title>Ejemplo de estilos CSS en el propio fichero</title>
</head>

<body>
    <h3 style="color: green;">Ejemplo del uso de estilos CSS en el propio fichero</h3>
    <p style="color: orange; font-family: Verdana;">En esta ocasión el encabezado h3 aparece en verde, mientras que en los ejemplos vistos hasta ahora el color de la letra era negro.</p>
</body>
</html>
```

[Código fuente de la imagen](#)

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



Ana. Previsualización del código (Todos los derechos reservados)

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.

+ X + = +
- X - = +
+ X - = -
- X + = -
+ : + = +
- : - = +
+ : - = -
- : + = -

Víctor Gómez. Sintaxis ([CCO](#))

Autoevaluación

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

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

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

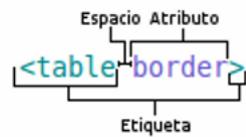
Solución

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

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.



[Psybeat. Atributos y elementos \(CC BY-NC\)](#)

Pasamos a verlos detenidamente en los siguientes apartados.

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

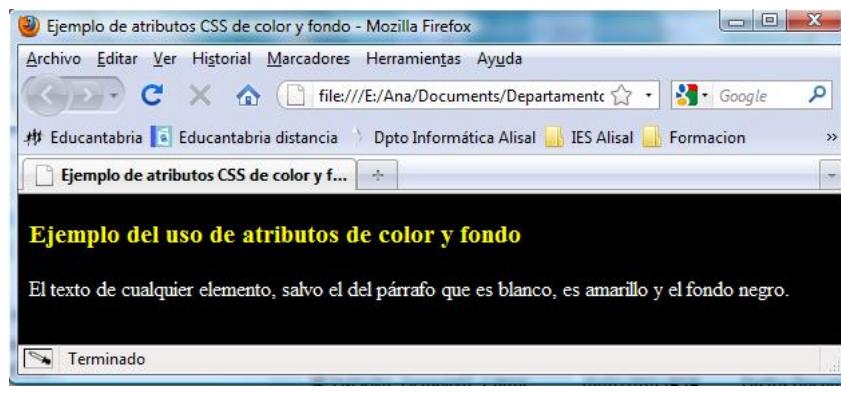
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

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



Ana. Previsualización del código HTML (Todos los derechos reservados)

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
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:

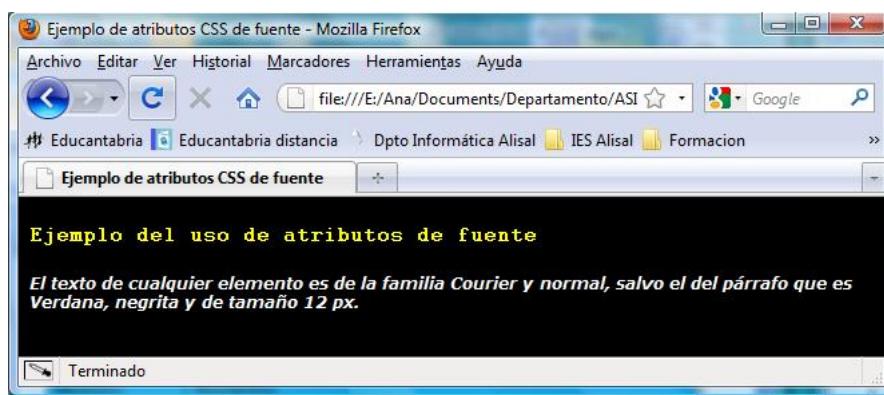
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<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:



Ana. Previsualización del código HTML (Todos los derechos reservados)

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 y por tanto, el espaciado entre líneas. 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 esta tabulado</p>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



6.4.4.- Atributos de caja.

Para comprender los atributos de caja puedes visualizar la siguiente presentación del modelo de cajas en CSS, que puedes descargar en formato ODP [aquí](#) (195417 B) o en formato PDF [aquí](#) (195417 B)

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:

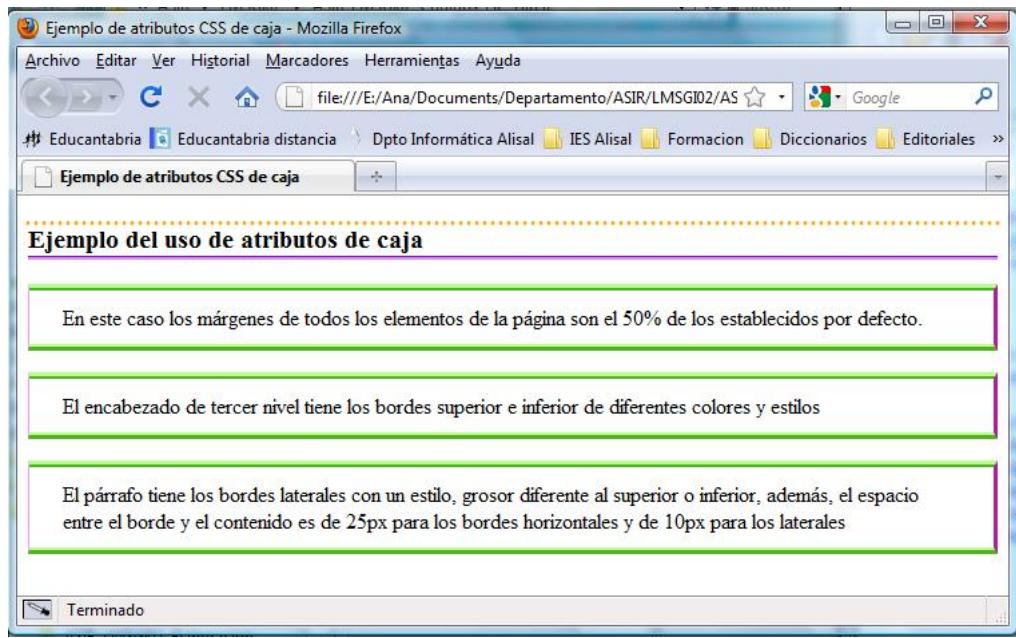
Elemento	Descripción
margin-left	Indica el tamaño del margen izquierdo. Puede usarse una longitud, en unidades CSS, o un porcentaje.
margin-right	Indica el tamaño del margen derecho. Puede usarse una longitud, en unidades CSS, o un porcentaje.
margin-top	Indica el tamaño del margen superior. Puede usarse 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?



Ana. Ejemplos de uso de atributos de caja (Todos los derechos reservados)

[Mostrar retroalimentación](#)

```
<!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>
```

[Código fuente de la imagen](#)

Autoevaluación

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

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

Incorrecto.

No es la opción correcta.

Correcto.

No es correcto.

Solución

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

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

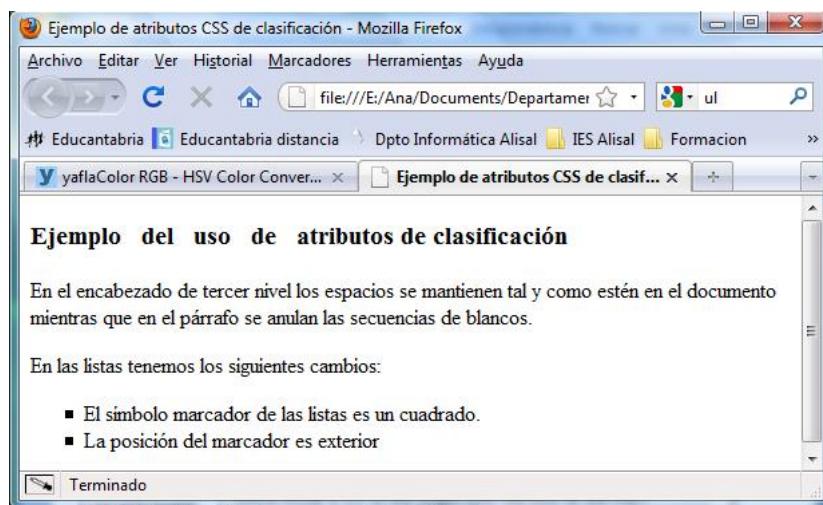
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <head>
        <title>Ejemplo de atributos CSS de clasificación</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 clasificación</h3>
        <p>En el encabezado de tercer nivel los espacios se mantienen tal y como estén en el documento mientras que en el párrafo se anulan las secuencias de blancos.</p>
        <p>En las listas tenemos los siguientes cambios:</p>
        <ul>
            <li>El símbolo marcador de las listas es un cuadrado.</li>
            <li>La posición del marcador es exterior</li>
        </ul>
    </body>
</html>
```

[Código fuente de la imagen](#)

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



6.5.- CSS de posicionamiento y visualización.

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.
display	Modifica la forma en la que se visualiza un elemento.
visibility	Indica si el elemento sobre el que actúa será visible o no. Los valores que puede tomar son: hidden y collapse.
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:

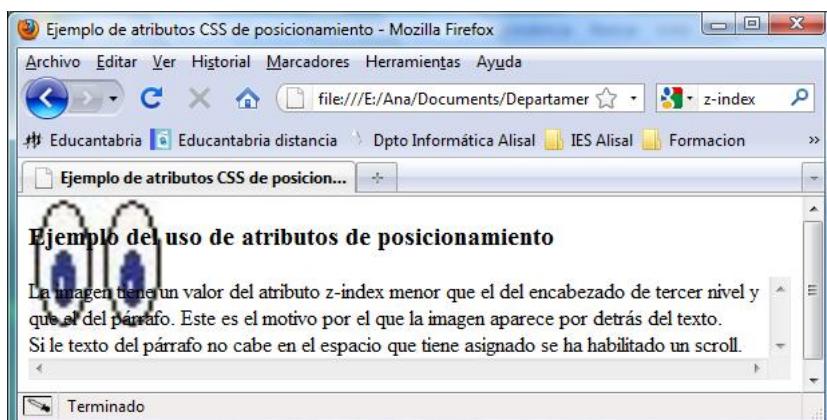
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <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 párrafo. Este es el motivo por el que la imagen aparece por detrás del texto. <br><br/>
            Si le texto del párrafo no cabe en el espacio que tiene asignado se ha habilitado un scroll.</p>
    </body>
</html>
```

[Código fuente de la imagen](#)

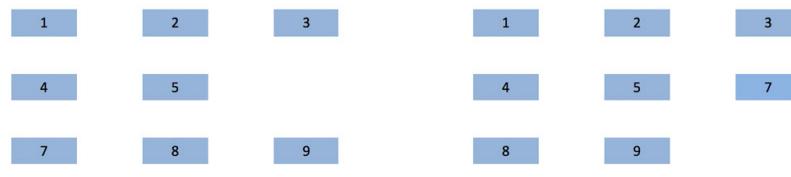
Al publicarlo en un navegador, por ejemplo en el Firefox, tendríamos:



Las propiedades **display** y **visibility** controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página. Habitualmente se utilizan junto con JavaScript para crear efectos dinámicos como mostrar y ocultar determinados textos o imágenes cuando el usuario pincha sobre ellos.

La diferencia entre ambas propiedades es que, mientras **display** oculta por completo un elemento haciendo que los demás elementos ocupen su lugar, la propiedad **visibility** hace que los otros elementos que componen la página respeten la posición de éste y reserven su espacio.

Las siguientes imágenes muestran la diferencia entre ocultar el rectángulo número 6 mediante la propiedad **display** o hacerla invisible mediante la propiedad **visibility**: En la Imagen 1, el rectángulo 6 tiene la propiedad **visibility: hidden** y en la Imagen 2, el rectángulo 6 tiene la propiedad **display: none**



Victor Gómez. *Visibility y Display* ([CC0](#))

La propiedad **display** es mucho más utilizada que **visibility**, pues normalmente no desea reservarse el espacio de contenido oculto.

- ✓ Propiedad **display**: Las posibilidades de la propiedad **display** son mucho más avanzadas que simplemente ocultar elementos. En realidad, la propiedad **display** modifica la forma en la que se visualiza un elemento.

Elemento	Descripción
<code>display: block</code>	Muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate. Se emplea frecuentemente para los enlaces que forman el menú de navegación.
<code>display: inline</code>	Visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate. Se puede utilizar en las listas que se quieren mostrar horizontalmente. (,)
<code>display: none</code>	Oculta un elemento y hace que desaparezca de la página

- ✓ Propiedad **visibility**: **Visibility** es una propiedad mucho más sencilla que **display**, pues sus posibilidades son mucho más limitadas y únicamente permite hacer visibles o invisibles los elementos de una página.

Elemento	Descripción
<code>visibility: hidden</code>	Convierte una caja en invisible para que no muestre sus contenidos. El resto de elementos de la página se muestran como si la caja todavía fuera visible, por lo que en el lugar donde originalmente se mostraba la caja invisible, ahora se muestra un hueco vacío.
<code>visibility: collapse</code>	Sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla. Su efecto es similar al de la propiedad display , ya que oculta completamente la fila y/o columna y se pueden mostrar otros contenidos en ese lugar. Si se utiliza el valor <code>collapse</code> sobre cualquier otro tipo de elemento, su efecto es idéntico al valor <code>hidden</code> .

6.6.- Unidades de medida.

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

- ✓ **Absolutas:** Una medida indicada mediante unidades absolutas está completamente definida, ya que su valor no depende de otro valor de referencia. A continuación se muestra la lista completa de unidades absolutas definidas por CSS y su significado. La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. Su principal desventaja es que son muy poco flexibles y no se adaptan fácilmente a los diferentes medios.

- ◆ *Milímetros (mm).*



JaulaDeArdilla, Flexómetro (CC BY-NC-ND)

```
/* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre si */  
p { word-spacing: 4mm; }
```

- ◆ *Centímetros (cm):* Cada centímetro son 10 mm.

```
/* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */  
h1 { line-height: 2cm; }
```

- ◆ *Pulgadas (in):* Cada pulgada equivale a 2,54 cm.

```
/* El cuerpo de la página debe mostrar un margen de media pulgada */  
body { margin: 0.5in; }
```

- ◆ *Puntos (pt):* Cada punto son 1/72 in.

```
/* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */  
a { font-size: 12pt; }
```

- ◆ *Picas (pc):* Cada pica son 12 pt.

```
/* Los elementos <span> deben tener un tamaño de letra de 1 pica */  
span { font-size: 1pc; }
```

- ✓ **Relativas:** La unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las mas utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios. A continuación se muestran las tres unidades de medida relativas definidas por CSS y la referencia que toma cada una para determinar su valor real.

Las unidades em y ex no han sido creadas por CSS, sino que llevan décadas utilizándose en el campo de la tipografía. Aunque no es una definición exacta, la unidad 1em equivale a la anchura de la letra M ("eme mayúscula") del tipo y tamaño de letra del elemento. La unidad em hace referencia al tamaño en puntos de la letra que se está utilizando. Si se utiliza una tipografía de 12 puntos, 1em equivale a 12 puntos. El valor de 1ex se puede aproximar por 0.5 em.

- ◆ *Element (em):* Expresa el tamaño relativo al tamaño de la fuente utilizada. (no confundir con la etiqueta de HTML)
- ◆ *X-height (ex):* Expresa el tamaño relativo al de la letra "x" del tipo y tamaño de letra del elemento.
- ◆ *Pixel (px):* Expresa el tamaño relativo respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

- ✓ **Porcentajes:** El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a em, ex y px. Un porcentaje está formado por un valor numérico seguido del símbolo % y siempre está referenciado a otra medida. Cada una de las propiedades de CSS que permiten indicar como valor un porcentaje, define el valor al que hace referencia ese porcentaje. Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de letra de los elementos:

```
body { font-size: 1em; }  
h1 { font-size: 200%; }  
h2 { font-size: 150%; }
```

Los tamaños establecidos para los elementos <h1> y <h2> mediante las reglas anteriores, son equivalentes a 2em y 1.5em respectivamente, por lo que es más habitual definirlos mediante em.

Los porcentajes también se utilizan para establecer la anchura de los elementos:

```
div#contenido { width: 600px; }  
div.principal { width: 80%; }  
  
<div id="contenido">  
  <div class="principal">  
    ...
```

```
</div>  
</div>
```

En el ejemplo anterior, la referencia del valor 80% es la anchura de su elemento padre. Por tanto, el elemento `<div>` cuyo atributo class vale principal tiene una anchura de $80\% \times 600\text{px} = 480\text{px}$.

Caso práctico

Si se considera el siguiente ejemplo:

```
p { margin: 1em; }
```

La regla CSS anterior indica que los párrafos deben mostrar un margen de anchura igual a 1em. Como se trata de una unidad de medida relativa, es necesario realizar un cálculo matemático para determinar la anchura real de ese margen.

La unidad de medida em siempre hace referencia al tamaño de letra del elemento. Por otra parte, todos los navegadores muestran por defecto el texto de los párrafos con un tamaño de letra de 16 píxel. Por tanto, en este caso el margen de 1em equivale a un margen de anchura 16px.

A continuación se modifica el ejemplo anterior para cambiar el tamaño de letra de los párrafos:

```
p { font-size: 32px; margin: 1em; }
```

El valor del margen sigue siendo el mismo en unidades relativas (1em) pero su valor real ha variado porque el tamaño de letra de los párrafos ha variado. En este caso, el margen tendrá una anchura de 32px, ya que 1em siempre equivale al tamaño de letra del elemento.

Si se quiere reducir la anchura del margen a 16px pero manteniendo el tamaño de letra de los párrafos en 32px, se debe utilizar la siguiente regla CSS:

```
p { font-size: 32px; margin: 0.5em; }
```

El valor 0.5em se interpreta como "*la mitad del tamaño de letra del elemento*", ya que se debe multiplicar por 0.5 su tamaño de letra ($32\text{px} \times 0.5 = 16\text{px}$). De la misma forma, si se quiere mostrar un margen de 8px de anchura, se debería utilizar el valor 0.25em, ya que $32\text{px} \times 0.25 = 8\text{px}$.

La gran ventaja de las unidades relativas es que siempre mantienen las proporciones del diseño de la página. Establecer el margen de un elemento con el valor 1em equivale a indicar que "*el margen del elemento debe ser del mismo tamaño que su letra y debe cambiar proporcionalmente*".

En efecto, si el tamaño de letra de un elemento aumenta hasta un valor enorme, su margen de 1em también será enorme. Si su tamaño de letra se reduce hasta un valor diminuto, el margen de 1em también será diminuto. El uso de unidades relativas permite mantener las proporciones del diseño cuando se modifica el tamaño de letra de la página.

El funcionamiento de la unidad ex es idéntico a em, salvo que en este caso, la referencia es la altura de la letra x minúscula, por lo que su valor es aproximadamente la mitad que el de la unidad em.

Por último, las medidas indicadas en píxel también se consideran relativas, ya que el aspecto de los elementos dependerá de la resolución del dispositivo en el que se visualiza la página HTML. Si un elemento tiene una anchura de 400px, ocupará la mitad de una pantalla con una resolución de 800x600, pero ocupará menos de la tercera parte en una pantalla con resolución de 1440x900.

Las unidades de medida se pueden mezclar en los diferentes elementos de una misma página, como en el siguiente ejemplo:

```
body { font-size: 10px; }  
h1 { font-size: 2.5em; }
```

En primer lugar, se establece un tamaño de letra base de 10 píxel para toda la página. A continuación, se asigna un tamaño de 2.5em al elemento `<h1>`, por lo que su tamaño de letra real será de $2.5 \times 10\text{px} = 25\text{px}$.

El valor de la mayoría de propiedades CSS se hereda de padres a hijos. Así por ejemplo, si se establece el tamaño de letra al elemento `<body>`, todos los elementos de la página tendrán el mismo tamaño de letra, salvo que indiquen otro valor. Sin embargo, el valor de las medidas relativas no se hereda directamente, sino que se hereda su valor real una vez calculado. El siguiente ejemplo muestra este comportamiento:

```
body {  
    font-size: 12px;  
    text-indent: 3em;  
}  
h1 { font-size: 15px }
```

Autoevaluación

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

- Verdadero.
- Falso.

Incorrecto.

Correcto.

Solución

1. Incorrecto
2. Opción correcta

6.7.- Definición y uso de clases "class".

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:

El encabezado de tercer nivel es ahora azul

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:

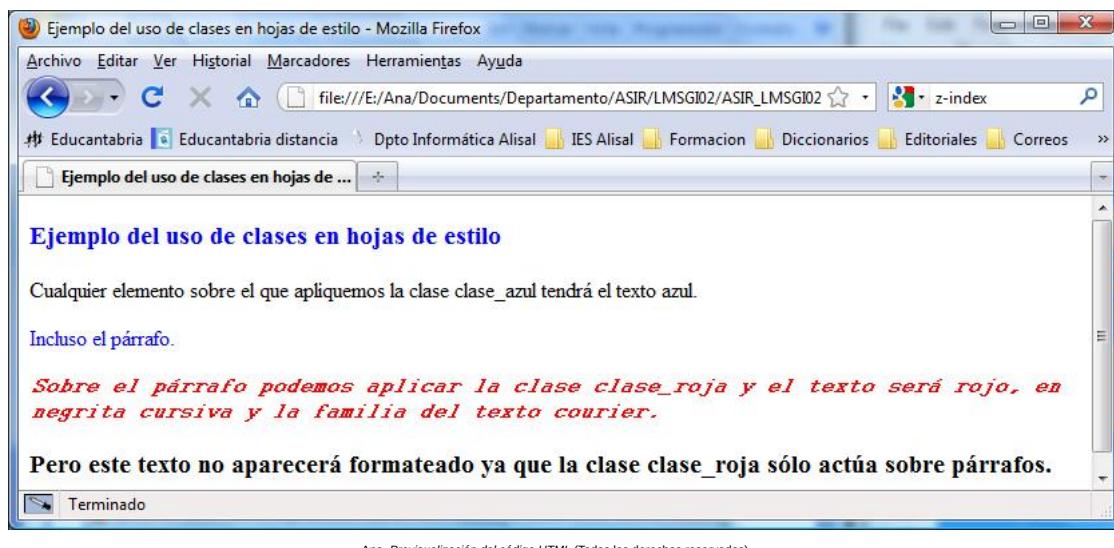
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

    <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:bold; 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 p&aacute;rrafo.</p>
        <p class="clase_roja">Sobre el p&aacute;rrafo 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:



Autoevaluación

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

- rgb(33,33,0).
- red.

- #f06.
- #ff06.

No es la opción correcta.

No es correcto.

Incorrecto.

Correcto.

Solución

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

6.8.- Definición y uso de identificadores "id".

Además de agrupar elementos, podrías necesitar identificar un elemento único. Esto se hace con el atributo `id`.

Lo que hace especial al atributo `id` es que no pueden existir dos elementos dentro del mismo documento con el mismo `id`. Cada `id` tiene que ser único. En cualquier otro caso, se debería usar el atributo `class` en su lugar. Ahora, examinemos un ejemplo de un posible uso del atributo `id`:

```
<h1>Capítulo 1</h1>
...
<h2>Pregunta 1.1</h2>
...
<h2>Pregunta 1.2</h2>
...
<h1>Capítulo 2</h1>
...
<h2>Pregunta 2.1</h2>
...
<h3>Apartado 2.1.2</h3>
...
```

El código anterior podría hacer referencia a encabezados de cualquier documento dividido en capítulos y párrafos. Sería normal asignar un `id` a cada capítulo de la siguiente manera:

```
<h1 id="c1">Capítulo 1</h1>
...
<h2 id="c1-1">Pregunta 1.1</h2>
...
<h2 id="c1-2">Pregunta 1.2</h2>
...
<h1 id="c2">Capítulo 2</h1>
...
<h2 id="c2-1">Pregunta 2.1</h2>
...
<h3 id="c2-1-2">Apartado 2.1.2</h3>
...
```

Digamos que el título del capítulo 1.2 tiene que estar en rojo y la fuente ha de ser Courier New. Usando el código CSS necesario, se podría hacer así:

```
#c1-2 {
    color: red;
}
```

[Ejemplo de documento HTML con identificador.](#)

[Hoja de estilos CSS asociada al documento HTML.](#)

El resultado en el navegador sería el siguiente:

Capítulo 1

...

Pregunta 1.1

...

Pregunta 1.2

...

Capítulo 2

...

Pregunta 2.1

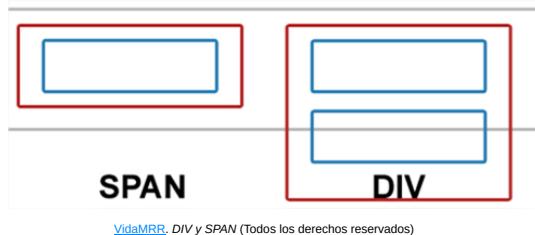
...

Apartado 2.1.2

Víctor Gómez. Previsualización del código HTML
[\(CC0\)](#)

Como se muestra en el ejemplo anterior, se pueden definir las propiedades de un elemento específico usando `#nombredelidentificador` en la hoja de estilo del documento.

6.9.- Agrupación de elementos "span" y "div".



VidaMRR. DIV y SPAN (Todos los derechos reservados)

Los elementos y <div> se usan para agrupar y estructurar un documento, también se usan junto con los atributos "class" e "id".

- ✓ "": El elemento es un elemento que no añade nada al documento en sí, pero con CSS se puede usar para añadir características visuales distintivas a partes específicas de texto en los documentos. Este elemento se usa dentro de un elemento a nivel de bloque. A continuación vemos un ejemplo:

```
<p>La etiqueta span sólo funciona como contenedor de una línea, es decir, no se puede ampliar para ocupar un párrafo.</p>
```

Queremos enfatizar en rojo que span solo funciona como contenedor de una línea. A cada elemento span se le añade el atributo style, de la siguiente forma:

```
<p>La etiqueta <span style="color: red;">span sólo funciona como contenedor de una línea</span>, es decir, no se puede ampliar
```

O bien podemos incluir un atributo class que podemos definir en nuestra hoja de estilo de la siguiente forma:

```
<p>La etiqueta <span class="funcionamiento">span sólo funciona como contenedor de una línea</span>, es decir, no se puede ampliar
```

El código CSS necesario para producir este efecto es el siguiente:

```
span.funcionamiento {  
    color:red;  
}
```

La etiqueta span sólo funciona como contenedor de una línea, es decir, no se puede ampliar para ocupar un párrafo.

Víctor Gómez. Resultado en navegador del ejemplo con SPAN (CC0)

- ✓ "<div>": El elemento <div> se usa para agrupar uno o más elementos a nivel de bloque, por ejemplo, se puede ampliar para ocupar un párrafo, ya que puede expandirse lo que sea necesario. Esta la principal razón por la que se usa para crear un sitio web, la estructura básica comienza con capas div. Ejemplo con dos listas de módulos de ciclos formativos de Informática, divididas según sea DAM o DAW.

```
<div id="dam">  
    <ul>  
        <li>Sistemas Informáticos</li>  
        <li>Bases de Datos</li>  
        <li>Programación</li>  
        <li>Entornos de Desarrollo</li>  
        <li>Acceso a Datos</li>  
        <li>Desarrollo de Interfaces</li>  
    </ul>  
</div>
```

```
<div id="daw">  
    <ul>  
        <li>Desarrollo web en entorno cliente</li>  
        <li>Desarrollo web en entorno servidor</li>  
        <li>Despliegue de aplicaciones web</li>  
        <li>Entornos de Desarrollo</li>  
        <li>Diseño de interfaces web</li>  
    </ul>  
</div>
```

```
<li>Lenguajes de marcas y sistemas de gestión de información</li>
</ul>
</div>
```

En nuestra hoja de estilo podemos utilizar la agrupación del mismo modo que antes:

```
#dam {
    background:LightSalmon;
}
#daw {
    background:LightCyan;
}
```

Módulos de ciclos formativos de Informática

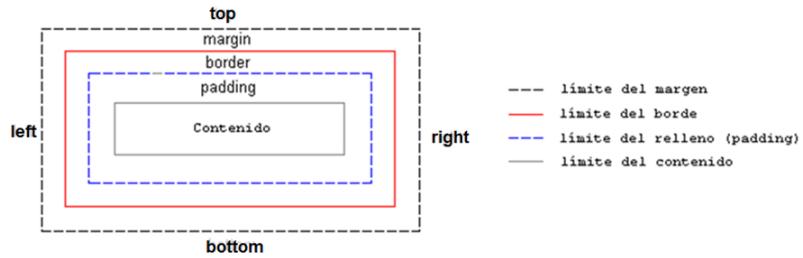
- Sistemas Informáticos
 - Bases de Datos
 - Programación
 - Entornos de Desarrollo
 - Acceso a Datos
 - Desarrollo de Interfaces
-
- Desarrollo web en entorno cliente
 - Desarrollo web en entorno servidor
 - Despliegue de aplicaciones web
 - Entornos de Desarrollo
 - Diseño de interfaces web
 - Lenguajes de marcas y sistemas de gestión de información

Víctor Gómez. *Resultado en navegador del ejemplo con DIV* ([CCO](#))

6.10.- El modelo de caja: margen y relleno.

El modelo de caja en CSS describe las cajas que se generan a partir de los elementos HTML y contiene opciones detalladas en lo referente al ajuste de márgenes, bordes, relleno (padding) y contenido de cada elemento. La imagen muestra cómo se construye el modelo de caja:

El modelo de caja en CSS



Manuel Rodríguez. *Modelo de caja en CSS* (Todos los derechos reservados)

Ejemplo de un encabezado y algo de texto. El código HTML de nuestro ejemplo es el siguiente:

```
<h1>Caja o capa</h1>
<p>Una caja determina el comportamiento de CSS y hace que todos los elementos de las páginas webs se representen mediante cajas rectangulares.</p>
```

Si añadimos algo de color e información sobre la fuente, el ejemplo se podría presentar así:

Caja o capa

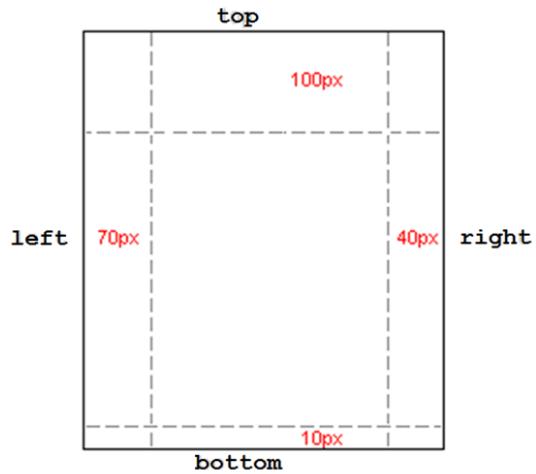
Una caja determina el comportamiento de CSS y hace que todos los elementos de las páginas webs se representen mediante cajas rectangulares.

El ejemplo contiene el elemento `<h1>` y el elemento `<p>`. La imagen muestra cómo cada elemento HTML está rodeado por cajas que se pueden ajustar usando CSS.



Manuel Rodríguez. *Cajas sobre elementos html* (Todos los derechos reservados)

- ✓ "`<margin>`" Establece el margen de un elemento. Todo elemento tiene cuatro lados: derecho, izquierdo, superior e inferior. La propiedad `margin` hace referencia a la distancia desde cada lado respecto al elemento colindante (o respecto a los bordes del documento). En el siguiente ejemplo definimos los márgenes del documento, es decir, del elemento `html`. La imagen nos muestra cómo queremos que sean los márgenes de nuestras páginas.



Manuel Rodríguez. Márgenes sobre un documento. (Todos los derechos reservados)

El código CSS necesario para esto es el siguiente:

```
body {
    margin-top: 100px;
    margin-right: 40px;
    margin-bottom: 10px;
    margin-left: 70px;
}
```

O bien usar la versión simplificada de `margin`, que queda como más elegante:

```
body {
    margin: 100px 40px 10px 70px;
}
```

También se puede establecer los márgenes de otros elementos. Por ejemplo, definir márgenes para todos los párrafos de texto marcados con `<p>`:

```
body {
    margin: 100px 40px 10px 70px;
}
<br />p {
    margin: 5px 50px 5px 50px;
}
```

- ✓ "<padding>": Establece el relleno de un elemento. Esta propiedad se entiende como "relleno". El relleno (padding) no afecta a la distancia de un elemento respecto a otros elementos, sino que sólo define la distancia interior entre el borde y el contenido del elemento. Un ejemplo del uso de padding es que todos los títulos tengan diferentes colores de fondo y cambiamos la cantidad de "relleno" que habrá alrededor del texto en cada uno de ellos:

```
h1 {
    background: yellow;
    padding: 20px 20px 20px 80px;
}
h2 {
    background: orange;
    padding-left: 120px;
}
```

Encabezados y relleno

Un primer párrafo.

Primer encabezado de tipo H2

Un segundo párrafo.

Segundo encabezado de tipo H2

Un tercer párrafo.

Víctor Gómez. Resultado en navegador del ejemplo con [\(CC0\)](#)

Caso práctico

Sitio web utilizando capas

En el siguiente ejemplo se muestran capas para incluir imágenes en una galería, así como su hoja de estilo asociada. Pulsa [aquí](#) para descargarlo.

Anexo I.- Videos explicativos y recursos

- ✓ Tutoriales para aprender a diseñar páginas y sitios web utilizando los lenguajes XHTML y CSS:
<https://www.youtube.com/playlist?list=PLBE0D44EEB18C6C1B>
- ✓ CSS desde 0 al infinito. CSS3:
<https://www.youtube.com/playlist?list=PLU8oAlHdN5BmpUDdnWSgllHflosElaVN>
- ✓ Curso del estándar HTML 5:
<https://www.youtube.com/playlist?list=PLU8oAlHdN5BnX63lyAeV0LzLnpGudgRrK>
- ✓ Videotutorial de CSS, en el que aprenderás los aspectos más importantes de las Hojas de Estilo en Cascada de una manera práctica y amena:
<http://www.desarrolloweb.com/manuales/videotutorial-css.html>
- ✓ Crear efecto hover en una imagen mediante CSS:
https://www.youtube.com/watch?feature=player_embedded&v=bWGOJcC88nU

Condiciones y términos de uso de los materiales

Materiales desarrollados inicialmente por el Ministerio de Educación, Cultura y Deporte y actualizados por el profesorado de la Junta de Andalucía bajo licencia Creative Commons BY-NC-SA.



Antes de cualquier uso leer detenidamente el siguiente [Aviso legal](#)

Historial de actualizaciones

Versión: 02.00.03	Fecha de actualización: 15/10/21	Autoría: Víctor Javier Gómez Arques
Actualización de materiales y correcciones menores.		
Versión: 02.00.00	Fecha de actualización: 20/06/20	Autoría: Víctor Javier Gómez Arques
<p>Ubicación: Puntos 2, 3.1, 3.2.2, 4.1, 6.6, 6.9 y 6.10 / Puntos 1.1, 3.2.8 y 5 / Puntos 6.1 y Anexo</p> <p>Mejora (tipo 1): Se añaden ejemplos con código explicado por pasos / Se desarrolla el apartado de una forma mas exhaustiva / Se actualiza a 2020</p> <p>Ubicación: Unidad 2 Utilización de Lenguajes de Marcas en Entornos Web</p> <p>Mejora (tipo 1): Sección 7, sección 7.1, sección 7.2, sección 7.3, sección 7.4, sección 7.5, sección 7.6, sección 7.7</p> <p>Ubicación: Sección 7: de la 7.1 a la 7.7</p> <p>Mejora (tipo 2): Añadir contenido referente a HTML5.</p> <p>No hay nada referente a la última evolución de HTML en aspectos importantes como:</p> <ul style="list-style-type: none">*Uso de versión y cabeceras más simples.*Estructura básica de un documento HTML5.*Integración de etiquetas semánticas.*Ejemplos en HTML5.*Descripción de las etiquetas semánticas.*Inserción de audio y vídeo en HTML5.*HTML5: nuevos elementos y atributos de formularios.*Como migrar de HTML4 a HTML5. <p>Ubicación: Todo el tema</p> <p>Mejora (tipo 3): No hay ejemplos para ir captando lo que se dice en el temario y tampoco hay vídeos explicando el proceso de resolución de ejercicios. Insisto en la necesidad de videoclases en todos los módulos prácticos. Se monta el vídeo una vez y te sirve para varios años.</p> <p>Me esperaba la formación a distancia bastante mejor en el aspecto práctico, a penas hay material práctico y vídeos propios resolviendo ejercicios, sobretodo esto último.</p> <p>Ubicación: Se actualiza con HTML5</p> <p>Mejora (Mapa conceptual): Se actualiza con HTML5</p> <p>Ubicación: Punto 6.6</p> <p>Mejora (Orientaciones del alumnado): Se actualiza con el término medida</p>		
Versión: 01.02.00	Fecha de actualización: 01/07/16	Autoría: Víctor Javier Gómez Arques
<p>Ubicación: Toda la Unidad</p> <p>Mejora (tipo 1): Incorporadas las licencias a todas las imágenes según el procedimiento de actualización de materiales.</p> <p>Ubicación: Sección 6.8.- Identificadores, Sección 6.9.- Elementos span y div y sección 6.10.- Uso de cajas</p> <p>Mejora (tipo 2): Incorporados 3 nuevos apartados fundamentales para el desarrollo de hojas XHTML con CSS.</p> <p>Ubicación: Sección 6.8.- Identificadores, Sección 6.9.- Elementos span y div y sección 6.10.- Uso de cajas</p> <p>Mejora (tipo 2): Se incorporan las secciones.</p> <ul style="list-style-type: none">6.8.- Definición y uso de identificadores id.6.9.- Agrupación de elementos span y div.6.10.- El modelo de caja: margen y relleno. <p>Ubicación: No especificada.</p> <p>Mejora (tipo 2): Falta añadir más ejemplos prácticos resueltos con hojas de estilos, y usar las capas o elementos div para crear la estructura de una página web, en lugar de los marcos o frameset que llevan desfasados muchos años.</p> <p>Ubicación: No hay ubicación en el mapa conceptual :)</p> <p>Mejora (Mapa conceptual): Desde la DGFP comentan:</p> <p>Mapa conceptual 2 en el título hace referencia a un ciclo (DAM) En el resto de módulos que comparten le han quitado la referencia al ciclo.</p>		
Versión: 01.01.00	Fecha de actualización: 28/10/14	Autoría: Víctor Javier Gómez Arques
<ul style="list-style-type: none">- Contenidos de HTML5, Capas y Propiedades de visibilidad CSS.- Actualización e inclusión de nuevos elementos HTML.- Videos explicativos y recursos de interoperabilidad web.- Corrección de erratas menores, código html y enlaces rotos.		

Versión: 01.00.00

Fecha de actualización: 28/04/14

Versión inicial de los materiales.

