



UF1841: Elaboración de documentos web mediante lenguajes de marcas

Certificado de Profesionalidad

IFCD0210 - Desarrollo de aplicaciones con tecnologías web



http://www.

IFCD0210 > MF0491_3 > UF1841

**Elaboración de documentos web mediante lenguajes
de marcas.
IFCD0210**

Ramón Guerrero Pérez

ic editorial

Elaboración de documentos web mediante lenguajes de marcas. IFCD0210

Autor: Ramón Guerrero Pérez

1^a Edición

© IC Editorial, 2014

Editado por: IC Editorial

C.I.F.: B-92.041.839

c/ Cueva de Viera, 2, Local 3 Centro Negocios CADI

29200 ANTEQUERA, Málaga

Teléfono: 952 70 60 04

Fax: 952 84 55 03

Correo electrónico: iceditorial@iceditorial.com

Internet: www.iceditorial.com

IC Editorial ha puesto el máximo esfuerzo en ofrecer una información completa y precisa. Sin embargo, no asume ninguna responsabilidad derivada de su uso, ni tampoco la violación de patentes ni otros derechos de terceras partes que pudieran ocurrir. Mediante esta publicación se pretende proporcionar unos conocimientos precisos y acreditados sobre el tema tratado. Su venta no supone para **IC Editorial** ninguna forma de asistencia legal, administrativa ni de ningún otro tipo.

Reservados todos los derechos de publicación en cualquier idioma.

Según el Código Penal vigente ninguna parte de este o cualquier otro libro puede ser reproducida, grabada en alguno de los sistemas de almacenamiento existentes o transmitida por cualquier procedimiento, ya sea electrónico, mecánico, reprográfico, magnético o cualquier otro, sin autorización previa y por escrito de IC EDITORIAL; su contenido está protegido por la Ley vigente que establece penas de prisión y/o multas a quienes intencionadamente reprodujeren o plagiaren, en todo o en parte, una obra literaria, artística o científica.

ISBN: 978-84-16433-03-2

Nota de la editorial: IC Editorial pertenece a Innovación y Cualificación S. L.

Presentación del manual

El **Certificado de Profesionalidad** es el instrumento de acreditación, en el ámbito de la Administración laboral, de las cualificaciones profesionales del Catálogo Nacional de Cualificaciones Profesionales adquiridas a través de procesos formativos o del proceso de reconocimiento de la experiencia laboral y de vías no formales de formación.

El elemento mínimo acreditable es la **Unidad de Competencia**. La suma de las acreditaciones de las unidades de competencia conforma la acreditación de la competencia general.

Una **Unidad de Competencia** se define como una agrupación de tareas productivas específica que realiza el profesional. Las diferentes unidades de competencia de un certificado de profesionalidad conforman la **Competencia General**, definiendo el conjunto de conocimientos y capacidades que permiten el ejercicio de una actividad profesional determinada.

Cada **Unidad de Competencia** lleva asociado un **Módulo Formativo**, donde se describe la formación necesaria para adquirir esa **Unidad de Competencia**, pudiendo dividirse en **Unidades Formativas**.

El presente manual desarrolla la Unidad Formativa **UF1841: Elaboración de documentos web mediante lenguajes de marcas**,

perteneciente al Módulo Formativo **MF0491_3: Programación web en el entorno cliente**,

asociado a la unidad de competencia **UC0491_3: Desarrollar elementos de software en el entorno cliente**,

del Certificado de Profesionalidad **Desarrollo de aplicaciones con tecnologías web**.

Índice

Portada

Título

Copyright

Presentación del manual

Índice

Capítulo 1 Diseño web

1. Introducción
2. Principios de diseño web
3. El proceso de diseño web
4. Diferencias entre diseño orientado a presentación y a impresión
5. Resumen
Ejercicios de repaso y autoevaluación

Capítulo 2 Lenguajes de marcado generales

1. Introducción
2. Origen de los lenguajes de marcado generales: SGML y XML
3. Características generales de los lenguajes de marcado
4. Estructura general de un documento con lenguaje de marcado
5. Documentos válidos y bien formados. Esquemas
6. Resumen
Ejercicios de repaso y autoevaluación

Capítulo 3 Lenguajes de marcado para presentación de páginas web

1. Introducción
2. Historia de HTML y XHTML. Diferencias entre versiones
3. Estructura de un documento
4. Color
5. Texto
6. Enlaces de hipertexto
7. Imágenes

8. Listas
 9. Tablas
 10. Marcos (*frames*)
 11. Formularios
 12. Elementos específicos para tecnologías móviles
 13. Elementos en desuso (deprecated)
 14. Resumen
- Ejercicios de repaso y autoevaluación

Capítulo 4 Hojas de estilo web

1. Introducción
 2. Elementos y estructura de una hoja de estilos
 3. Diseño de estilos para diferentes dispositivos
 4. Tipos de hojas de estilos: estáticas y dinámicas. Las pseudo-clases
 5. Buenas prácticas en el uso de hojas de estilo
 6. Resumen
- Ejercicios de repaso y autoevaluación

Bibliografía

Capítulo 1

Diseño web

1. Introducción

El diseño web se puede definir como aquella tarea destinada a la planificación, el diseño y la implementación de páginas y sitios web. Para diseñar adecuadamente una página web, es muy importante tener en cuenta aspectos tales como:

- Navegabilidad.
- Interactividad.
- Usabilidad de la página.
- Arquitectura y distribución de la información presentada.
- Integración de recursos tales como imágenes, audio, vídeo, etc.

El diseño y desarrollo web profesional es un tarea que implica el aprendizaje de ciertos lenguajes y técnicas de programación, fundamentalmente HTML y CSS. No obstante, también es muy importante distribuir y estructurar adecuadamente la información, además de presentar los contenidos de manera que el conjunto sea lo más atractivo e intuitivo posible para el usuario.

2. Principios de diseño web

Actualmente, miles de millones de páginas web se encuentran alojadas en Internet y su número va creciendo día a día. De todos estos sitios web, solo una pequeña parte están bien diseñados, ya que la mayoría de desarrolladores no tienen en cuenta todos o la mayor parte de los principios básicos que rigen el diseño web.



Internet está constituido por millones de ordenadores conectados en una compleja estructura de redes.



Importante

El diseño web es una técnica basada en un conjunto de reglas más o menos definidas que sirven para crear páginas que sean útiles y visualmente atractivas.

Además de diseñar gráficamente un sitio web, también es necesario planificar adecuadamente los contenidos de la página, así como estructurar cada uno de los menús, definir la manera en la que el usuario va a interactuar con cada uno de los elementos, etc. En definitiva, este tipo de diseño va mucho más allá de darle un aspecto y colorido a una página web.

El diseño web se puede dividir en dos partes fundamentales: la **funcionalidad** o usabilidad y el **aspecto** o estética. Cuando se diseña una página web, es indispensable tener en cuenta ambos factores, ya que una página puede funcionar muy bien, pero si no es visualmente atractiva no llamará la atención de posibles usuarios, mientras que aquellas páginas que se vean muy bien pero que funcionen mal, harán que los usuarios que las visiten (atraídos por su diseño gráfico) no vuelvan a hacerlo al haber experimentado estas carencias funcionales.



Actividades

1. ¿La temática de una página web puede influir de algún modo en la usabilidad de la misma? Razone la respuesta.



Definición

Página web

Documento electrónico realizado para la WWW (World Wide Web). Estos documentos pueden contener información tal como texto, imágenes, vídeo, animaciones, etc.



La Web consiste en un sistema de documentos enlazados y accesibles a través de Internet.

Uno de los principales elementos que caracterizan a las páginas web son los denominados hipervínculos (también conocidos como links o enlaces), que permiten, entre otras cosas, que el usuario “salte” de una página web a otra.

Un aspecto muy importante a tener en cuenta en el desarrollo de cualquier página web es que el diseño debe ser acorde con el contenido de la página en cuestión, es decir, con la información que el usuario espera obtener de ella.



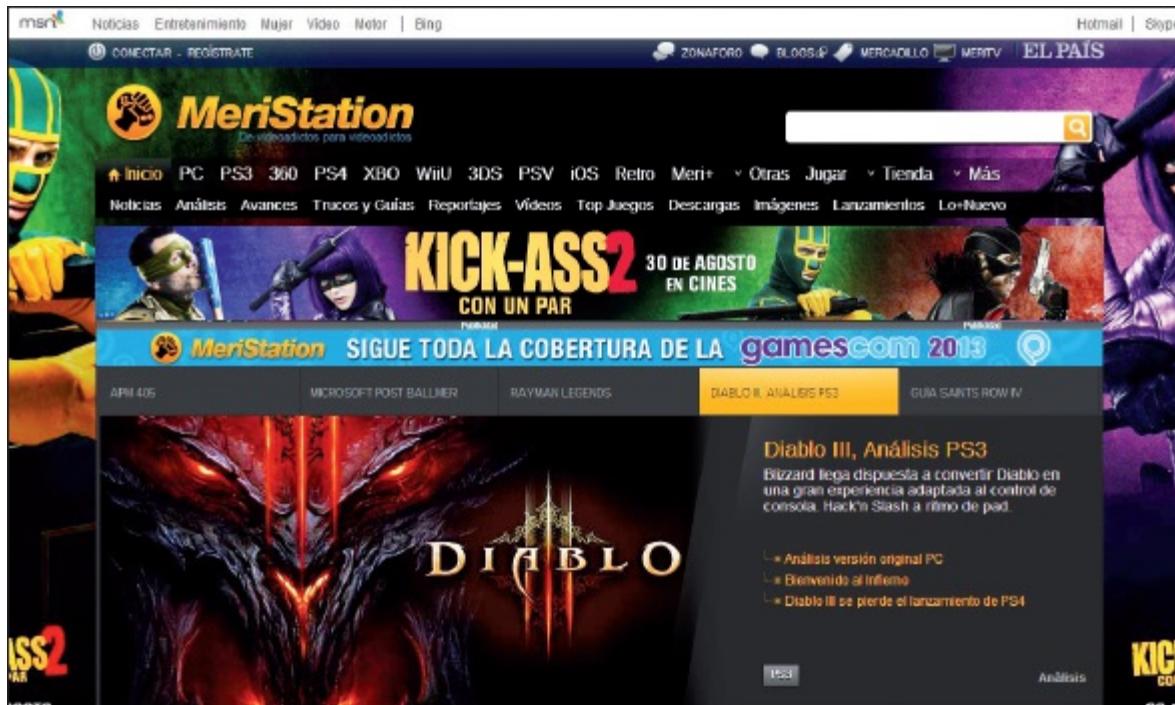
Aplicación práctica

Trabaja en una empresa de desarrollo de aplicaciones web y le encargan diseñar el aspecto de dos páginas distintas: la de un periódico digital y la de un sitio dedicado a video-juegos. ¿Crees que podría utilizar el mismo diseño para ambos trabajos? Razona su respuesta.

SOLUCIÓN

Como se ha comentado antes, es importante que el diseño de toda página web sea

acorde con el contenido de la misma. En el caso que se plantea, ambos sitios web presentarán contenidos bien diferenciados, por lo que también deberían serlo sus respectivos diseños. Por ejemplo, en el caso de la página dedicada a videojuegos, los colores fuertes podrían predominar sobre los demás y el uso de imágenes sería un aspecto muy importante a tener en cuenta en su diseño. Por otro lado, la página del periódico digital debería ofrecer un aspecto serio, elegante y ordenado y su diseño debería basarse en el uso de imágenes y texto.



Web de MeriStation, un ejemplo de página dedicada a video-juegos

SITUACIÓN CRÍTICA

Francia se suma a EE UU para atacar Siria

30.08.13 - 14:13 - AGENCIAS | WASHINGTON

Comentarios | Twitear | 1 | Compartir | Recomendar | 3

- La Casa Blanca deja claro que el rechazo británico no frenará sus planes
- Hollande dice que su país podría participar si se dan las condiciones necesarias
- Alemania descarta cualquier posibilidad de formar parte de una eventual coalición
- Los expertos de la ONU retoman sus investigaciones en el último día de su misión



VÍDEO

El presidente de Francia, François Hollande. / Foto: Iñaki Lanigorski (Efe) | Vídeo: Atres

NOTICIAS RELACIONADAS

- El Reino Unido no participará en una intervención en Siria
- **28/08/13:** «El uso de armas químicas no puede quedar sin respuesta»
- **28/08/13:** «No habrá acción militar en Siria hasta conocer el informe de la ONU»
- **28/08/13:** Siria será el «cementerio de los invasores»
- **28/08/13:** Rusia avisa de que un ataque desestabilizará Oriente Próximo
- **28/08/13:** Tensión en Israel ante la posible represalia siria
- **28/08/13:** Hackers sirios atacan la web del 'New York Times'
- **28/08/13:** El-Asad encara su mayor desafío
- **27/08/13:** «Francia está dispuesta a castigar a los que gasean a inocentes»
- **27/08/13:** El Parlamento británico votará el jueves sobre una acción militar
- **27/08/13:** Rusia pide «prudencia» a EE UU sobre un eventual ataque
- **Datos:** Principales intervenciones militares sin mandato de la ONU

Ejemplo de periódico digital. Web del Diario Sur de Málaga

2.1. Los principios básicos de un buen diseño estético

Para que una página web sea visualmente competente, es importante tener en cuenta ciertas consideraciones o principios básicos. A continuación, se enumeran los más importantes:

- **Balance:** equilibrio que debe haber entre los distintos elementos que constituyen la web. Por ejemplo: las imágenes y el texto, los elementos grandes y pequeños, las zonas oscuras y claras, etc., deben estar balanceadas de manera que la web resulte atractiva y, a su vez, que consiga llamar la atención del visitante sobre aquellos contenidos en los que interese.
- **Contraste:** manera de diferenciar los elementos que se desean destacar, llamando la atención del visitante sobre los elementos más importantes de una página pueden ser: diferencias tipográficas, diferencias en las formas (círculos, cuadrados, rectángulos, etc.), diferencias de tamaño, texturas y fondos distintos, etc.
- **Énfasis:** manera de diferenciar algo dentro de un elemento. Un ejemplo puede ser esta misma lista, en la que aparecen señalados (y se definen a continuación)

cada uno de los principios básicos de un buen diseño estético.

- **Repetición o ritmo:** manera de definir patrones con los que se ordenan los elementos que forman parte de la página web. Al hacer que los elementos de una página web sigan un patrón predefinido, es mucho más fácil para los usuarios comprender el contenido y visitarlo en poco tiempo. Este principio es fundamental para mantener la usabilidad del sitio.
- **Proximidad o unidad:** consiste en situar juntos o próximos los elementos que se encuentren relacionados. Esto permitirá que se formen unidades visuales que hagan que el contenido de la web sea más coherente.



Actividades

2. Estudie el diseño de alguna página y razoné si en ella se cumplen los principios de balance, contraste, énfasis, repetición y proximidad.
-

2.2. Diseño orientado al usuario. Diseño orientado a objetivos. Diseño orientado a la implementación

Fundamentalmente, existen tres tipos de orientaciones que se suelen seguir cuando se diseña un sitio web:

- Diseño orientado a objetivos.
- Diseño orientado al usuario.
- Diseño orientado a la implementación.

El diseño orientado a los objetivos consiste en definir y planificar los objetivos que se deseen alcanzar con el desarrollo del correspondiente sitio web. En definitiva, en este proceso se han de identificar los requerimientos del proyecto en cuestión, contestándose a la pregunta: ¿qué es lo que se pretende conseguir?

El diseño web centrado en el usuario se caracteriza por asumir que todo el proceso de diseñar y desarrollar la correspondiente página web debe estar conducido por el usuario, lo que significa que para dicho diseño se deben tener muy en cuenta las necesidades, características y objetivos que este desea alcanzar. Este tipo de diseño hace que sea necesario, desde el principio, implicar a los usuarios durante el desarrollo de la página, por lo que se hace fundamental conocer cómo son, qué es lo que necesitan

y para qué utilizan la página. En el diseño web centrado en el usuario es necesario evaluar el sitio con los propios usuarios, analizar sus impresiones ante el diseño, prestar atención a su experiencia de navegación (e innovar continuamente el diseño para mejorarla), etc. En definitiva, es muy importante que todo lo que se vaya diseñando sea evaluado constantemente.



En el diseño web orientado al usuario, es muy importante implicar a este durante todo el desarrollo de la página.



Importante

Evaluar una página web únicamente cuando se ha terminado su desarrollo hace que sea mucho más costosa y compleja la reparación de sus posibles errores de usabilidad, ya que siempre es más rentable replantear un diseño sobre la marcha que rehacerlo de nuevo una vez finalizado.

Por último, **el diseño web orientado a la implementación** consiste en centrar el diseño de la página en las posibilidades tecnológicas que están a la disposición del diseñador y que este es capaz de implementar. Un ejemplo de diseño centrado en la implementación puede consistir en diseñar un sitio según las posibilidades que ofrece la quinta versión del lenguaje de marcas HTML.

3. El proceso de diseño web

Durante el proceso de diseño de una página web es necesario seguir una serie de etapas que permitan transformar la información y el contenido que se desea poner a disposición del usuario en un sitio que haga que el navegante se sienta a gusto y pueda encontrar fácil y rápidamente lo que busque. A continuación, se enumeran una serie de etapas que pueden ayudar a conseguir esto:

1. Delimitar el tema.
2. Recolectar información.
3. Agregación y descripción.
4. Estructura de un sitio web y navegabilidad.
5. Estructura y composición de páginas.
6. Ensamble final.
7. Testeo. Compatibilidad con navegadores.



Consejo

Es recomendable no empezar a escribir ninguna línea de código (HTML, CSS, etc.) hasta que se finalicen las primeras etapas de diseño.

3.1. Delimitar el tema

En esta etapa inicial, se define qué es lo que se va a tratar en el sitio web y el contenido que va a añadir (y el que no). Se hace necesario definir los objetivos (tanto primarios como secundarios) que se desean alcanzar con el desarrollo de la página.

En esta fase también es muy importante evaluar correctamente la cantidad de tiempo disponible para elaborar el sitio, que tiene que ser acorde al necesario para desarrollarlo.

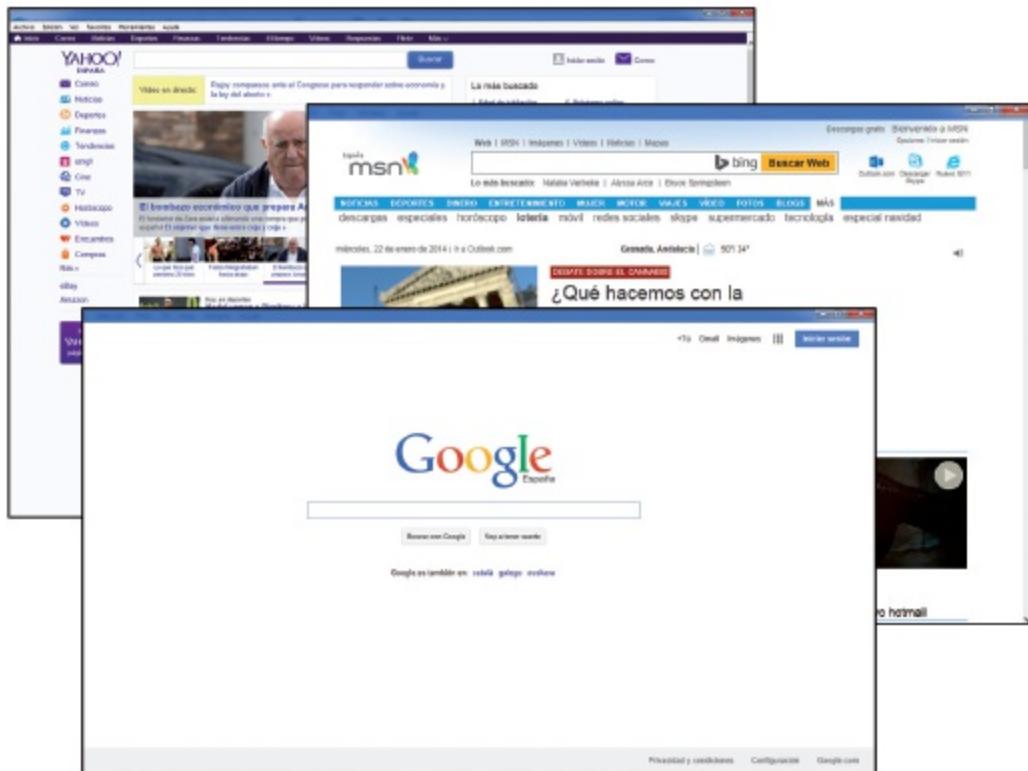


Actividades

3. ¿Es recomendable que un sitio web esté dedicado únicamente a un tema muy específico? Razone la respuesta.

3.2. Recolección de información

En esta etapa se obtiene la información que se desea añadir a la página, según las especificaciones de la primera etapa. Una vez recolectada, se hace necesario llevar a cabo un proceso de descarte sobre aquella información que no se considere relevante.



La propia Web es una fuente casi inagotable de información.

3.3. Agregación y descripción

Una vez que se dispone de la información que se alojará en la página, se hace necesario organizarla adecuadamente. En este punto, es muy importante establecer un equilibrio entre la linealidad y la jerarquización.



Recuerde

Además de diseñar gráficamente un sitio web, también es necesario planificar adecuadamente los contenidos de la página, así como estructurar cada uno de los menús, definir la manera en la que el usuario va a interactuar con cada uno de los elementos, etc.

Por un lado, es importante aplicar una clasificación de tipo lineal a aquellos trozos de información que requieren que el usuario que los consulte avance paulatinamente en el conocimiento de algo. Respecto a aquellos fragmentos de información que sean complementarios o que dependan uno de otro, se debería establecer una clasificación de tipo jerárquico, como apartados y subapartados.



Actividades

4. Imagine que trabaja en una empresa de diseño web y le asignan diseñar una página donde se alojará el contenido de un manual de informática. Dicho manual ya existe en formato papel y el trabajo consiste en transformarlo a formato web. Describa brevemente cómo se distribuiría el contenido para que quede organizado adecuadamente.
-

3.4. Estructura de un sitio web y navegabilidad

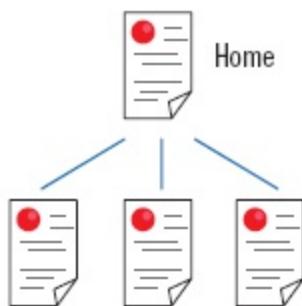
Esta fase consiste en definir la manera en la que se enlazan las distintas páginas, según la agregación de contenidos establecida en la etapa anterior.

Se podrían añadir:

- Ayudas para la navegación.
- Enlaces que permitan la jerarquización establecida.
- Hipervínculos entre elementos que pertenezcan a un mismo nivel jerárquico.

Una vez definida la estructura y navegabilidad del sitio web, el desarrollador ya puede empezar a implementarla con los lenguajes de programación correspondientes (HTML, CSS, etc.).

En la inmensa mayoría de los casos, el tipo de estructura de una web va a estar determinado por los contenidos que va a albergar



3.5. Estructura y composición de páginas

Para estructurar el contenido de cada uno de los documentos web que va constituir el sitio, es fundamental conocer las tres partes fundamentales en las que se suelen estructurar la inmensa mayoría de ellos. Estas son:

- Cabecera (*header*).
- Cuerpo (*body*).
- Pie de página (*footer*).

La **cabecera** se encuentra ubicada en la parte superior de la página web y, por lo general, contiene información relacionada con la temática del sitio. Las cabeceras suelen constar de:

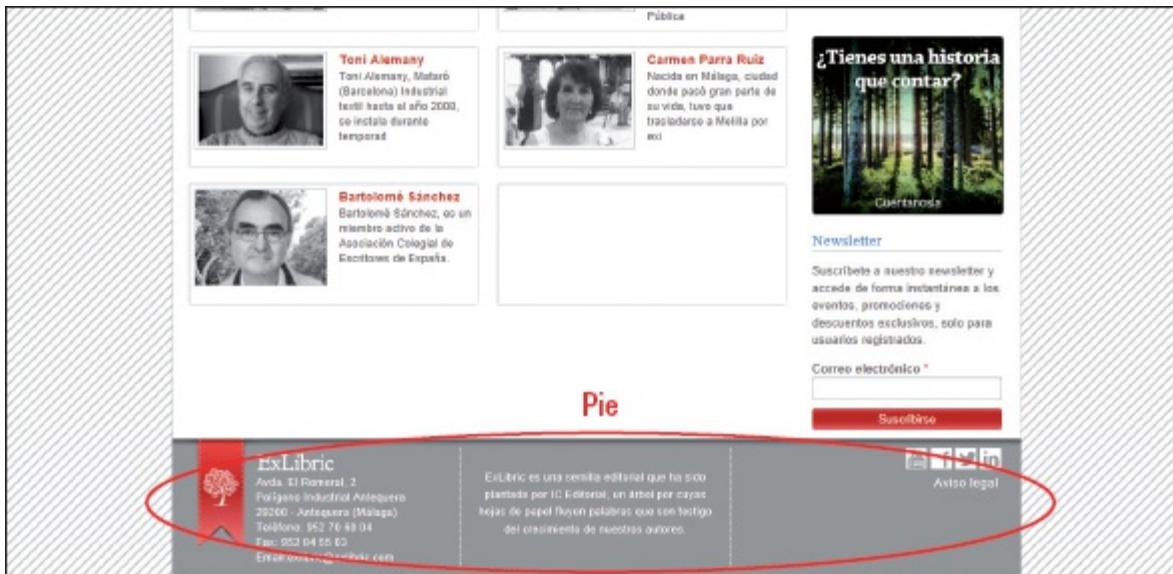
- Logo y título.
- Barra de navegación para acceder a las secciones principales del sitio.
- Información de contacto (si no va incluido en el pie).
- A veces, se incluyen campos de texto que funcionan como buscadores de contenido dentro de la propia página.

Ejemplo de cabecera de una web

En el **cuerpo** se localiza el contenido más relevante de un documento web. En esta parte, se pueden añadir barras laterales, las cuales suelen usarse para incluir contenido complementario y/o para mostrar *links* a otras páginas o secciones.

Ejemplo de cuerpo de una web

Por lo general, en el **pie de página** se puede encontrar información de contacto, menús que direccionan a las secciones más relevantes, *links* a otras páginas externas y, en general, contenido complementario relacionado con el tema de la página.



Ejemplo de pie de página de una web



Recuerde

Al hacer que los elementos de una página web sigan un patrón predefinido, es mucho más fácil para los usuarios comprender el contenido y visitarlo en poco tiempo. Este principio es fundamental para mantener la usabilidad del sitio.



Aplicación práctica

Está realizando el diseño de una página web que consiste en un manual interactivo de motores eléctricos. Desea que el sitio disponga de los siguientes elementos:

- Título del centro de formación que imparte el curso.
- Título del curso.
- Barra de navegación para acceder rápidamente a cada uno de los

capítulos.

- **Menú para acceder rápidamente a los apartados de cada capítulo.**
- **Contenido de cada capítulo (imágenes, texto, vídeos, etc.)**
- **Datos de contacto (correo electrónico, localidad donde se encuentra el centro de formación, etc.).**

Elabore una tabla que indique en qué parte de la página (cabecera, cuerpo o pie) insertará cada uno de los elementos anteriores.

SOLUCIÓN

Parte o zona de la página	Elementos
Cabecera	Título del centro de formación que imparte el curso. Título del curso. Barra de navegación para acceder rápidamente a cada uno de los capítulos.
Cuerpo	Menú para acceder rápidamente a los apartados de cada capítulo. Contenido de cada capítulo (imágenes, texto, vídeos, etc.).
Pie	Datos de contacto.

En cuanto al **estilo gráfico** de los sitios web, es importante tener en cuenta que un diseño gráfico adecuado puede ser algo determinante a la hora de hacer que los usuarios que la visiten se sientan cómodos.

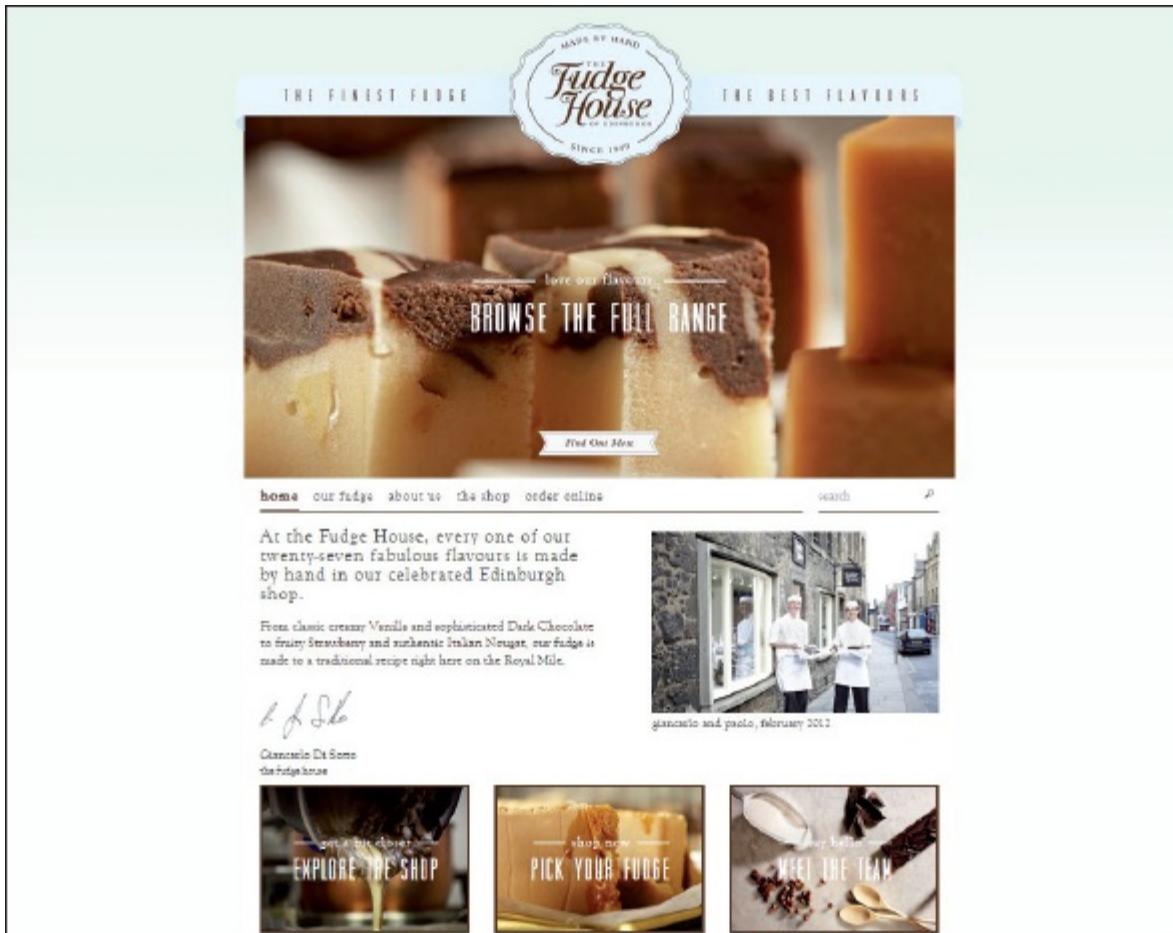


Nota

Una página puede funcionar muy bien pero, si no es visualmente atractiva, no llamará la atención de los usuarios.

Respecto al número y las dimensiones de las imágenes, es necesario establecer un equilibrio. Existen páginas que están repletas casi exclusivamente de grandes y lentas imágenes, mientras que también hay otros sitios que casi todo lo que contienen es texto, desaprovechando el uso de los recursos multimedia. Aunque el esquema a elegir siempre variará de diseñador en diseñador, hay que tener presente la búsqueda de un

equilibrio entre ambos extremos.



Cuando una página web es visualmente atractiva, se hace mucho más fácil captar la atención de los usuarios.

Otro aspecto importante en el diseño gráfico del sitio es intentar, en la medida de lo posible, mantener una cierta coherencia gráfica, además de atreverse a innovar en este sentido. Los colores de fondo, las texturas y las composiciones interesantes de los elementos dentro de una página es algo que puede perdurar en la memoria del usuario durante mucho tiempo.

3.6. Ensamble final

En esta etapa, el diseñador concreta el proyecto con los últimos contenidos que sea necesario añadir e inserta el estilo gráfico a las páginas del sitio. En este punto, la construcción de la página queda finalizada y ya solo queda testearla.

3.7. Testeo. Compatibilidad con navegadores

Por último, es indispensable revisar: la coherencia general del sitio, que no existan enlaces que no direccionen a ninguna parte (*links rotos*), la redacción y ortografía de todos los textos, subdividir aquellas páginas que tengan demasiado contenido (para separarlo en dos o más páginas), etc.

Puede ser muy interesante hacer que personas ajenas al diseño de la página interactúen con ella y saquen sus propias conclusiones. Es importante que esto se haga no solo al final, sino durante todo el desarrollo del sitio.



Recuerde

Evaluar una página web únicamente cuando ha terminado su desarrollo hace que sea mucho más costosa y compleja la reparación de sus posibles errores de usabilidad, ya que siempre es más rentable replantear un diseño sobre la marcha que rehacerlo una vez finalizado.

Otro aspecto fundamental en el testeo de cualquier página que se desarrolle es comprobar que se **visualiza correctamente** en, al menos, los navegadores más utilizados. El principal motivo por el que se debe hacer este tipo de comprobación es que, a veces, un mismo código HTML o CSS presenta comportamientos distintos dependiendo del navegador que lo esté interpretando.



Definición

Navegador

Programa que canaliza las peticiones del usuario que solicita una información por la Web. La mayoría de estos programas son públicos, aunque existen algunos de pago.

Los navegadores más utilizados en la actualidad son:

- Internet Explorer.
- Mozilla Firefox.
- Google Chrome.
- Safari.

- Opera.



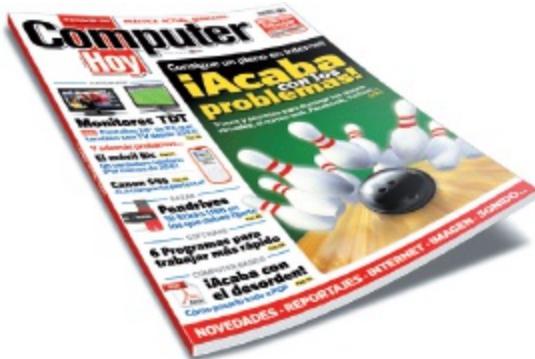
Logotipos de los navegadores más utilizados

4. Diferencias entre diseño orientado a presentación y a impresión

En el proceso de diseño orientado a impresión, se diseñan contenidos que están destinados a ser impresos en algún formato físico, como periódicos, revistas, carteles, etc.

El diseño web es un diseño orientado a la presentación de contenidos a través del navegador y, aunque estos programas suelen permitir la impresión en formato físico de las páginas que visualizan, es importante tener siempre en cuenta las evidentes diferencias que existen entre este tipo de diseño y el orientado a la impresión.

Por ejemplo, cuando se diseña y maqueta el contenido de una revista física siempre se piensa en establecer tamaños y distribuir contenidos de manera que se vean correctamente en un formato que tiene unas dimensiones fijas (tamaño de la hoja de la revista). Por el contrario, cuando se diseñan contenidos web, hay que ser consciente de que la página podrá visualizarse en un amplio rango de resoluciones y tamaños de pantalla (PC, móviles, tabletas, etc.).



El diseño de una revista física es un ejemplo de diseño orientado a la impresión.

Otra gran diferencia entre ambos diseños está relacionada con la ausencia o no de recursos interactivos y multimedia, tales como vídeos, audio, etc., ya que la existencia de estos recursos es nula en cualquier contenido impreso.

5. Resumen

El proceso de diseño de una página web no es un proceso que se limite únicamente a escribir y programar código, sino que se trata de una tarea mucho más compleja para el diseñador, ya que debe tener en cuenta muchas otras variables para conseguir que el sitio web a desarrollar presente unos requisitos mínimos de usabilidad y atractivo gráfico.

Para diseñar contenidos web competentes y atractivos para el usuario, es importante conocer:

- Los principios básicos del diseño web profesional.
- Los tipos y características de las principales orientaciones de diseño (objetivos, usuario e implementación).
- Las etapas que se pueden seguir para diseñar adecuadamente un sitio web:
 - Delimitar el tema.
 - Recolectar información.
 - Estudiar la agregación y descripción de contenidos.
 - Diseñar la estructura y navegabilidad del sitio web.
 - Definir la estructura y composición de las páginas.
- Efectuar el ensamblaje final del sitio.

- Testear el sitio y verificar que es compatible con los navegadores que más se suelan utilizar.



Ejercicios de repaso y autoevaluación

1. Defina brevemente el concepto de diseño web.

2. El diseño web se puede dividir en dos partes fundamentales. Estas son:

- a. Funcionalidad y aspecto.
- b. Implementación en HTML y desarrollo CSS.
- c. Usabilidad y funcionalidad.
- d. Ninguna de las respuestas anteriores es correcta.

3. Complete el siguiente texto.

Un aspecto muy importante a tener en cuenta en el desarrollo de cualquier página web es que el _____ debe ser acorde con el _____ de la página en cuestión, es decir, con la _____ que el usuario espera _____ de ella.

4. Explique el concepto de balance en el diseño de una página web.

5. Explique en qué consiste el diseño web orientado al usuario.

6. El diseño web orientado a la implementación consiste en centrar el diseño de la página en...

- a. ... la experiencia del usuario.
- b. ... los objetivos que se desean alcanzar.
- c. ... las posibilidades tecnológicas disponibles y que el desarrollador

- sea capaz de implementar.
- d. ... el atractivo gráfico.

7. Complete el siguiente texto.

Una vez que se dispone de la información que se alojará en la página, se hace necesario _____ adecuadamente. En este punto, es muy importante establecer un equilibrio entre la _____ y la _____.

8. Una vez definida la estructura y navegabilidad del sitio web, el desarrollador ya puede empezar a...

- a. ... recolectar información.
- b. ... estructurar el contenido.
- c. ... programarla.
- d. Las respuestas b. y c. son correctas.

9. Enumere las distintas partes en las que se suele estructurar un documento web.

10. Complete la siguiente frase.

La _____ se encuentra ubicada en la parte superior de la página web y, por lo general, contiene información relacionada con la _____ del sitio.

11. ¿Qué contenido se suele incluir en el *footer* de una página web?

12. Respecto al número imágenes de una página web, es necesario...

- a. ... que sean muy numerosas.
- b. ... que sean poco habituales.
- c. ... que se mantenga un equilibrio.
- d. ... que sean poco numerosas pero de grandes dimensiones.

13. Explique en qué consiste el proceso de testeo de una página web, como última etapa en el diseño de la misma.

14. Nombre algunos de los navegadores más usados en la actualidad.

Capítulo 2

Lenguajes de marcado generales

1. Introducción

Los lenguajes de marcado o lenguajes de marcas se pueden definir como una manera de codificar documentos donde, junto con el texto, se establecen etiquetas, marcas o anotaciones que contienen información relacionada con la estructura de dicho texto, su forma de visualización, etc.

Aunque HTML es el lenguaje de marcas que se utiliza para crear páginas web, no es el único lenguaje de este tipo que existe. En este capítulo, se va estudiar la evolución histórica de estos lenguajes, desde sus orígenes hasta la aparición del metalenguaje XML. También se van a analizar las características de algunos de los lenguajes de marcas más conocidos y se enumerarán los principales elementos a partir de los cuales se estructura un elemento escrito con lenguaje de marcas.

2. Origen de los lenguajes de marcado generales: SGML y XML

El origen de los lenguajes de marcas empezó con la necesidad de estandarizar el formato de los documentos electrónicos y, en la década de los 70, la empresa estadounidense IBM fue la encargada de solucionar el problema de la descripción de documentos.

Logo de IBM



2.1. Origen: GML y SGML

En los años ochenta, IBM propuso un sistema de documentos que era capaz de identificar los elementos lógicos que contenían (títulos, capítulos, párrafos, listas, etc.) con algún tipo de etiqueta dentro del documento correspondiente. Esto no se refería al aspecto del documento, sino a su estructura lógica, de manera que la visualización e impresión de los documentos podía no depender del *hardware* en particular. No obstante, un sistema de estas características necesitaría algún tipo de *software* concreto para poder visualizarse o imprimirse. Lo que hizo IBM fue crear algo parecido a un pseudolenguaje que combinara únicamente texto e instrucciones de formateado. Este lenguaje se denominó **lenguaje de marcas o de marcado** (*markup language*) e IBM le dio el nombre de Lenguaje de Marcas Generalizado o GML (*Generalized Markup Language*).



Definición

Estructura física y lógica

En el estudio de los lenguajes de marcas, es importante conocer y diferenciar los conceptos de estructura lógica y física de un documento.

La estructura lógica está formada por las partes y relaciones que lo constituyen. Por otro lado, la estructura física indica el aspecto del documento, ya sea sobre el papel o en la pantalla donde se visualice, incluyendo sus componentes físicos, el lugar donde se colocan sus elementos y la tipografía de los textos que contiene.

IBM no llegó muy lejos con GML, pero, en 1986, la Organización Internacional de Estándares (ISO) presentó la publicación ISO 88879, que definía un lenguaje de marcas que servía para crear documentos estandarizados, el cual fue creado a partir de GML. A este nuevo lenguaje se le dio el nombre de SGML (Standard Generalized Markup Language).



SGML es un sistema para organizar y etiquetar documentos que fue creado por la Organización Internacional de Estándares (ISO).

Al igual que GML, SGML no almacena el diseño de los documentos, sino la estructura lógica de los mismos. Esto se consigue gracias al uso de etiquetas o *tags* que se incluyen entre los signos “<” y “>”. Estas etiquetas indican el comienzo y el final de elementos tales como párrafos, listas, títulos, etc. En definitiva, el objetivo de la creación de este lenguaje era asegurar que los documentos electrónicos importantes fuesen independientes de los formatos de archivo binario (*Word*, *WordPerfect*, etc.) y de los sistemas operativos (*Windows*, *Mac OS*, etc.).

El SGML no prosperó debido fundamentalmente a su complejidad y al elevado coste que suponía adquirir una de las herramientas que se utilizaban para su creación.

2.2. El lenguaje HTML

En 1990, Tim Berners-Lee, del Laboratorio Europeo de Física de Partículas, creó el lenguaje HTML (*HyperText Markup Language*), partiendo de SGML y de la idea de que en un documento se pudieran establecer enlaces que permitieran navegar de un documento a otro (hipertextos). Esto constituyó la base de la WWW (*World Wide Web*), conocida comúnmente como Web.

HTML resultó ser un lenguaje muy sencillo, lo cual ayudó considerablemente en la popularización de la Web. El crecimiento de esta última fue tal que se decidió crear el W3C (*World Wide Web Consortium*) cuya misión sería el desarrollo de lenguajes y estándares para la WWW.

El W3C es una comunidad internacional que trabaja para crear estándares para la WWW



Durante años, fueron apareciendo distintas versiones de este lenguaje de marcas, las cuales permitieron mejorar, ampliar e innovar sus posibilidades respecto a la implementación de elementos y funcionalidades nuevas en las páginas web.



Actividades

-
1. Enumere plataformas que sean capaces de conectarse a Internet y visualizar páginas web.
-

2.3. XML y XHTML

Con el paso del tiempo, se fue apreciando que HTML era un lenguaje destinado a la presentación de documentos a través de la WWW, pero no para impresión o tareas de diseño. Además, en HTML la estructura y el aspecto de los documentos se mezcla, lo que suponía problemas a la hora de intentar navegar con múltiples tecnologías (teléfonos móviles, navegadores modo texto, etc.).

En 1998, el W3C creó una versión más sencilla de SGML, con el fin de poder afrontar los problemas de compatibilidad y adaptación de las nuevas tecnologías que iban surgiendo. A esta versión se le dio el nombre de XML (Lenguaje de Marcas eXtensible o *eXtensible Markup Language*).



Definición

XML

Metalenguaje que sirve como estándar para el intercambio de información estructurada entre diferentes plataformas. Este metalenguaje presenta una amplia variedad de aplicaciones, como pueden ser: editores de texto, hojas de cálculo, bases de datos, etc.

Pie de imagen: XML es un metalenguaje extensible de etiquetas creado por el W3C.

Existe una diferencia fundamental entre HTML y XML: en XML no existen elementos predefinidos, es decir, el desarrollador puede crear, siguiendo ciertas reglas, su propio lenguaje y vocabulario para el formateo de la información contenida en los documentos.



Ejemplo

En HTML, los documentos se estructuran utilizando una serie de etiquetas predefinidas, por ejemplo:

```
...
<p>Nombre:</p>
<p>Ramón</p>
<p>Primer apellido:</p>
<p>Guerrero</p>
...
```

Por el contrario, en XML, el usuario puede decidir el nombre de las etiquetas que usará para organizar el contenido del documento que esté escribiendo. Por ejemplo:

```
...
<nombre>Ramón</nombre>
<primer_apellido>Guerrero</primer_apellido>
...
```

Otra diferencia importante que existe entre HTML y XML es que este último es totalmente estructural, es decir, los documentos escritos en XML no incluyen ninguna información relacionada con el aspecto y el diseño de los mismos, almacenando únicamente datos e información estructural.

No obstante, la utilización de XML para crear documentos web resultaba realmente complicada y, además, muchos navegadores tenían problemas para interpretar correctamente estos documentos. Por este motivo, se definió un nuevo lenguaje que incluyera las posibilidades de HTML y que mantuviera la filosofía de XML: el XHTML (*eXtensible HyperText Markup Language*).

<?xml?>

XHTML se creó reformulando el lenguaje HTML, haciéndolo compatible con XML.

Durante mucho tiempo, el XHTML se presentó como una evolución de HTML, el cual se debería abandonar en el futuro. No obstante, para la inmensa mayoría de los desarrolladores de páginas web suponía un esfuerzo considerable adaptar lo que se escribía en HTML a XHTML, por lo que, cada vez más, iban abandonando su uso.

Por este motivo, el W3C aceptó, con el tiempo, que el lenguaje XHTML podía coexistir con HTML.



Actividades

2. ¿Contendrá un documento escrito en XML información acerca del tipo de letra (Arial, Times, etc.) en el que debe mostrarse un párrafo determinado? Razona la respuesta.
-

3. Características generales de los lenguajes de marcado

Por lo general, los lenguajes de marcas presentan las siguientes características fundamentales:

- Combinan la información contenida en un documento (por lo general, texto) con marcas o indicaciones relacionadas con su estructura o la forma de representarlo.
- El propio lenguaje de marcas es el que especifica los tipos de etiquetas que se pueden emplear, en qué lugares pueden colocarse y el significado que tiene cada una de ellas.
- A diferencia de otros lenguajes de programación, los lenguajes de marcas carecen de variables y funciones aritméticas.
- La presencia de etiquetas intercaladas en un documento escrito con lenguaje de marcas refleja inequívocamente su estructura y jerarquía.
- Los programas que interpretan los documentos escritos con lenguajes de marcas no suelen mostrar al usuario las etiquetas propiamente dichas, ya que este suele estar interesado en visualizar únicamente el propio contenido del documento.
- Fundamentalmente, existen tres términos que se suelen usar para describir las partes de un documento escrito con lenguajes de marcas: las etiquetas, los elementos y los atributos (ver a continuación).



Recuerde

Los lenguajes de marcado o lenguajes de marcas se pueden definir como una manera de codificar documentos donde, junto con el texto, se establecen etiquetas, marcas o anotaciones que contienen información relacionada con la estructura de dicho texto, su forma de visualización, etc.

4. Estructura general de un documento con lenguaje de marcado

En un documento escrito con un lenguaje de marcas se pueden distinguir las siguientes partes fundamentales:

- Metadatos e instrucciones de proceso.
- Codificación de caracteres. Caracteres especiales (escape).
- Etiquetas o marcas.
- Elementos.
- Atributos.
- Comentarios.

4.1. Metadatos e instrucciones de proceso

Por un lado, los **metadatos** consisten en información complementaria (metainformación) que se puede incluir dentro de los documentos escritos con lenguaje de marcas (y en otros muchos tipos de archivos), la cual está relacionada con el documento en sí. A continuación, se listan algunos ejemplos de metainformación que puede añadirse a las páginas web escritas en HTML:

- Nombre del desarrollador de la página.
- Nombres de aplicaciones informáticas utilizadas en el desarrollo de la página.
- Temática de la web.
- Palabras clave relacionadas con el contenido de la página (las cuales pueden utilizarse por los buscadores).

Para incluir metainformación en HTML, se utiliza la etiqueta <meta>.

Por otro lado, las **instrucciones de proceso** indican, al programa correspondiente, la manera en la que debe interpretar el documento en cuestión.

Por ejemplo, en los documentos escritos en XML, es necesario incluir una instrucción de proceso que consiste en la declaración de documento XML. Por ejemplo:

```
<?xml version="1.0"?>
```

Como se puede apreciar, la sintaxis de las declaraciones de proceso empiezan con "`<?"` y termina con "`?>`". Dentro de estos caracteres, lo primero que se indica es el identificador de la instrucción (xml). Posteriormente, se pueden ir añadiendo parámetros o argumentos que la instrucción soporte.

4.2. Codificación de caracteres. Caracteres especiales (escape)

En informática, la codificación de caracteres consiste en un método que se utiliza para convertir un carácter de un lenguaje natural en un símbolo perteneciente a otro sistema de representación, como por ejemplo en un número. Esto es fundamental para solucionar problemas de almacenamiento de texto en computadoras y demás dispositivos, además de facilitar la transmisión de texto a través de la redes de telecomunicaciones.

Para que los documentos escritos con lenguajes de marcas sean compatibles con los caracteres propios de nuestro idioma (y de Europa Occidental), es necesario utilizar la codificación **UTF-8**.

No obstante, para asegurar que una página se va a ver correctamente en cualquier ordenador del mundo, también se puede recurrir a los denominados **caracteres especiales** o de escape.

Por ejemplo, en HTML, existen dos formas de incluir caracteres especiales:

- Mediante un **número decimal**, siguiendo la sintaxis (el número decimal corresponderá al código del carácter que se quiera incluir):

&# + número del carácter +;

- Mediante un nombre **mnemotécnico**, siguiendo la sintaxis:

& + número del carácter +;



Ejemplo

El siguiente ejemplo muestra la manera de escribir una frase utilizando nombres mnemotécnicos:

Texto normal:

- La taza de té estaba demasiado fría

Texto con caracteres especiales:

- La taza de té estaba demasiado fría

A continuación, se muestra una tabla que indica los caracteres especiales a utilizar para algunos de los símbolos y caracteres más utilizados:

Algunos de los caracteres especiales más utilizados en las páginas web

Signo	Código	Mnemo
<		<
>		>
&		&

“		"
á	á	á
Á	Á	Á
é	é	á
É	É	É
í	í	í
Í	Í	Í
ó	ó	ó
Ó	Ó	Ó
ú	ú	ú
Ú	Ú	Ú
ñ;	ñ	ñ
Ñ	Ñ	Ñ
(Espacio en blanco)		



Aplicación práctica

Está realizando una página web en la que desea incluir un párrafo utilizando los caracteres especiales de la tabla anterior. El texto es el siguiente:

Las etiquetas están constituidas por una serie de códigos “encerrados” dentro de dos signos: “<” y “>”. Cuando se trate de un etiqueta de cierre, es necesario añadir el carácter “/” justo después del signo “<”.

Escriba el texto tal y como lo incluiría en el documento web en cuestión.

Nota: utilice preferiblemente códigos decimales.

SOLUCIÓN

Las etiquetas están constituidas por una serie de caracteres encerrados dentro de dos signos: " < " y " > ". Cuando se trate de un etiqueta de cierre, es necesario añadir el carácter " / " justo después del signo " < ".



Actividades

3. Escriba de nuevo el texto de la aplicación práctica anterior utilizando preferiblemente nombres mnemotécnicos.
-

4.3. Etiquetas o marcas

Una etiqueta (*tag*), también conocida como marca o directiva, es un texto que va “encerrado” entre el símbolo menor que (<) y el símbolo mayor que (>).

Existen dos tipos fundamentales de etiquetas: las de inicio (como por ejemplo: <nOMBRE>) y etiquetas de fin (como puede ser: </nOMBRE>). Como se puede observar, las etiquetas de cierre se escriben igual que las de apertura, pero indicando el signo "/" justo antes del nombre de la etiqueta en cuestión.

Las etiquetas siempre afectan al texto (y otras directivas) que se encuentran dentro de su apertura y cierre.

Existen etiquetas que no requieren de apertura y cierre. Para estas, se suele añadir el carácter "/" justo antes de ">". Por ejemplo:
, , <meta/>, etc.



Recuerde

En HTML, los documentos se estructuran utilizando una serie de etiquetas predefinidas, por ejemplo:

```
...  
<p>Nombre:</p>  
<p>Ramón</p>  
<p>Primer apellido:</p>  
<p>Guerrero</p>  
...
```

Por el contrario, en XML, el usuario puede decidir el nombre de las etiquetas que usará para organizar el contenido del documento que esté escribiendo. Por ejemplo:

```
...  
<nombre>Ramón</nombre>  
<primer_apellido>Guerrero</  
primer_apellido>  
...
```

4.4. Elementos

Los elementos de los lenguajes de marcas son las estructuras a partir de las cuales se organizará el contenido del documento y las acciones que se ejecutarán cuando el navegador lo interprete. En definitiva, los elementos constan de la etiqueta de inicio, la etiqueta de fin y todo el contenido que se encuentra encerrado entre ambas.



Ejemplo

En los documentos escritos con lenguajes de marcas, es muy común que exista confusión entre lo que es una etiqueta y lo que es un elemento. En el siguiente ejemplo, se puede apreciar claramente la diferencia entre ambos:

```
<nombre>Juan Fernández</nombre>
```

En este código XML, se puede identificar lo siguiente:

<nombre>:	etiqueta de apertura.
</nombre>:	etiqueta de cierre.
<nombre>Jorge</nombre>:	elemento (nombre)
Juan Fernández:	contenido del elemento nombre.

Los elementos de los lenguajes de marcas pueden contener tanto texto como otros elementos. A continuación, se muestra un ejemplo de elemento HTML (párrafo) que contiene únicamente texto:

```
<p>Producto con precio rebajado debido a su escasa demanda</p>
```



Nota

Es posible que un elemento HTML no tenga contenido o, lo que es lo mismo, que esté vacío. Por ejemplo: <p></p> .

Por otro lado, en el siguiente fragmento de código XML, se puede apreciar un elemento (datos) que contiene otros dos elementos (nombre y apellido):

```
<datos>
    <nombre>Juan</nombre>
    <apellido>Fernández</apellido>
</datos>
```

Es importante tener en cuenta que todos los elementos que se escriben con lenguajes de marcas deben abrirse y cerrarse con la etiqueta que los define. Por ejemplo, el siguiente fragmento de código sería incorrecto:

```
<datos>
    <nombre>Juan</nombre>
    <apellido>Fernández</datos>
</apellido>
```



Actividades

4. Detecte y corrija los errores del siguiente fragmento de código XML:

```
<coche>
    <km>180.000</conductor>
    <conductor>Juan</km>
    <precio>12.000 €<precio>
</vehiculo>
```



Sabía que...

El XML es sensible al uso de mayúsculas y minúsculas, ya que los trata como caracteres diferentes. Por ejemplo, si un elemento se ha definido como “ELEMENTO”, no sería correcto referirse a él como “elemento”, “Elemento”, etc.

4.5. Atributos

Un atributo es un conjunto formado por un nombre y un valor que se localiza dentro de la etiqueta de inicio de un elemento. Este par (nombre-valor) indica alguna propiedad asociada al elemento en cuestión.

A continuación, se muestra un ejemplo de uso de un atributo en un elemento:

```
<elemento propiedad1="valor1">Contenido</elemento>
```

Como se puede observar, después del nombre del atributo se hace necesario incluir el símbolo “=” y, a continuación, se indica su valor entre comillas.

También es posible incluir más de un atributo en un elemento. Por ejemplo:

```
<elemento propiedad1="valor1"  
propiedad2="valor2">Contenido</elemento>
```



Actividades

5. Identifique los atributos (nombre y valor) del siguiente código HTML:

```
<div class="slide" id="slide_1">
    <p class="parrafo">Esto es un párrafo</p>
    <p class="parrafo">Esto es otro párrafo</p>
</div>
```



Nota

Los atributos que se definen dentro de los elementos HTML se suelen incluir para alterar el comportamiento de los mismos.

4.6. Comentarios

Los comentarios son bloques de texto que se pueden incluir en los documentos escritos con lenguajes de marcas y que son ignorados por los programas que visualizan el contenido de los mismos (por ejemplo un navegador).

Los comentarios empiezan por la cadena “`<!--`” y terminan con la cadena “`-->`”. Por ejemplo:

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

Los comentarios son muy útiles para los desarrolladores, ya que pueden indicar anotaciones y aclaraciones en el código que escriben sin que estas afecten al procesamiento del documento en cuestión.

Por ejemplo, el siguiente fragmento de código corresponde al contenido de un documento web, el cual está constituido únicamente por un párrafo y un comentario:

```
...
<p>Esto es un párrafo</p>
<!--Esto es un simple comentario--&gt;
...</pre>
```

Si se abriera el documento en el navegador, únicamente se visualizaría el contenido de la etiqueta párrafo (<p>) :



Visualización de un documento web con comentario



Aplicación práctica

Se dispone a modificar un documento escrito en XML cuyo código es el siguiente:

```

<?xml version="1.0"?>
<nivel1 estado="visible">
<titulo_n1>Interpretaci&oacute;n gr&aacute;fica</titu-
lo_n1>
<contenido_n1>El dibujo es una de las herramientas
m&aacute;s &uacute;tiles y necesarias a la hora de
plasmar, en un papel y de manera r&aacute;pida, el
concepto que queremos construir y hacer realidad;
por esto hay ciertas necesidades t&eacute;cnicas
que debemos plantear en este t&iacute;tulo.</conte-
nido_n1>
</nivel1>
<nivel2 estado="visible">
    <titulo_n2>Escalas</titulo_2>
</nivel2>
<nivel2 estado="oculto">
    <titulo_n2>Planos</titulo_2>
</nivel2>

```

Antes de hacer nada, desea analizar el contenido que ya tiene, identificando sus partes principales.

Elabore una tabla en la que indique la presencia, en este código XML, de etiquetas de apertura, atributos, caracteres especiales e instrucciones de proceso.

SOLUCIÓN

Tipo	Código
Instrucción de proceso	<?xml version="1.0"?>
Caracteres especiales	ó á á ú á é í
Etiquetas de apertura (atributos)	<nivel1> (estado), <titulo_n1>, <contenido_n1>, <nivel2> (estado), <titulo_n2>, <nivel2> (estado), <titulo_n2>

5. Documentos válidos y bien formados. Esquemas

La validación XML consiste en la comprobación de que un documento escrito en este estándar está bien formado y además es válido.

Se considera que un documento XML está bien formado si cumple las siguientes reglas:

- Todos los elementos tienen que tener su correspondiente cierre.
- Tener siempre presente que XML es sensible a mayúsculas.
- El anidamiento de los elementos debe ser correcto.
- Cada documento XML debe presentar un elemento raíz que no puede ser duplicado. Este es el primero que es “abierto” y el último que se “cierra”.
- Cada uno de los atributos deben escribirse entre comillas.
- Respecto a los atributos que son booleanos, su valor debe explicitarse de manera redundante.

Nota: Una variable de tipo booleano es una variable que solo acepta los valores 0 y 1.

- Deben preservarse los espacios en blanco.
- Los comentarios deben tener la sintaxis: <!-- comentario -->.
- Los elementos pueden nombrarse con: letras, números y los signos de puntuación: guión (-), guión bajo (_) y punto (.).
- Los nombres de los elementos no pueden comenzar por números o un signo de puntuación (esto no es aplicable al guión bajo). Tampoco pueden empezar por la cadena “xml” (ya sea mayúsculas o minúsculas).
- Los elementos no pueden nombrarse con ningún espacio.

Por otro lado, un documento XML es considerado válido si cumple las limitaciones que le imponga una DTD o un esquema. A continuación, se describe el significado de estos dos documentos:

- Una DTD es un documento que está escrito en un lenguaje creado a partir de SGML y, gracias a este, se pueden especificar nombres de elementos concretos y asignarles los atributos que pueden aceptar y sus valores posibles, qué elementos se pueden anidar dentro de ellos y con qué asiduidad pueden (o tienen que) aparecer, etc.
- Un esquema es un documento que ha sido creado en un lenguaje basado en XML, presentando la misma finalidad que la de una DTD: definir restricciones para un documento escrito en XML. El W3C creó un documento explicativo sobre esquemas (XML Schema) y ha desarrollado:

- Una recomendación sobre sus estructuras (<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028/>).
- Una recomendación sobre los tipos de datos (<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/>).
- Un boceto sobre sus componentes (<http://www.w3.org/TR/2005/WD-xmlschema-ref-20050329/>).

6. Resumen

Un lenguaje de marcas (o de marcado) consiste en un lenguaje que marca textos de manera que el ordenador sea capaz de manipularlos. La mayoría de los lenguajes de marcas pueden ser interpretados gracias a que dichas marcas se establecen con una sintaxis determinada que las diferencia claramente del texto en sí.

HTML es el lenguaje de marcas con el que se construyen las páginas web. No obstante, este no es el único lenguaje de este tipo que existe.

En el estudio de los lenguajes de marcado generales, se hace fundamental conocer:

- Origen y evolución de los lenguajes de marcas.
- Definición y características de algunos de los lenguajes de marcas más conocidos (SGML, XML, HTML, XHTML, etc.)
- Características generales de los lenguajes de marcado.
- Principales elementos que pueden aparecer en un documento escrito con lenguaje de marcas (instrucciones de proceso, etiquetas, atributos, caracteres especiales, etc.).
- Validación XML: documentos válidos y bien formados.



Ejercicios de repaso y autoevaluación

1. Complete el siguiente texto.

Los lenguajes de marcado o lenguajes de marcas se pueden definir como una _____ de _____ documentos donde, junto con el texto, se establecen etiquetas, marcas o anotaciones que contienen _____ relacionada con la _____ de dicho texto, su forma de visualización, etc.

2. ¿Cuál de los siguientes lenguajes de marcas es más antiguo?

- a. GML.
- b. SGML.
- c. XML.
- d. HTML.

3. De las siguientes afirmaciones, diga cuál es verdadera o falsa.

El lenguaje SGML fue creado a partir de GML.

- Verdadero
- Falso

SGML almacena el diseño y la estructura lógica de los documentos.

- Verdadero
- Falso

El W3C (World Wide Web Consortium) tiene la misión de desarrollar lenguajes y estándares para la WWW.

- Verdadero
- Falso

4. ¿Qué es XML?

5. Indique las diferencias fundamentales que existen entre HTML y XML.

6. El lenguaje que combina las posibilidades de HTML y la filosofía de XML es:

- a. DHTML.
- b. SGML.
- c. XHTML.
- d. JavaScript.

7. Complete el siguiente texto.

Los metadatos consisten en información _____ (metainformación) que se puede incluir dentro de los documentos escritos con lenguaje de marcas (y en otros muchos tipos de archivos), la cual está relacionada con el _____ en sí.

8. ¿Qué son las etiquetas? ¿Qué tipos existen?

9. Identifique las etiquetas, el contenido y los elementos del siguiente código HTML:

```
<div>Esto es un ejemplo de texto</div>
```

10. El conjunto formado por un nombre y un valor que se localiza dentro de la etiqueta de inicio de un elemento se denomina...

- a. ... instrucción.
- b. ... propiedad.
- c. ... atributo.
- d. Todas las opciones son incorrectas.

11. Imagine que desea indicar un comentario en un documento HTML que tenga el siguiente texto: “Esto es el principio de la página.” Escriba el código HTML de dicho comentario.

12. En el lenguaje XML, ¿es lo mismo escribir <etiqueta> que <Etiqueta>? Razone su respuesta.

13. De las siguientes afirmaciones, diga cuál es verdadera o falsa.

Para que un documento XML esté bien formado, el valor de los atributos debe ir sin comillas.

- Verdadero
- Falso

En los documentos XML bien formados, los nombres de los elementos pueden empezar con un número.

- Verdadero
- Falso

En los documentos XML bien formados, los nombres de los elementos no pueden nombrarse con espacios.

- Verdadero
- Falso

14. Complete el siguiente texto.

Un documento XML es considerado _____ si cumple las limitaciones que le imponga una DTD o un _____.

15. ¿Qué es una DTD?

Capítulo 3

Lenguajes de marcado para presentación de páginas web

1. Introducción

HTML es el lenguaje básico que se utiliza para escribir los documentos que se alojan en la *World Wide Web*. Cuando el navegador accede a estos, interpreta el código que los constituye y muestra la información traducida al usuario.

Básicamente, los documentos HTML son ficheros de texto (con la extensión “.html” o “.htm”) que pueden ser editados y creados a partir de cualquier editor de texto plano. Estas aplicaciones se utilizan para escribir texto sin formato y no permiten la inclusión de imágenes ni demás recursos multimedia. Para poder escribir código HTML es muy importante disponer de algún programa de este tipo, aunque también existen aplicaciones de edición web que incorporan su propio editor de código HTML (además de otras muchas herramientas). El *Bloc de notas de Windows* y la aplicación *TextWrangler* para *Mac* son claros ejemplos de editores de texto plano.

En este capítulo, se van a estudiar los fundamentos de este lenguaje de marcas y se hará especial hincapié en el aprendizaje de la implementación y configuración de los principales elementos que pueden formar parte de los documentos web.

2. Historia de HTML y XHTML. Diferencias entre versiones

Por lo general, la mayoría de los lenguajes de programación permanecen estables en el tiempo y, en caso de sufrir alguna modificación, cada una de las versiones queda identificada inequívocamente respecto a las demás. El desarrollo de cualquier actualización en un lenguaje de programación es un proceso complejo que requiere un acuerdo con las diferentes organizaciones de estandarización (ISO, ANSI, IEEE, etc.).

En cuanto a HTML, existen un conjunto de niveles definidos por el W3C que se corresponden con las diferentes versiones de este lenguaje que se han ido estableciendo por este organismo.



Recuerde

El W3C es una comunidad internacional que trabaja para crear estándares para la WWW.

El **nivel 0** del lenguaje HTML define la estructura básica de un documento web, en la que existe una etiqueta principal: `<html>`. Dentro de esta, se establecen dos secciones principales: `<head>` y `<body>`. En este nivel, queda definido el contenido la etiqueta `<head>` y gran parte del la etiqueta `<body>` (estilos lógicos, listas, enlaces, imágenes, cabeceras y el conjunto de caracteres especiales).

El siguiente nivel, la **versión 1.0** (nivel 1) añade algunos estilos físicos y lógicos. Esta definición del lenguaje resultó ser insuficiente, por lo que fue necesario añadir ciertos comandos que mejoraran el aspecto de los documentos web.



Recuerde

En el estudio de los lenguajes de marcas, es importante conocer y diferenciar los conceptos de estructura lógica y física de un documento.

La estructura lógica de un documento está formada por las partes y relaciones que lo constituyen. Por otro lado, la estructura física indica el aspecto del documento, ya sea sobre el papel o en la pantalla donde se visualice, incluyendo: sus componentes físicos, el lugar donde se colocan sus elementos y la tipografía de los textos que contiene.

Posteriormente, apareció la **versión 2.0**, que implementaba el nivel 2 de este lenguaje de marcas. La principal novedad de esta versión consistía en la inclusión de un nuevo elemento: los formularios.

Más adelante apareció el nivel 3, conocido como **HTML 3.0**. Esta nueva versión se desarrolló con el objetivo de evolucionar, desde el punto de vista lógico, el estándar. No obstante, la aparición del primer navegador de Netscape hizo necesaria la definición de un nuevo estándar, ya que dicho navegador incorporaba una gran variedad de mejoras especiales. La versión **3.2** se encargó de legalizar la mayoría de estas mejoras, incorporándolas formalmente al modelo propuesto por el W3C.

La versión **HTML 4.0** se propuso por primera vez en 1997 y trató de normalizar el lenguaje y añadirle una serie de mejoras propuestas por los fabricantes de navegadores. Este estándar fue modificado ligeramente en 1999, adoptando el nombre de **HTML 4.01**. Durante casi una década, este fue el último estándar del lenguaje HTML.

Desde que se publicó HTML 4.01, el trabajo de estandarización de HTML se detuvo y el W3C se dedicó a desarrollar el estándar XHTML. La primera versión de este se denominó **XHTML 1.0** y fue publicada a principios del año 2000, revisándose en agosto de 2002.

XHTML 1.0 surgió como una adaptación de HTML 4.01 al lenguaje XML, por lo que conserva casi todas las etiquetas y características del primero, aunque añadiendo ciertas restricciones y elementos del segundo. El borrador de la versión **XHTML 1.1** ya ha sido publicado (y también el de **XHTML 2.0**) y su objetivo fundamental es la modularización de XML 1.0.



Recuerde

Durante mucho tiempo, el XHTML se presentó como una evolución de HTML, que se debería abandonar en el futuro. No obstante, para la inmensa mayoría de los desarrolladores de páginas web, suponía un esfuerzo considerable adaptar lo que se escribía en HTML a XHTML, por lo que, cada vez más, iban abandonando su uso. Por este motivo, el W3C aceptó, con el tiempo, que el lenguaje XHTML pudiera coexistir con HTML.

Después de varios años sin actividad, se constituyó (en el año 2004), al margen del W3C, el WHATWG (*Web Hypertext Application Technology Working Group*), el cual estaba promovido por empresas tales como Apple, Opera, Google o la fundación Mozilla. El propósito de este grupo era el desarrollo de una nueva versión del estándar, enfocándola fundamentalmente desde un punto de vista práctico y no académico (como hasta ahora). Esta versión se conocería como **HTML5**.

En el año 2007, el W3C reconoció este trabajo y lo utilizó como base para su propia actividad. El WHATWG publicó la última versión a finales del año 2009.

El nuevo estándar no cambia el lenguaje en lo esencial, aunque sí que incorpora recursos para facilitar el trabajo con las nuevas herramientas de gestión de contenidos tales como blogs, agregadores, páginas personales, etc., y con la inclusión de elementos

multimedia tales como vídeo y audio.



Uno de los principales objetivos que se han tenido en cuenta en el desarrollo de HTML5 ha sido el de solucionar problemas de tipo práctico, lo cual explica el esfuerzo en orientar esta versión a hacer más fácil el trabajo en situaciones reales.

Otra de las grandes novedades que incorpora el nuevo estándar HTML5 es la inclusión en el estándar del DOM, que siempre había sido tratado de forma separada.



Definición

DOM (Document Object Model)

Estructura de objetos que genera el navegador en el momento en el que carga un documento web. Esta estructura se puede alterar (por ejemplo, mediante JavaScript) con el propósito de modificar dinámicamente los contenidos y la apariencia de la página en cuestión.

Es importante tener en cuenta que no todos los navegadores actuales soportan todos los elementos y funcionalidades nuevas definidas en HTML5. No obstante, día a día, los principales fabricantes de navegadores van mejorando la compatibilidad de sus productos con este nuevo estándar.

3. Estructura de un documento

A continuación, se muestra un fragmento de código escrito en HTML. Si bien está incompleto, permite comprender la estructura básica de los documentos HTML:

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

3.1. Versiones

La línea de código que se muestra a continuación es lo primero que hay que incluir en cualquier documento escrito en HTML5, la cual informa al navegador acerca del tipo y versión de documento que se está creando:

```
<!DOCTYPE html>
```

Esta etiqueta es un comentario especial que no se visualiza en el navegador y debe escribirse sin espacios ni líneas que la precedan. De esta manera, el modo estándar del navegador es activado y las etiquetas y demás novedades propias del nuevo estándar HTML5 son interpretadas siempre que el navegador sea capaz de reconocerlas. En caso contrario, son ignoradas.



Sabía que...

La simplificación de la etiqueta DOCTYPE ha sido uno de los cambios del estándar HTML5. En la versión anterior, la sintaxis de esta directiva era mucho más compleja y su contenido era ignorado por la mayoría de navegadores.

3.2. Cabecera

Después de la etiqueta `DOCTYPE`, es necesario incluir la directiva `<html>`, dentro de la cual se incluyen dos grandes bloques: la **cabecera** y el **cuerpo**.

La etiqueta de cabecera (`<head>`) tiene la siguiente sintaxis:



El propósito y la sintaxis de esta etiqueta no han sufrido variaciones con respecto a las versiones anteriores del estándar. El código HTML que se incluye dentro de esta directiva va a permitir, fundamentalmente:

- Definir el título del documento web.
- Declarar la codificación de caracteres a utilizar.
- Indicar información relacionada con el documento en sí.
- Incorporar archivos externos con estilos CSS y código JavaScript.
- Escribir código JavaScript y CSS.

Todo esto se consigue insertando y configurando una serie de etiquetas. A continuación, se estudiará cuáles son, para qué sirven y cómo se utilizan.



Nota

Excepto el título del documento (y algunos iconos), casi toda la información que se incluye dentro de la directiva `<head>` no es visualizada por el usuario.

Elementos de cabecera

Algunas de las etiquetas más importantes que se suelen incluir dentro de la cabecera de los documentos HTML son: `<meta>`, `<title>` y `<link>`.

<meta>

Esta etiqueta se puede utilizar para definir el juego de caracteres a utilizar en la visualización del documento. En el caso de que se desee asegurar la compatibilidad del documento con los caracteres propios de nuestro idioma (acentos, ñ, etc.) y de los de Europa Occidental, será necesario escribir el siguiente código:

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

La etiqueta `<meta>` también se suele utilizar para otros propósitos, que se pueden intuir a la vista de los principales atributos que suelen formar parte de esta directiva:

- `name`: indica un tipo de información (metainformación) que se desea incluir, la cual está relacionada con el documento en sí:
 - `description`: indica que la información se refiere a los contenidos más relevantes de la página.
 - `keywords`: indica que la información se refiere a un conjunto de palabras clave a utilizar por los buscadores.
 - `author`: indica que la información se refiere al nombre del autor de la página.
 - `generator`: indica que la información se refiere al nombre de una o varias aplicaciones informáticas que se han utilizado en el desarrollo de la web.
- `content`: en este atributo se escribe la metainformación propiamente dicha.
- `http-equiv`: este atributo se usa para especificar algún tipo de instrucción a tener en cuenta por el navegador.



Recuerde

Un atributo es un conjunto formado por un nombre y un valor que se localiza dentro de la etiqueta de inicio de un elemento. Este par (nombre-valor) indica alguna propiedad asociada al elemento en cuestión.



Ejemplo

Si una página web tiene las siguientes palabras clave: fútbol, deporte, campeonato, liga; la sintaxis de la etiqueta `<meta>` que sería necesaria implementar para incluir esta metainformación sería:

```
<meta name="keywords" content="fútbol, deporte, campeonato, liga"/>
```



Actividades

1. Escriba la sintaxis de la etiqueta `<meta>` que se debería implementar si se está realizando una web que trate acerca de “La electrónica digital y analógica”.
2. Identifique y corrija los errores del siguiente fragmento de código HTML:

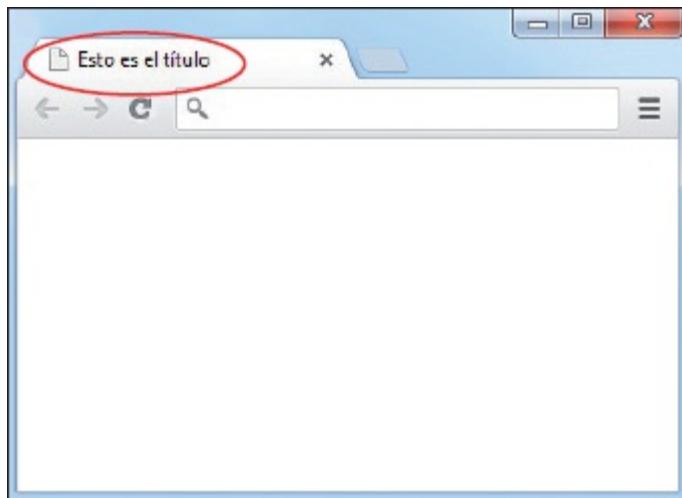
```
<meta char="UTF8"></meta>
```

`<title>`

Dentro de la etiqueta `<title>` se escribe el título de un documento web. Su sintaxis es:

```
<title>...</title>
```

Por lo general, el título de un documento web aparece en la parte superior de la ventana o pestaña en la que se está visualizando la página. Por ejemplo, si un documento web tiene como título el siguiente código: <title>Esto es el título</title>. Dicho título se visualizaría de la siguiente manera en el navegador *Google Chrome*:



Visualización del título de un documento HTML vacío con el navegador Google Chrome

<link>

La etiqueta <link> es otra de las directivas que más se suelen incluir dentro de las cabeceras de los documentos web. Esta etiqueta se utiliza para incorporar, al documento web en cuestión, archivos externos, usándose sobre todo para ajuntar hojas de estilo CSS. Para este caso, la sintaxis de esta etiqueta es la siguiente:

```
<link rel="stylesheet" href="..."/>
```

Con el atributo `rel` se especifica el tipo de relación que existe entre el documento web y el archivo externo que se está incorporando, cuya localización se especifica en el atributo `href`. Si el archivo de estilos se encuentra en el mismo directorio que el documento HTML correspondiente, el valor del atributo `href` coincidirá con el nombre del archivo ".css" (esto se entenderá mejor más adelante).

Hojas de estilo CSS (Cascading Style Sheet o Hoja de Estilos en Cascada)

Archivo que contiene un conjunto de reglas que definen el aspecto de uno o varios de los elementos que constituyen a un documento o sitio web completo.



La aparición de las hojas de estilo en cascada fue un avance fundamental en el mundo del diseño y desarrollo web profesional.

Por ejemplo, si se desea incorporar una hoja de estilos llamada “estilos.css”, la cual se encuentra en el mismo directorio que el documento HTML en cuestión, la sintaxis de la etiqueta <link> sería:

```
<link rel="stylesheet" href="estilos.css"/>
```

**Nota**

Antes de la aparición de HTML5, se hacía necesario añadir un atributo adicional a la etiqueta <link>: el atributo `type`, el cual se utilizaba para indicar el tipo (MIME) de fichero que estaba incorporando. En el caso de las hojas de estilo, este atributo debía tener el valor `text/css`.

Aunque es recomendable el uso de archivos externos, esta no es la única manera de incluir código CSS en un documento web. Dicho código también puede escribirse (utilizando la etiqueta <style>) dentro del propio documento HTML.



Aplicación práctica

Está desarrollando un documento HTML (“index.html”) y desea incluir la siguiente información en la cabecera de la página:

- La temática de la web: “La dieta mediterránea”.
- Palabras clave asociadas a la web: “cocina”, “dieta”, “mediterránea”, “comida”, “recetas”.
- Desea incorporar una hoja de estilos llamada “styles.css”, la cual se encuentra en el mismo directorio que “index.html”.
- El título del documento será: “Comida sana”.

Escriba el fragmento de código HTML correspondiente a la cabecera de su documento web teniendo en cuenta que este debe ser compatible con los caracteres propios de Europa Occidental.

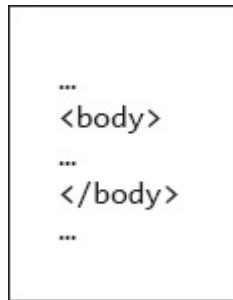
SOLUCIÓN

```
...
<head>
    <title>Comida sana</title>
    <meta charset="UTF-8"/>
    <link rel="stylesheet" href="styles.css"/>
    <meta name="keywords" content="cocina, dieta, mediterránea,
    comida, recetas"/>
    <meta name="description" content="La dieta mediterránea"/>
</head>
...
```

3.3. Cuerpo

El cuerpo es la parte principal donde se organiza el contenido de un documento web. Por lo general, los elementos que se incluyan dentro de esta etiqueta van a tener su correspondiente representación en el área de visualización del navegador. La etiqueta

que se utiliza para albergar el contenido del cuerpo de los documentos web es `<body>` y su sintaxis tampoco ha cambiado en la nueva especificación HTML5:



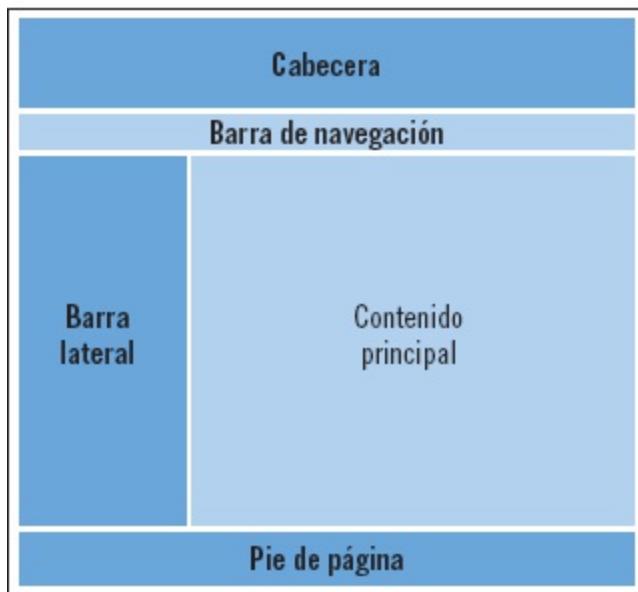
Elementos del cuerpo del documento

Como se acaba de comentar, el cuerpo de un documento HTML es el lugar donde se incluyen cada unos de los elementos que se deseen insertar en una página web, por lo que su contenido puede ser extremadamente variado: tablas, enlaces, párrafos, formularios, imágenes, vídeo, audio y listas.

Cuando se incluyen elementos en el cuerpo de un documento de este tipo, es muy importante organizar y delimitar las zonas donde se situarán cada uno de ellos. Antes de la aparición de HTML5, se solían utilizar tablas (`<table>`) y capas (`<div>`) para estructurar el contenido de las páginas web donde, dentro de directivas de este tipo, se iban colocando los diferentes elementos del cuerpo de dichos documentos. No obstante, la nueva especificación HTML5 ha traído consigo una serie de etiquetas que sirven para estructurar el contenido de los documentos web. Estas etiquetas consisten en capas o áreas dentro de las cuales se incluye el contenido que corresponda.

A continuación, se muestra un esquema en el que se delimitan una serie de secciones o bloques que representan la estructura del contenido de un ejemplo de página web muy genérica. Esto permitirá entender mejor el significado y la utilidad de las nuevas directivas anteriormente mencionadas:

**Representación visual de un ejemplo de diseño
de página web muy genérico y habitual**



La **cabecera** es la zona donde normalmente se localizan el título y el logo de la página web. También puede contener información relacionada con la descripción de la página.



Ejemplo de cabecera de una página web

Justo debajo de la cabecera, se suele incluir la **barra de navegación**, que suele consistir en un menú con enlaces que permiten la navegación a través de las principales secciones que forman parte de la web en cuestión.



Ejemplo de barra de navegación de una página web

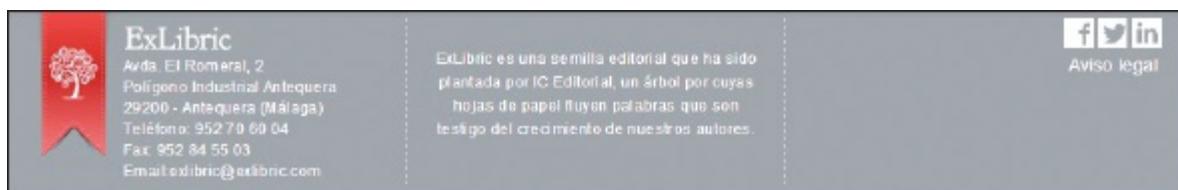
El **contenido principal** de una página web se suele incluir en su centro y puede contener texto, imágenes, enlaces y demás contenido relevante. Generalmente, esta

información se suele estructurar, a su vez, en varias filas y/o columnas.



Ejemplo de distribución del contenido principal de una web en dos columnas

Por último, en la inmensa mayoría de las páginas web se incluye una barra o sección de **pie de página**, que se suele localizar en la parte baja de la web. Por lo general, en esta zona se muestra información relacionada con el sitio web en cuestión, su autor o la empresa desarrolladora, vínculos que informan acerca de las condiciones de uso, *copyright*, etc.



Ejemplo de pie de página de una web

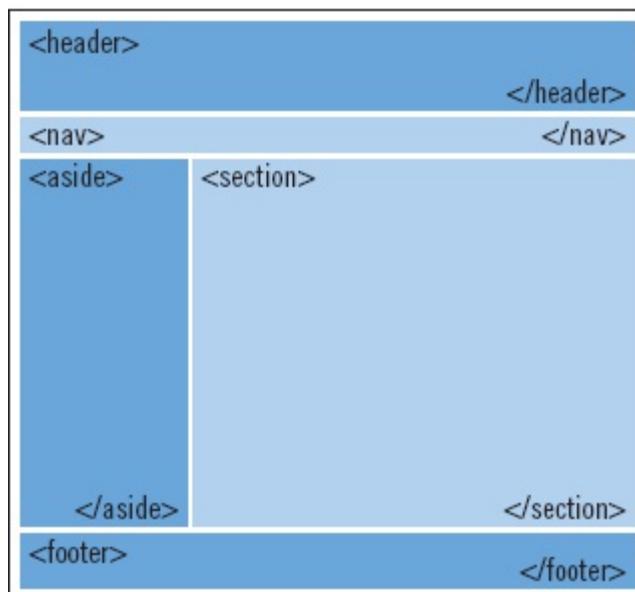


Actividades

3. Acceda a cualquier web e identifique en ella los bloques principales que delimitan y organizan su contenido.

Como se ha comentado anteriormente, la nueva especificación HTML5 añade un conjunto de etiquetas que sirven para delimitar las secciones principales de un documento web (cabecera, barra de navegación, pie de página, etc.). Estas son: `<header>`, `<nav>`, `<section>`, `<aside>` y `<footer>`. A continuación, se muestra el esquema-ejemplo de las secciones que constituyen la estructura del ejemplo de página presentado antes pero, sobre él, se han indicado las etiquetas (y su sintaxis) que se utilizan, en HTML5, para delimitar cada uno de sus bloques:

Etiquetas utilizadas en HTML5 para albergar las secciones principales de una página web



A continuación, se describe brevemente el significado y la utilidad de cada una de estas etiquetas:

- `<header>`: dentro de esta etiqueta (no confundir con `<head>`), se incluye la cabecera del contenido de un documento web. Esta directiva no puede incluirse dentro de `<footer>` u otra `<header>`.
- `<nav>`: esta etiqueta se usa para albergar los elementos de navegación de un

documento HTML.

- <aside>: esta directiva se usa para delimitar un área de un documento web en el que se deseé añadir algo que complemente o aclare el texto y demás contenido principal al que hace referencia.
- <section>: con la etiqueta <section> se puede delimitar un área dentro de un documento HTML.
- <footer>: dentro de esta etiqueta, se incluye el contenido que se desea añadir al pie de página.



Nota

Para posicionar y dar estilo a cada uno de estos bloques o secciones, es imprescindible utilizar CSS.

4. Color

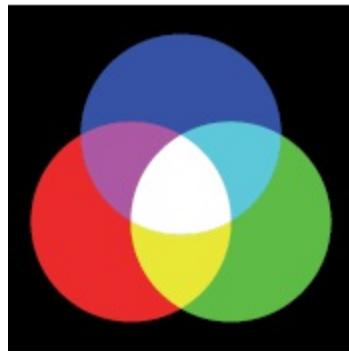
CSS permite, entre otras muchas cosas, modificar y definir el color de los elementos que están presentes en los documentos web, siendo posible establecer: el color del texto que contienen, el de su fondo, el del borde que los delimita, etc.; esto proporciona infinidad de posibilidades de diseño.

4.1. Codificación de colores

Cuando se define un color, siempre se indica a partir de una combinación de unos colores básicos que dependen del sistema de codificación de colores que se esté utilizando. Hay dos sistemas de codificación de colores fundamentales: el aditivo (RGB) y el sustractivo (CMY).

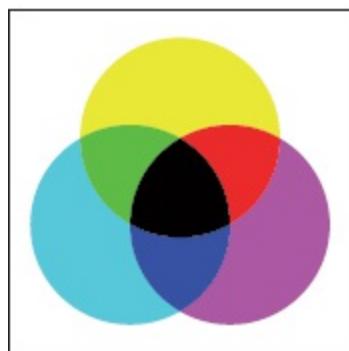
El sistema RGB usa tres colores básicos: el rojo, el verde y el azul; se trata de un sistema aditivo, es decir, si se van sumando colores, se van obteniendo otros cada vez más claros hasta llegar, finalmente, al blanco.

Esquema del modelo RGB



Por otro lado, el sistema CMY usa el cian, el magenta y el amarillo, y es un sistema sustractivo, es decir, cada vez que se van agregando los colores se van obteniendo tonos más oscuros, hasta llegar al negro.

Esquema del modelo CMY



El sistema de codificación que se utiliza en HTML es el RGB. Existen 256 posibles valores para cada uno de los colores básicos, lo cual hace que se puedan obtener 16.777.216 combinaciones diferentes de colores. A la hora de especificar la cantidad de cada color básico, se usa la notación hexadecimal, pudiendo tener un valor comprendido entre 00 (ausencia de ese color) y FF (máxima cantidad de ese color). Los colores en HTML se escriben según la siguiente sintaxis:

```
#RRGGBB
```

Donde:

- #: carácter que indica que el color está en formato RGB.
- RR: cantidad de color rojo.
- GG: cantidad de color verde.
- BB: cantidad de color azul.



Ejemplo

Mediante el uso del atributo `style`, se pueden definir reglas CSS que afecten al elemento en el que se incluye el atributo. Por ejemplo, con el siguiente código HTML se establece un párrafo (`<p>`) de color azul:

```
...  
<p style="color:#0000FF">Esto es un párrafo</p>  
...
```

4.2. Colores tipo

Además de la notación hexadecimal, en HTML existen algunos colores que pueden escribirse con un nombre, el cual coincide con su denominación en inglés. A continuación, se muestra una tabla en la que se relacionan algunos de los colores más utilizados con su denominación en inglés y su notación hexadecimal.

Notación hexadecimal y RGB de algunos de los colores más utilizados

Color	Hexadecimal	Nombre	Color	Hexadecimal	Nombre
	#FFFFFF	Blanco		#800000	Marrón
	#000000	Negro		#800080	Púrpura

	#000080	Marino		#808000	Oliva
	#0000FF	Azul		#808080	Gris
	#008000	Verde		#C0C0C0	Plata
	#008080	Verde azulado		#FF0000	Rojo
	#00FF00	Lima		#FF00FF	Fucsia
	#00FFFF	Agua		#FFFF00	Amarillo



Ejemplo

El siguiente código es equivalente al del ejemplo anterior. Como se puede apreciar, la diferencia está en la notación utilizada para referenciar el color azul:

```
...
<p style="color:blue">Esto es un párrafo</p>
...
```



Actividades

4. Indique si los siguientes fragmentos de código son equivalentes. Razone la respuesta:

```
<p style="color:#FF00FF">Esto es un párrafo con color</p>
<p style="color:yellow">Esto es un párrafo con color</p>
```

4.3. Colores seguros

Los colores seguros para la Web son los que se muestran de la misma manera en todos los navegadores y sistemas operativos.

En total, existen 216 colores seguros y la notación en RGB que pueden presentar viene dada por un valor hexadecimal que combina los pares 00, 33, 66, 99, CC o FF.

Cuando aparecieron los primeros navegadores, la mayoría de los equipos únicamente eran capaces de mostrar 265 colores. Hoy en día, casi todos los equipos pueden mostrar miles o millones de colores.

Esto último hace que, en la actualidad, no esté tan justificado el único uso de colores seguros en el diseño de un sitio web.



Actividades

5. ¿Es el color #CCAAFF un color seguro para la web? Razone su respuesta.
6. Indique si en la siguiente línea de código se está utilizando un color seguro.

Razone su respuesta:

```
<div style="color:lime">...</div>
```

A continuación, se muestra una tabla en la que se puede observar el aspecto y el código hexadecimal de los 216 colores seguros para la web.

Tabla de los 216 colores seguros para la web

#000000	#000033	#000066	#000099	#0000CC	#0000FF
#003300	#003333	#003366	#003399	#0033CC	#0033FF
#006600	#006633	#006666	#006699	#0066CC	#0066FF
#009900	#009933	#009966	#009999	#0099CC	#0099FF
#00CC00	#00CC33	#00CC66	#00CC99	#00CCCC	#00CCFF
#00FF00	#00FF33	#00FF66	#00FF99	#00FFCC	#00FFFF
#330000	#330033	#330066	#330099	#3300CC	#3300FF
#333300	#333333	#333366	#333399	#3333CC	#3333FF
#336600	#336633	#336666	#336699	#3366CC	#3366FF
#339900	#339933	#339966	#339999	#3399CC	#3399FF
#33CC00	#33CC33	#33CC66	#33CC99	#33CCCC	#33CCFF
#33FF00	#33FF33	#33FF66	#33FF99	#33FFCC	#33FFFF
#660000	#660033	#660066	#660099	#6600CC	#6600FF
#660033	#663333	#663366	#663399	#6633CC	#6633FF
#666600	#666633	#666666	#666699	#6666CC	#6666FF
#669900	#669933	#669966	#669999	#6699CC	#6699FF
#66CC00	#66CC33	#66CC66	#66CC99	#66CCCC	#66CCFF
#66FF00	#66FF33	#66FF66	#66FF99	#66FFCC	#66FFFF
#990000	#990033	#990066	#990099	#9900CC	#9900FF
#993300	#993333	#993366	#993399	#9933CC	#9933FF
#996600	#996633	#996666	#996699	#9966CC	#9966FF
#999900	#999933	#999966	#999999	#9999CC	#9999FF
#99CC00	#99CC33	#99CC66	#99CC99	#99CCCC	#99CCFF
#99FF00	#99FF33	#99FF66	#99FF99	#99FFCC	#99FFFF
#CC0000	#CC0033	#CC0066	#CC0099	#CC00CC	#CC00FF
#CC3300	#CC3333	#CC3366	#CC3399	#CC33CC	#CC33FF
#CC6600	#CC6633	#CC6666	#CC6699	#CC66CC	#CC66FF
#CC9900	#CC9933	#CC9966	#CC9999	#CC99CC	#CC99FF
#CCCC00	#CCCC33	#CCCC66	#CCCC99	#CCCCCC	#CCCCFF
#CCFF00	#CCFF33	#CCFF66	#CCFF99	#CCFFCC	#CCFFFF
#FF0000	#FF0033	#FF0066	#FF0099	#FF00CC	#FF00FF
#FF3300	#FF3333	#FF3366	#FF3399	#FF33CC	#FF33FF
#FF6600	#FF6633	#FF6666	#FF6699	#FF66CC	#FF66FF
#FF9900	#FF9933	#FF9966	#FF9999	#FF99CC	#FF99FF
#FFCC00	#FFCC33	#FFCC66	#FFCC99	#FFCCCC	#FFCCFF
#FFFF00	#FFFF33	#FFFF66	#FFFF99	#FFFFCC	#FFFFFF

5. Texto

Para organizar, estructurar y dar estilo al texto de un documento HTML, es importante que este se vaya insertando dentro de una serie de etiquetas. En este apartado se estudiarán algunas de las más importantes.



Nota

El texto de un documento HTML también puede ir encerrado dentro de etiquetas pertenecientes a otros elementos web, tales como tablas, formularios, enlaces, etc.

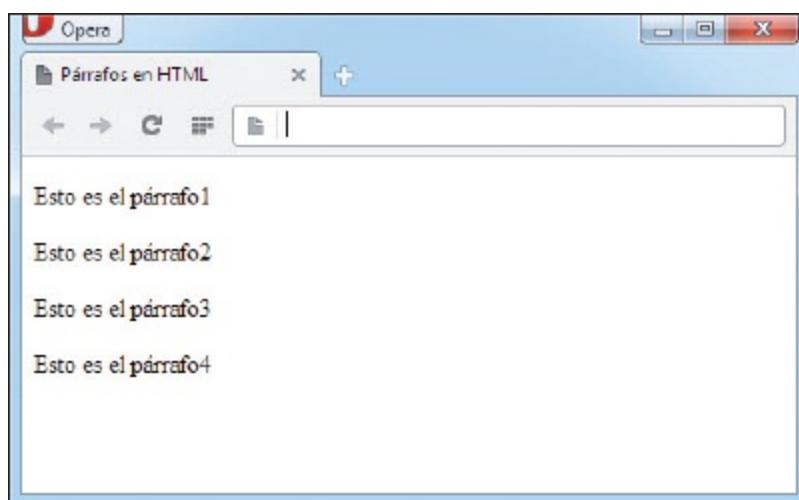
5.1. Párrafos

Los párrafos en HTML son unos de los elementos básicos de texto que forman parte de este lenguaje y se suelen utilizar para insertar texto sin significado especial. Su sintaxis es muy sencilla, utilizándose únicamente la etiqueta `<p>` para albergar el texto perteneciente a cada párrafo. A continuación, se muestra un ejemplo de párrafo en HTML:

```
<p>Esto es un párrafo</p>
```

Por defecto, cada vez que se escribe un párrafo, el navegador inserta un salto de línea automático al final de cada uno ellos, por lo que, siempre que se escriba un elemento de este tipo, este va a aparecer en una línea nueva. Por ejemplo, un documento web que contenga en su cuerpo únicamente el siguiente código, se visualizaría como se indica más abajo:

```
...
<p>Esto es el párrafo1</p>
<p>Esto es el párrafo2</p>
<p>Esto es el párrafo3</p>
<p>Esto es el párrafo4</p>
...
```



Visualización de 4 párrafos seguidos en HTML



Nota

En este manual, se presentan muchos ejemplos de código HTML en el que aparecen líneas punteadas (...), normalmente en la parte superior e inferior de los ejemplos.

Esto significa que, para que estas líneas de código sean interpretadas correctamente por los navegadores, debe añadirse más código HTML (`doctype`, `<head>`, `<body>`, etc.) en las zonas donde aparecen las líneas puntuadas.

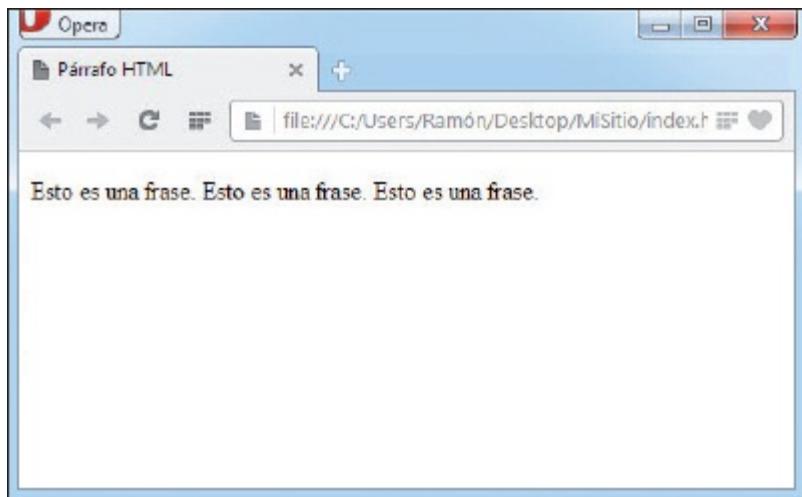
Esto se hace así en aquellos ejemplos en los que no resulte relevante mostrar este código adicional.



Importante

Los saltos de línea que se hacen dentro del propio código HTML no son interpretados como tales por el navegador. Por ejemplo, el siguiente fragmento de código se visualizaría sin saltos de línea (ver más abajo):

```
...
<p>
Esto es una frase.
Esto es una frase.
Esto es una frase.
</p>
...
```



Visualización de un párrafo con saltos de línea insertados en el propio código



Actividades

7. Describa el resultado que se visualizará al escribir la siguiente línea de

código en un documento HTML:

```
...  
<p>Hola mundo</p>Hola mundo</p><p>Hola mundo</p></p>
```

```
...
```

5.2. Encabezados. Jerarquía y estructura del contenido de un documento

Para establecer importancia en los textos de una página web (títulos, subtítulos, etc.), se pueden utilizar las etiquetas de encabezado. Estas son: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`. La primera etiqueta es la que, por defecto, establece más importancia en el texto, seguida de `<h2>` y así sucesivamente. Lo que hacen estas etiquetas es cambiar el estilo del texto que contienen, modificando ciertas características del mismo para destacarlo y diferenciarlo de los demás.

Estas directivas son muy similares a la etiqueta `<p>`, ya que, cuando se insertan, el navegador fuerza un salto de línea antes y después del texto que contienen.



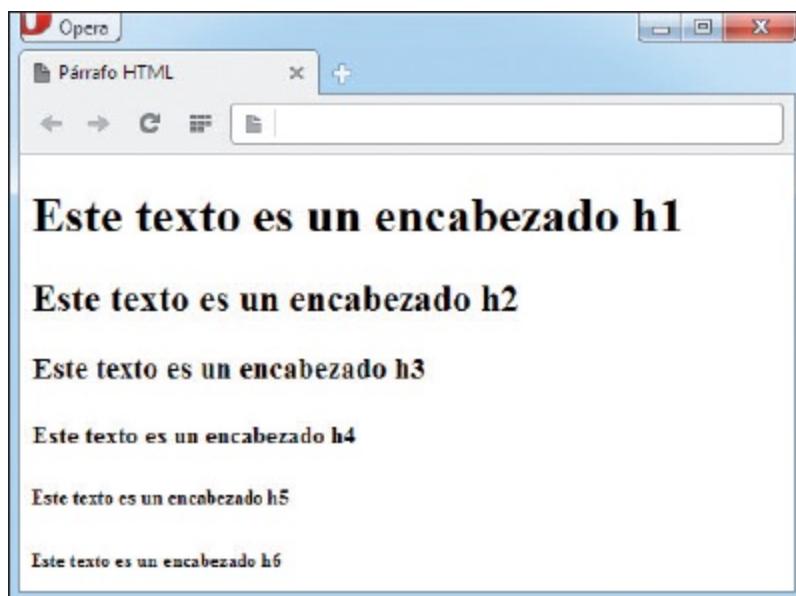
Sabía que...

El estilo que por defecto establecen las etiquetas de encabezado puede modificarse fácilmente utilizando CSS.

A continuación, se muestra un ejemplo de uso de cada una de estas etiquetas en HTML:

```
...
<h1>Este texto es un encabezado h1</h1>
<h2>Este texto es un encabezado h2</h2>
<h3>Este texto es un encabezado h3</h3>
<h4>Este texto es un encabezado h4</h4>
<h5>Este texto es un encabezado h5</h5>
<h6>Este texto es un encabezado h6</h6>
```

```
...
```



Visualización de los encabezados <h1>, <h2>... <h6>

Aplicación práctica. Ejemplo de documento web con texto

Desea escribir un documento HTML (“index.html”) que contenga únicamente texto. Escriba el código HTML que debe implementar para obtener exactamente el mismo resultado que el que se muestra a continuación.

2.1. Elementos de una instalación automatizada

Los elementos que constituyen una instalación automatizada se pueden clasificar en dos bloques fundamentales: elementos operativos y elementos de mando.

Elementos operativos

Los elementos operativos son capaces de actuar directamente sobre una máquina o bien detectar variaciones en el entorno que condicionarán el funcionamiento de la instalación. Estos elementos pueden hacer que una máquina se mueva o efectúe una acción determinada. Los elementos operativos se dividen en dos grupos: los accionadores o actuadores (motores) y los sensores o captadores (fotodioides, sensores de temperatura, etc.).

Accionadores o actuadores

Un accionador es un dispositivo que, en respuesta a la señal procedente de un elemento de mando, actúa sobre alguna variable de salida, ya que es capaz de transformar la señal que recibe (por lo general, procedente de un automatismo) en un trabajo útil.

Los accionadores suelen estar gobernados por un elemento de mando y pueden requerir de algún preaccionamiento para amplificar la señal que se les suministra. Esta amplificación previa consiste en permitir o interrumpir el suministro de energía desde la fuente al accionador.

Visualización de “index.html”

Consideraciones a tener en cuenta:

- El título del documento HTML será: “Ejemplo de texto”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.
- Para establecer el texto del cuerpo del documento, utilice únicamente las etiquetas `<p>`, `<h1>`, `<h2>` y `<h3>`.

Solución

```

<!DOCTYPE html>
<html>
<head>
    <title>Ejemplo de texto</title>
    <meta charset="UTF-8"/>
</head>
<body>
    <h1>2.1. Elementos de una instalación  
automatizada</h1>
    <p>Los elementos que constituyen una instalación  
automatizada se pueden clasificar en dos bloques  
fundamentales: elementos operativos y elementos de  
mando.</p>
    <h2>Elementos operativos</h2>
    <p>Los elementos operativos son capaces de actuar  
directamente sobre una máquina o bien detectar  
variaciones en el entorno que condicionará el  
funcionamiento de la instalación. Estos elementos  
pueden hacer que una máquina se mueva o efectúe  
una acción determinada. Los elementos operativos se  
dividen en dos grupos: los accionadores o actuadores  
(motores) y los sensores o captadores (fotodiodos,  
sensores de temperatura, etc.).</p>
    <h3>Accionadores o actuadores</h3>
    <p>Un accionador es un dispositivo que, en  
respuesta a la señal procedente de un elemento de  
mando, actúa sobre alguna variable de salida, ya que  
es capaz de transformar la señal que recibe (por lo  
general, procedente de un automatismo) en un  
trabajo útil.</p>
    <p>Los accionadores suelen estar gobernados por un  
elemento de mando y pueden requerir de algún  
preaccionamiento para amplificar la señal que se les  
suministra. Esta amplificación previa consiste en  
permitir o interrumpir el suministro de energía desde  
la fuente al accionador.</p>
</body>
</html>

```

3. Estilos del texto

La nueva especificación HTML5 recomienda que, para modificar la apariencia del texto y demás elementos que puedan formar parte de una página web, se utilicen hojas

de estilo CSS.

No obstante, existe una etiqueta llamada `` que permite establecer diferentes características en el texto, como pueden ser su tamaño, el tipo de fuente y su color.

Esta etiqueta no es soportada por la nueva especificación del estándar, pero todavía sigue siendo interpretada por los principales navegadores web.

Características de letra: tipos, tamaño y colores

La etiqueta `` soporta (o soportaba) principalmente los siguientes atributos:

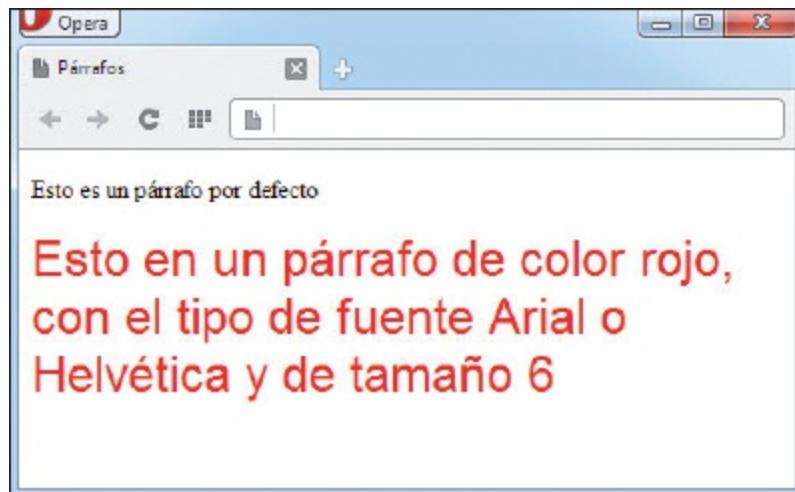
- `face`: el valor de este atributo indica el tipo de fuente que va a tener el texto encerrado en la correspondiente etiqueta ``. Es muy recomendable indicar más de una familia de fuentes (separadas por comas) por si el navegador no reconoce alguna de ellas, aplicándose la primera que esté disponible en la máquina del usuario.
- `color`: con este atributo se puede modificar el color con el que se va a ver un texto. Como se ha visto anteriormente, el color puede indicarse de dos maneras: con su nombre en inglés (*black, green, white, etc.*) y con su valor RGB (#AA00CC, #00222F, etc.).
- `size`: este atributo sirve para establecer el tamaño de la fuente del texto en cuestión y puede tener valores enteros comprendidos entre 1 (el más pequeño) y 7 (el más grande).



Ejemplo

A continuación, se muestra un fragmento de código HTML con dos párrafos. El aspecto del primero es el establecido por defecto. En el segundo, se ha modificado su apariencia con la etiqueta ``.

```
...
<p>Esto es un párrafo por defecto</p>
<p><font face="Arial, Helvetica" color="#FF0000"
size="6">Esto es un párrafo de color rojo, con el tipo
de fuente Arial o Helvética y de tamaño 6</font></p>
...
```



Visualización de los párrafos en el navegador Opera

A continuación, se muestra un fragmento de código HTML con dos párrafos. El aspecto del primero es el establecido por defecto. En el segundo, se ha modificado su apariencia con la etiqueta ``.

```
...
<p>Esto es un párrafo por defecto</p>
<p style="font-family:Arial,
hevetica;color:#FF0000;font-size:6px"></p>
...
```



Actividades

8. Modifique el código anterior para establecer las siguientes características en el primer párrafo:
- Tipo de fuente: Times.
 - Color: Verde.
 - Tamaño: 2.
9. Identifique y corrija, en el siguiente fragmento de código, los errores que se detecten:

```
<font color="AABBCC" size="small">Esto es un texto</font>
```

Alineación, espaciado y sangrado del texto

Además de poder modificar el tamaño, el tipo de fuente y el color del texto, en HTML también se pueden definir otras características, como son: la alineación del texto, el espaciado entre palabras y su sangrado.

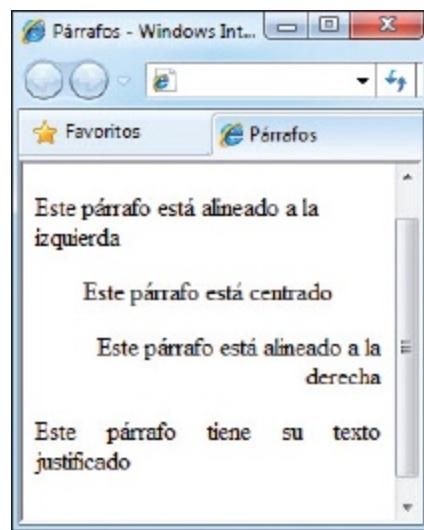
Al igual que ocurre con las primeras, la nueva especificación HTML5 recomienda el uso de hojas de estilo para establecer estas propiedades en los textos de los documentos web.

Existe un atributo, desaprobado por la nueva especificación HTML5, que permitía modificar la alineación del texto contenido dentro de una etiqueta.

Este es `align` y puede tener los siguientes valores: `center` (centro), `left` (izquierda), `right` (derecha) y `justify` (justificado).

A continuación, se muestra un ejemplo de uso de este atributo para modificar la alineación del texto contenido en cuatro párrafos:

```
...
<p align="left">Este párrafo está alineado a la izquierda</p>
<p align="center">Este párrafo está centrado</p>
<p align="right">Este párrafo está alineado a la derecha</p>
<p align="justify">Este párrafo tiene su texto justificado</p>
...
```



Visualización de varios párrafos con distintas alineaciones de texto

Aunque el atributo `align` sigue siendo reconocido por la mayoría de los navegadores actuales, se recomienda que, para alinear textos, se utilicen hojas de estilo. A continuación, se muestra un fragmento de código en el que se usan reglas CSS para obtener el mismo resultado que el del ejemplo anterior:

```
...  


Este párrafo está alineado a la izquierda



Este párrafo está centrado



Este párrafo está alineado a la derecha



Este párrafo tiene su texto justificado



...


```

Gracias a la utilización de CSS se pueden establecer muchas otras opciones de estilo para los textos de los documentos web. Por ejemplo, para modificar el espaciado entre las letras del texto de un párrafo, se puede usar la siguiente sintaxis:

```
<p style="letter-spacing:...">Ejemplo de texto</p>
```



Recuerde

Mediante el uso del atributo `style`, se pueden definir reglas CSS que afecten al elemento en el que se incluye el atributo.

`letter-spacing` es una propiedad CSS que permite modificar el espaciado entre las letras del texto contenido dentro de una etiqueta. Dicho espaciado puede definirse utilizando valores absolutos (píxeles), relativos (em) o bien mediante un texto (`normal`, `length`, `inherit`).

Por otro lado, para establecer sangrado en los textos de las páginas web, se puede utilizar la propiedad CSS `text-indent` y su valor puede indicarse en: píxeles, `em`,

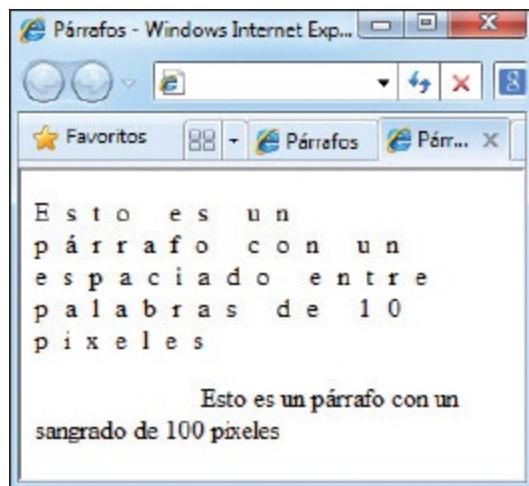
porcentaje, etc.



Ejemplo

A continuación, se muestra un fragmento de código en el que se hace uso de las propiedades CSS letter-spacing y text-indent:

```
...
<p style="letter-spacing:10px">Esto es un párrafo con
un espaciado entre letras de 10 píxeles</p>
<p style="text-indent:100px">Esto es un párrafo con
un sangrado de 100 píxeles</p>
...
```



Visualización de dos párrafos en los que se ha modificado, respectivamente, el espaciado entre letras y su sangrado



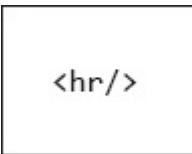
Definición

Sangrado

Espaciado que se suele establecer en la primera línea y antes de la primera palabra del mismo.

Separadores de texto

Para separar párrafos y demás elementos web, se puede incluir una línea horizontal que puede ser modificada, tanto en tamaño como en grosor. Para ello, se puede utilizar la directiva `<hr>`, la cual no necesita de etiqueta de cierre. Su sintaxis, según la especificación XHTML es:

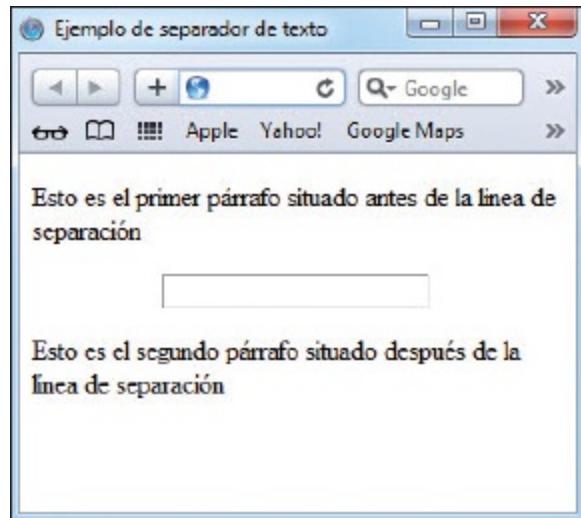


Para modificar las características predeterminadas de esta franja, se pueden utilizar los siguientes atributos (no soportados por la nueva especificación HTML5, pero sí por la mayoría de navegadores):

- `align`: en este atributo, se especifica la alineación del separador.
- `noshade`: cuando se incluye este atributo (sin valor), la franja se visualiza con un color sólido.
- `size`: especifica el alto de la franja.
- `width`: especifica el ancho de la franja.

A continuación, se muestra un ejemplo de uso de la etiqueta `<hr/>` para separar dos párrafos. La línea ocupa la mitad de la pantalla (50%), está alineada en el centro y tiene un ancho de 10 píxeles:

```
...
<p>Esto es el primer párrafo situado antes de la línea de
separación</p>
<hr width="50%" size="20" align="center"/>
<p>Esto es el segundo párrafo situado después de la línea
de separación</p>
...
```



Ejemplo de separador de texto



Actividades

10. Modifique el código anterior para que la línea tenga las siguientes características:
- Ocupará todo el ancho la pantalla.
 - Medirá 5 píxeles de alto.
 - Estará alineada a la derecha.



Actividades

11. ¿Qué errores hay en el siguiente fragmento de código? Razone su respuesta:

```
<hr width="600px"></hr>
```

Etiquetas específicas para el marcado de texto. Estilos lógicos

Para modificar la apariencia texto de los documentos web, se pueden utilizar los denominados estilos físicos y lógicos. Los estilos físicos son aquellos que producen siempre el mismo efecto (negrita, cursiva, etc.).

Por otro lado, los estilos lógicos son aquellos que marcan un tipo de texto determinado (*cita*, *e-mail*, etc.) y hacen que este se muestre de una manera determinada, según el estilo lógico que corresponda.

Como se verá a continuación, se pueden utilizar ambos tipos de estilos para conseguir un mismo efecto.



Ejemplo

Para hacer que un texto aparezca en negrita, se puede utilizar la etiqueta **** (estilo físico) o bien la directiva **** (estilo lógico).

A continuación, se muestra una tabla en la que se enumeran los estilos físicos y lógicos del lenguaje HTML.

Etiqueta	Tipo de estilo	Tipo de efecto / descripción
	Físico	Negrita
<i>	Físico	Cursiva
<tt>	Físico	Texto mecanografiado (Etiqueta no soportada en HTML5)
<sub>	Físico	Subíndice
<sup>	Físico	Superíndice
<big>	Físico	Texto grande (tamaño de fuente más grande) (Etiqueta no soportada en HTML5)
<small>	Físico	Texto pequeño (tamaño de fuente más pequeña)
<blink>	Lógico	Texto parpadeante (Etiqueta no soportada en HTML5)
<address>	Lógico	Especifica direcciones de correo electrónico
<blockquote>	Lógico	Cita textual
<cite>	Lógico	Título de una obra (película, libro, etc.)

<code>	Lógico	Código fuente de un lenguaje de programación
<dfn>	Lógico	Definición
	Lógico	Texto al que se le quiere dar énfasis (por defecto, cursiva)
<kbd>	Lógico	Texto introducido por teclado
<samp>	Lógico	Mensaje de estado de una computadora
	Lógico	Texto tachado
	Lógico	Texto destacado (por defecto, negrita)
<var>	Lógico	Especifica una variable

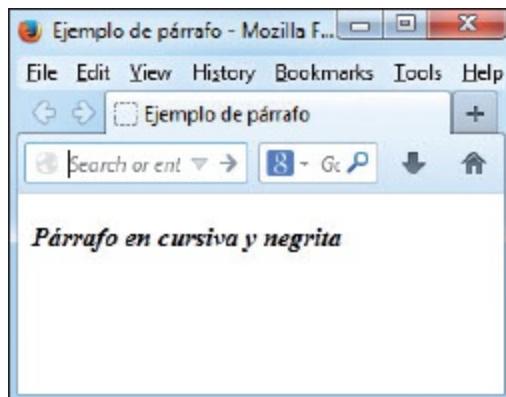


Ejemplo

Un texto puede estar afectado por más de un estilo lógico o físico. En el siguiente ejemplo se aplican los efectos de cursiva y negrita en el texto de un párrafo:

```
...
<p><b><i>Párrafo en cursiva y negrita</i></b></p>
...

```



Visualización de un párrafo en cursiva y negrita

6. Enlaces de hipertexto

Seguramente, la característica más interesante y que más ha influido en el notable desarrollo de la WWW ha sido el denominado hipertexto, que se puede definir (en el ámbito de la WWW) como una herramienta que permite establecer enlaces o vínculos en los documentos web que sean capaces de, entre otras cosas, dirigir al usuario hacia otra sección de la página u otro sitio web externo, con tan solo hacer un simple clic de ratón.

6.1. Estructura de un enlace: la dirección de Internet o URL

Para insertar enlaces en los documentos HTML se utiliza la etiqueta `<a>`. A continuación, se muestra la sintaxis básica de un enlace en HTML cuyo fin es dirigir al usuario a otro sitio, ya sea este una página externa u otra sección del propio sitio web.

```
<a href="...">Texto del enlace</a>
```

El texto o contenido de esta etiqueta será el que, al pulsarlo, permita dirigir al usuario al destino indicado en el correspondiente atributo `href`.



Importante

La funcionalidad de los enlaces no se limita únicamente a dirigir al usuario a otra parte de la página o a un sitio web externo.



Actividades

12. Navegue por alguna página e identifique (si los tiene) los enlaces que dirijan a un sitio web ajeno a la página.

Cuando un enlace se establece para dirigir al usuario a un sitio web externo, es necesario especificar, en el atributo `href` correspondiente, la dirección de Internet o URL donde se aloja la página web en cuestión. Esta dirección tendrá siempre la siguiente sintaxis:

Servicio://maquina.dominio:puerto/camino/fichero

Donde:

- `servicio`: protocolo de Internet, como puede ser: http, ftp, etc.
- `Maquina.dominio`: servidor que proporciona el recurso (por ejemplo: www.google.es).
- `puerto`: esta parte es opcional y, por lo general, no se utiliza.
- `camino`: ruta de directorios que es necesario seguir para acceder al recurso correspondiente. Para separar los distintos subdirectorios, es necesario utilizar el carácter “/” (por ejemplo: noticias/novedades).
- `fichero`: nombre de la página o recurso que se desea abrir, acompañado de su correspondiente extensión (por ejemplo: “index.html”, “fichero.zip”, “image.png”, etc.).



Actividades

13. Identifique, en la siguiente URL, las distintas partes:

<http://www.ejecutivos.es/noticia/34141/Biblioteca-Ejecutivos/Las-recetas-que-adelgazan-de-Marta-Aranzadi.html>

6.2. Atributos específicos: título, destino, atajos de teclado, etc.

La etiqueta `<a>` admite ciertos atributos, algunos de los cuales son específicos para esta directiva. A continuación, se listan los más importantes:

- `href`: aquí se escribe la dirección asociada al enlace.
- `title`: en este atributo se establece un texto que se desea que aparezca junto al enlace en el momento en el que se posicione el ratón sobre el mismo.
- `target`: este atributo se usa para definir el lugar donde se va a abrir el contenido asociado al enlace. Puede tener los siguientes valores:
 - `_blank`: el documento al que apunta el vínculo se abrirá en una ventana/pestaña nueva.
 - `_parent`: el documento de destino se mostrará en el *frameset*, padre del *frame* actual (esto se entenderá mejor cuando se estudie el apartado correspondiente a los marcos).
 - `_self`: el documento de destino se mostrará en la ventana o *frame* donde se ubica el enlace.
 - `_top`: el enlace se mostrará usando todo el espacio de la ventana del navegador.
- `acceskey`: este atributo sirve para definir un atajo de teclado que servirá para acceder directamente al enlace en cuestión. Al pulsar la tecla [Alt] junto con la definida en este atributo, el navegador dirigirá al usuario al lugar donde se encuentre el enlace dentro de la web, pero sin ejecutarlo. Para abrirlo, se puede pulsar la tecla [Intro] (o hacer clic sobre él).



Nota

En el momento de escribir estas líneas, el atributo `acceskey` no es soportado por el navegador *Opera*.



Aplicación práctica

Está desarrollando un documento web (“index.html”) en el que, por el momento, únicamente desea incluir un enlace con las siguientes características:

- **Texto del enlace: “Ir a Google”.**
- **Dirección asociada al enlace: <http://www.google.es>.**
- **Título o texto que se mostrará al posicionar el cursor sobre el mismo:**

- “Este enlace le direccionará a la página de Google”.**
- La página de destino se abrirá en una ventana/pestaña nueva.
 - Si el usuario pulsa la combinación de teclas [Alt+ i], el navegador dirigirá al usuario al sitio donde se encuentre el enlace.

Escriba el código HTML del documento “index.html”, teniendo en cuenta las siguientes consideraciones:

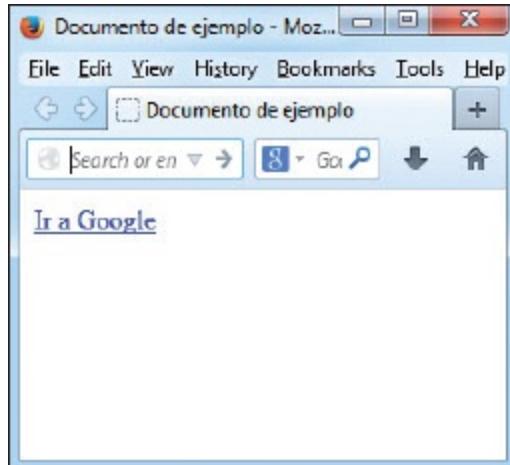
- El título del documento será: “Documento de ejemplo”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.

SOLUCIÓN

```
<!DOCTYPE html>
<html>
    <head>
        <title>Documento de ejemplo</title>
        <meta charset="UTF-8"/>
    </head>
    <body>
        <a href="http://www.google.es" title="Este enlace le direccionará a la página de Google" target="_blank"
            acceskey="i">Ir a Google</a>
    </body>
</html>
```

6.3. Estilos de enlaces

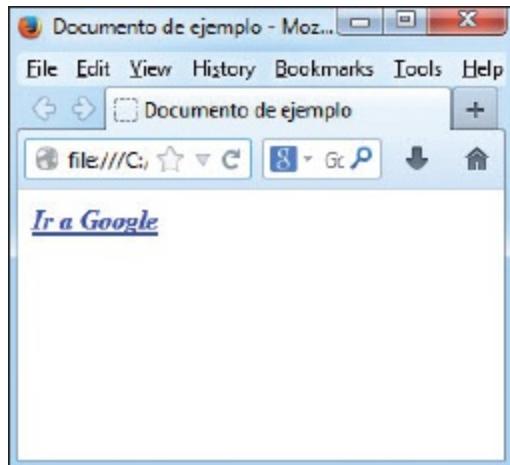
Por defecto, los textos de los enlaces que se escriben en los documentos web se visualizan de color azul y subrayados. A continuación, se muestra cómo se vería el documento correspondiente a la aplicación práctica anterior desde el navegador *Mozilla Firefox*.



Ejemplo de visualización de un enlace en el navegador Mozilla Firefox

Aunque los enlaces se visualicen de esta manera, su aspecto puede ser modificado fácilmente. Lo recomendable es utilizar hojas de estilos CSS, aunque también es posible modificar la apariencia usando etiquetas de estilos lógicos y físicos. A continuación, se muestra un ejemplo de enlace cuyo texto se visualiza en negrita y cursiva:

```
...
<b><i><a href="http://www.google.es" title="Este
enlace le direccionará a la página de Google"
target="_blank" acceskey="i">Ir a Google</a></i></b>
...
...
```



Ejemplo de visualización de un enlace en cursiva y negrita



Ejemplo

A continuación, se muestra un ejemplo de uso de CSS para modificar el aspecto de un enlace en HTML. El texto de dicho link se visualizará en rojo y sin subrayado:

```
...
<a href="http://www.google.es" style="text-decoration:none; color:red">Ir a Google</a>
...
```



Actividades

14. Identifique y corrija los errores en el siguiente fragmento de código HTML:

```
<b><i><a href="www.google.es" title=Enlace/>Texto  
del enlace<a></b></i>
```

6.4. Diferencias entre enlaces absolutos y relativos

Cuando se incluye un enlace en una página web, la dirección asociada al mismo se puede escribir de dos maneras diferentes: utilizando una ruta absoluta, o bien, relativa.

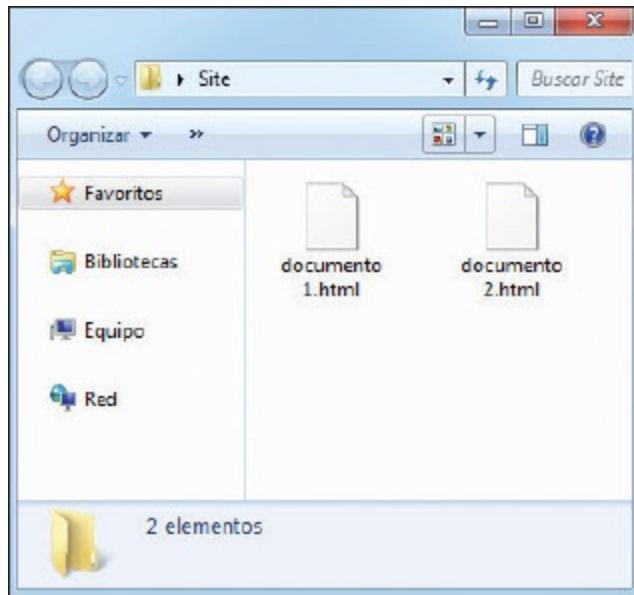
Los **enlaces absolutos** son aquellos que contienen una dirección completa, incluyendo los nombres del servicio, dominio, ruta, etc. A continuación, se muestra un ejemplo de URL absoluta:

http://www.ejemplo_de_servidor.com/carpeta1/carpeta2/index.html

Por otro lado, en los **enlaces relativos** no se incluye la dirección completa del documento o archivo al que apuntan, sino que se omite la información correspondiente a servicio, maquina.dominio y, probablemente también la ruta. Este tipo de enlaces se utilizan para direccionar a documentos o archivos que se encuentren en el mismo servidor que el documento en el que se localiza el enlace en cuestión. A continuación, se muestra un ejemplo que ayudará a entender la sintaxis y el funcionamiento de estos enlaces.

Un sitio web está constituido por dos documentos HTML: “documento1.html” y “documento2.html” y, en este último, se desea incluir un enlace que direccione a “documento1.html”. Ambos documentos se localizan en el mismo directorio, por lo que, para ello, bastaría con poner el nombre del archivo de destino en el atributo href de dicho enlace, ya que ambos ficheros se encuentran en el mismo directorio o nivel:

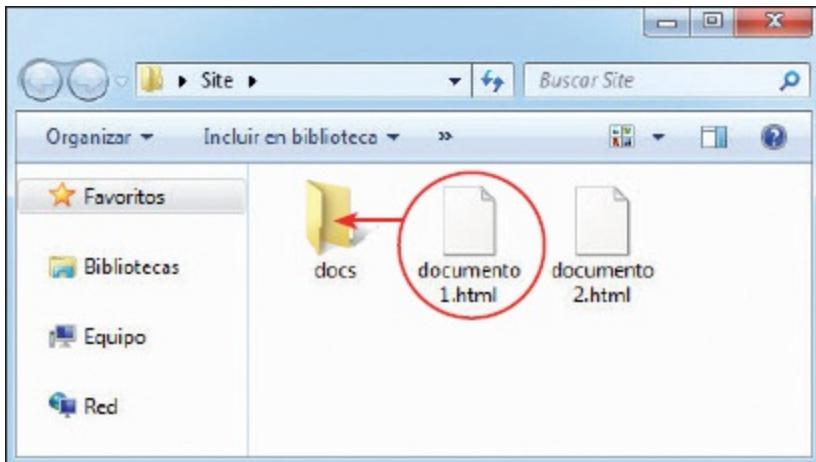
```
... href="documento1.html"...
```



Estructura de archivos de un sitio web sencillo

Por otro lado, si el documento de destino se encontrara dentro de una carpeta llamada "docs" que se ubicara en el mismo directorio que "documento2.html", la dirección que habría que poner en el atributo href sería:

```
... href="docs/documento1.html"...
```



Al incluir, dentro de `docs`, el documento de destino, será necesario añadir el nombre de este directorio a la ruta relativa del atributo `href` del enlace.

También es posible que se quiera dirigir a un fichero que se localice en un nivel superior al del documento HTML que contenga el vínculo. En este caso, sería necesario añadir la cadena “`..`” por cada nivel que separe los archivos de origen y destino.

Por ejemplo, en el caso de que se deseara establecer un enlace en “`documento1.html`” que dirigiera a “`documento2.html`”, estando el primero ubicado en el directorio “`docs`”, la cadena que habría que establecer en el atributo `href` del enlace, sería:

```
... href="..../documento2.html"...
```

En resumen, cuando se establecen rutas relativas, se toma como base la ruta absoluta en la que se encuentra el documento de origen y, a partir de esta, se escribe la ruta de directorios que hay que seguir hasta llegar al destino.



Actividades

15. Siguiendo con el ejemplo del sitio anterior, indique la ruta relativa que

habría que escribir en el atributo `href` de un enlace que se quisiera establecer en “documento1.html” teniendo en cuenta que hay dos niveles o directorios de diferencia entre el origen y el destino (“documento2.html”). Nota: “documento2.html” se encuentra en la raíz o directorio principal del sitio.

6.5. Enlaces internos

Los enlaces internos (también conocidos como anclas) son vínculos que direccionan al usuario hacia otras secciones de un mismo documento HTML.

Un ejemplo muy habitual de este tipo de enlaces puede ser el típico *link* que se establece al final de una página web y que, al seleccionarlo, conduce de nuevo al usuario al principio del documento en cuestión.

En resumen, para crear un enlace interno en HTML, es necesario:

- Referenciar el lugar de destino con algún nombre, utilizando el atributo `name`.
- Escribir, en el atributo `href` del enlace, el nombre que se ha indicado en el paso anterior (destino), pero anteponiéndole el carácter “#”.



Ejemplo

Un documento web contiene dos enlaces, uno al principio y otro al final de la página y se desea que cada uno se comporte de la siguiente manera:

- En enlace superior (Ir abajo) dirigirá hacia donde se encuentra el enlace inferior (Ir arriba).
- En enlace inferior (Ir arriba) dirigirá hacia donde se encuentra el enlace superior (Ir abajo).

A continuación, se muestra un fragmento de código HTML válido para implementar dichas anclas:

```
...
<a href="#abajo" name="arriba">Ir abajo</a>
...
...
<a href="#arriba" name="abajo">Ir arriba</a>
...
```

También es posible establecer enlaces que direccionen a lugares concretos dentro de otros documentos HTML. Para hacer esto, basta con escribir, en el atributo `href` del enlace en cuestión, el nombre del documento HTML de destino seguido del carácter “#” y del nombre indicado en el atributo `name` del ancla. Por ejemplo:

```
<a href="documento1.html#final">Ir al final de documento1.html</a>
```

6.6. Enlaces especiales: correo electrónico. Enlaces de descarga

Además de los usos vistos hasta ahora, los enlaces HTML también se suelen utilizar para:

- Establecer enlaces de descarga.
- Enviar correos electrónicos.

Enlaces hacia otros archivos (vínculos de descarga)

Los vínculos hacia otros archivos son aquellos que direccionan a ficheros que no son documentos web. Para crearlos, es necesario indicar, en el atributo `href` del *link* que corresponda, la ruta en la que se ubica el fichero en cuestión, seguida del nombre y la extensión del mismo. A continuación, se muestra un ejemplo de código de un enlace que apunta a una imagen (“imagen.png”):

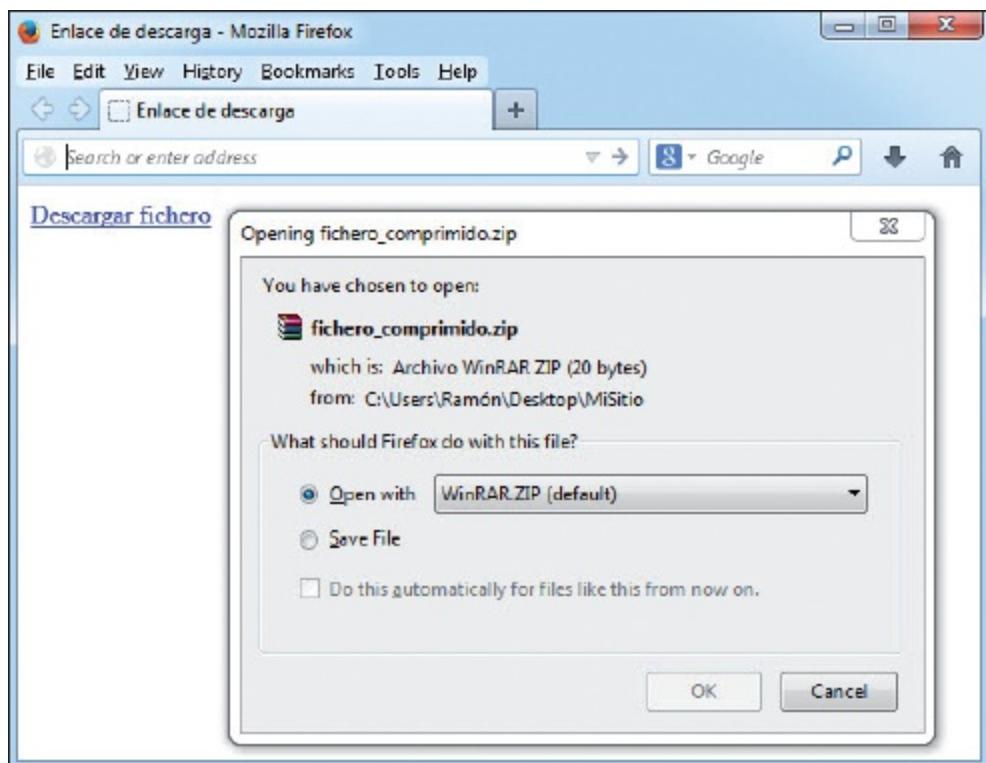
```
<a href="images/imagen.png">Ver imagen</a>
```

Además de abrir documentos web, los navegadores también suelen ser capaces de abrir otros tipos de archivos, como por ejemplo imágenes. En el caso de que el navegador no sea capaz de abrir un fichero hacia el que apunta un enlace, procederá a descargarlo o bien informará al usuario acerca de si quiere visualizarlo (utilizando algún programa que tenga instalado en su equipo) o bien descargarlo.



Sabía que...

Existen infinidad de *plugins* que pueden instalarse en los navegadores web para hacer que puedan abrir determinados tipos de archivos.



Ejemplo de ventana de descarga en el navegador Mozilla Firefox



Actividades

-
16. Cree un enlace de descarga de un archivo llamado “doc1.pdf”, sabiendo que dicho fichero se localiza en un nivel superior respecto al directorio donde se encuentra el documento que contiene el enlace.
-

Enlaces de correo electrónico

Otra funcionalidad que se puede implementar en los enlaces HTML consiste en la creación de *links* de correo electrónico, a partir de los cuales se pueden enviar *e-mails* a una dirección de correo específica. En estos casos, el correo electrónico es enviado a través de algún asistente de correo que el usuario tenga instalado en su equipo.



Definición

Asistente de correo electrónico

Aplicación que se utiliza para enviar y recibir correos electrónicos. Algunos ejemplos de programas de este tipo son: Mozilla Thunderbird, IncrediMail, etc.



Incredimail es un ejemplo de asistente de correo electrónico.

Para que un enlace apunte a una dirección de correo electrónico, es necesario indicar, en su atributo `href`, la cadena `mailto:` seguida de la dirección de correo de destino. A continuación, se puede añadir el carácter `?` para incluir información adicional, como puede ser el asunto del mensaje, el cual se especifica con la cadena `subject=` seguida del texto de dicho asunto. También se puede añadir una dirección de correo alternativa para que reciba el *e-mail*. Para ello, es necesario añadir (a continuación del asunto) la cadena `cc=` seguida de la dirección de correo adicional.

También hay que incluir el carácter & como separador entre ambas informaciones (asunto y dirección de correo alternativa).

A continuación, se muestra un ejemplo de enlace de correo electrónico HTML en el que se ha indicado la siguiente información:

- Dirección de correo: correo@correo_ejemplo.com
- Asunto: “Saludos”.
- Dirección de correo alternativa: correo2@correo_ejemplo.com

```
<a href="mailto:correo@correo_ejemplo.com?subject=Saludos&cc=correo2@correo_ejemplo.com">Enviar correo</a>
```

Aplicación práctica. Creación de enlaces

Desea desarrollar un sitio web con la siguiente estructura:

- En la raíz del sitio, se incluirá un documento web (“index.html”) y dos carpetas (“docs” y “ficheros”).
- Dentro de directorio “docs”, se añadirán dos documentos HTML: “descargas.html” y “contacto.html”.
- En el directorio “ficheros”, se incluyen otras dos carpetas: “zip” y “pdf”. La primera contendrá el archivo: “aplicación.zip”, mientras que la segunda albergará el documento: “leer.pdf”.

Como se puede observar, el sitio web contiene tres documentos HTML: “index.html”, “novedades.html” y “descargas.html”. En ellos, se desean establecer los siguientes enlaces:

- “index.htm”:
 - Un enlace que direccione a “contacto.html” (Contactar).
 - Un enlace que direccione a “descargas.html” (Ir a descargas).
- “descargas.html”:
 - Un enlace de descarga del archivo “aplicacion.zip” (Descargar programa).
 - Un enlace a “leer.pdf” (Descargar manual).

- Un enlace que direccione a “index.html” (Ir a la página principal).
 - Un enlace que direccione a “contacto.html” (Contactar).
- “contacto.html”:
- Un enlace de correo electrónico con las siguientes características:
 - Texto: “Contacta con nosotros”.
 - Dirección de correo de destino: admin@dominio_de_ejemplo.com
 - Asunto: “Contacto”.
 - Dirección de correo alternativa: info@dominio_de_ejemplo.com
 - Un enlace que direccione a “index.html” (Ir a la página principal).
 - Un enlace que direccione a “descargas.html” (Ir a descargas).

Escriba el código HTML correspondiente a cada uno de estos enlaces.

Solución

El código HTML correspondiente a cada uno de estos enlaces es:

- “index.htm”:
- Un enlace que direccione a “contacto.html” (Contactar):
- ```
Contactar
```
- Un enlace que direccione a “descargas.html” (Ir a descargas):
- ```
<a href="docs/descargas.html">Ir a descargas</a>
```
- “descargas.html”:
- Un enlace de descarga del archivo “aplicacion.zip” (Descargar

programa):

```
<a href="../ficheros/aplicacion.zip">Descargar programa</a>
```

- Un enlace a “leer.pdf” (Descargar manual):

```
<a href="../ficheros/leer.pdf">Descargar manual</a>
```

- Un enlace que direccione a “index.html” (Ir a la página principal):

```
<a href="../index.html">Ir a la página principal</a>
```

- Un enlace que direccione a “contacto.html” (Contactar):

```
<a href="contacto.html">Contactar</a>
```

- “contacto.html”:

- Un enlace de correo electrónico con las siguientes características:

- Texto: “Contacta con nosotros”.
- Dirección de correo de destino: admin@dominio_de_ejemplo.com
- Asunto: “Contacto”.
- Dirección de correo alternativa: info@dominio_de_ejemplo.com

```
<a href="mailto:admin@dominio_de_ejemplo.com?subject=Contacto&cc=info@dominio_de_ejemplo.com">Contacta con nosostros</a>
```

- Un enlace que direccione a “index.html” (Ir a la página principal):

```
<a href="../index.html">Ir a la página principal</a>
```

- Un enlace que direccione a “descargas.html” (Ir a descargas):

```
<a href="descargas.html">Ir a descargas</a>
```



Actividades

17. Modifique los enlaces de la aplicación práctica anterior (en los casos en los que sea necesario) teniendo en cuenta que el documento “index.html” se encuentra ahora dentro del directorio docs.
18. Cree un documento web con un enlace de correo electrónico que apunte a su dirección de correo personal. Una vez realizado, ejecute el enlace. ¿Qué ocurre? Razone su respuesta.

7. Imágenes

El uso de imágenes en los documentos HTML es un aspecto muy importante a tener en cuenta en el diseño y desarrollo web profesional, ya que la utilización de estas es fundamental a la hora de crear y desarrollar sitios atractivos, amenos y competentes. En

este apartado, se va a estudiar la manera de insertar y configurar este tipo de recursos.

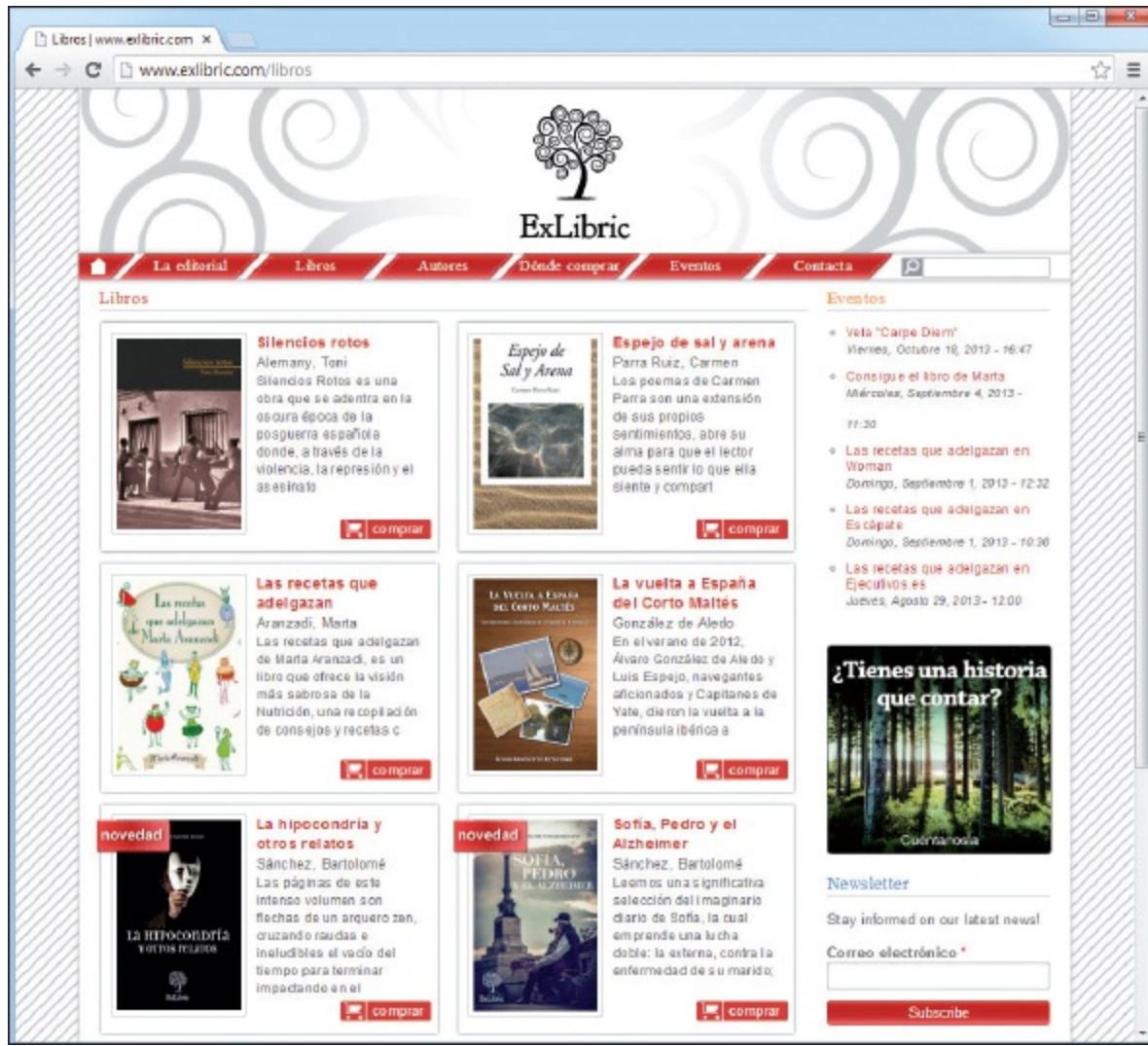


Recuerde

Respecto al número y dimensiones de las imágenes de los documentos web, es necesario establecer un equilibrio. Existen páginas que están repletas casi exclusivamente de grandes y lentas imágenes, mientras que también hay otros sitios que casi todo lo que contienen es texto, desaprovechando el uso de los recursos multimedia. Aunque el esquema a elegir siempre variará de diseñador en diseñador, hay que encontrar un equilibrio entre ambos extremos.

7.1. Formatos de imágenes

Los archivos de imágenes se clasifican, fundamentalmente, atendiendo al formato al que pertenecen. Dicho formato se puede definir como el tipo de compresión a partir del cual se almacena la información binaria de la imagen. Los formatos que más se suelen utilizar en las páginas web son: GIF, JPEG y PNG.



Las imágenes son recursos multimedia utilizados en la inmensa mayoría de páginas web.

GIF

El formato GIF es un método de compresión de imágenes muy utilizado en las páginas web, el cual se caracteriza fundamentalmente por:

- Permitir que las imágenes tengan zonas transparentes (lo que hay detrás de dichas zonas es visible).
- Establecer animaciones en las imágenes. Esta es una de las características más peculiares que caracterizan este formato.

JPEG

El formato JPEG es el más utilizado a la hora de insertar imágenes en las páginas

web. Esto se debe fundamentalmente a la excelente relación calidad/tamaño que presentan las imágenes comprimidas en este formato.



Importante

Para que una página web sea usable, es importante optimizar el tiempo que va a tardar en cargarse, por lo que es fundamental controlar el peso de las imágenes y demás contenidos externos que se inserten en ella.

PNG

El formato PNG es el preferido a la hora de insertar archivos de imágenes con transparencias y se caracteriza, fundamentalmente, por:

- Cuando se usa en las páginas web, su visualización comienza antes de descargarse por completo.
- Los colores de estas imágenes se visualizan exactamente igual en todas las plataformas.
- Comparadas con las imágenes en formato GIF, estas son aproximadamente un 30% más pequeñas.

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

Para insertar imágenes en las páginas web, se utiliza la etiqueta ``, cuya sintaxis básica es la siguiente:

```

```

Los principales atributos soportados por esta directiva, son:

- `src`: en este atributo se especifica la ruta (junto con su nombre y extensión) donde se ubica el archivo de imagen a insertar.

- **title**: gracias a este atributo, se puede indicar un texto que aparecerá cuando el ratón se posicione sobre la imagen.
- **alt**: en este atributo, se puede indicar un texto alternativo que se mostrará en el caso en el que el navegador no pueda cargar la imagen.
- **width**: especifica el ancho de la imagen (en píxeles).
- **height**: especifica el alto de la imagen (en píxeles).



Aplicación práctica

Desea crear un documento HTML (“index.html”) que contenga únicamente una imagen. Escriba el código HTML de dicho documento teniendo en cuenta las siguientes consideraciones:

- La imagen se localiza en una carpeta llamada “images”, ubicada en el mismo directorio que “index.html”. El nombre de la imagen es: “imagen.jpg”.
- Cuando el ratón se posicione sobre la imagen, aparecerá el texto: “Dispositivos de medida”.
- Si el navegador no es capaz de visualizar la imagen, se verá en su lugar el siguiente texto: “Imagen de cuatro dispositivos de medida”.
- La imagen se visualizará con las siguientes dimensiones: ancho: 600 píxeles, alto: 150 píxeles.
- El título del documento será: “Ejemplo de documento html con una imagen”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.

SOLUCIÓN

```
<!DOCTYPE html>
<html>
<head>
    <title>Ejemplo de documento html con una imagen</title>
    <meta charset="UTF-8"/>
</head>
<body>
    
</body>
</html>
```



Ejemplo de documento HTML con una imagen



Actividades

19. Modifique el código del documento de la aplicación práctica anterior teniendo en cuenta que ahora la imagen se ubica en el mismo directorio que “index.html”.
20. Identifique y corrija los errores que se han cometido al escribir la siguiente

línea de código:

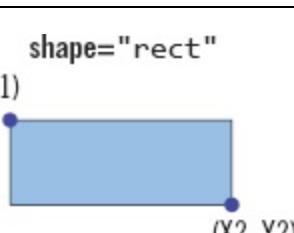
```
<img href="imagen.mp3" title="Título de la imagen"></img>
```

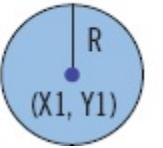
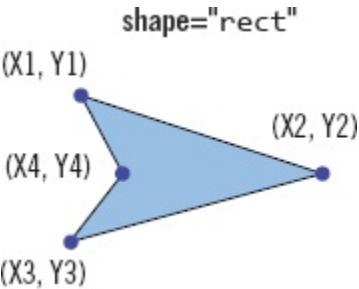
7.3. Enlaces en imágenes

En HTML, también existe la posibilidad de crear lo que se conoce como mapas de imágenes. Estos elementos consisten en imágenes dentro de las cuales se han establecido enlaces o áreas de clic, a las cuales se les puede asociar una URL como si se tratara de vínculos convencionales. Para crear un mapa de imágenes en HTML se utiliza la directiva `<map>`, dentro de la cual se van creando las diferentes áreas de clic con etiquetas `<area>`. Los principales atributos que soporta la directiva `<area>` son:

- `title`: aquí se indica un texto que aparecerá cuando el ratón se posicione sobre el área de clic correspondiente.
- `shape`: con este atributo, se puede definir el tipo de área de clic. Sus posibles valores son: "rect" (rectángulo), "circle" (círculo) y "poly" (polígono).
- `coords`: en este atributo, se especifican las coordenadas que delimitan al área de clic. Como se verá a continuación, el número de coordenadas que hay que especificar varía en función del tipo de área que corresponda.
- `href`: aquí se indica la URL a la que el navegador direcciará cuando se haga clic en el área.

A continuación, se muestra una tabla que informa acerca de la manera de configurar cada área de clic según el tipo del que se trate.

Valor de <code>shape</code>	Descripción	Sintaxis	Ejemplo
<code>rect</code>	Área de clic rectangular. Para definirla, se especifican las coordenadas de los puntos de la esquina superior izquierda (X_1, Y_1) y de la esquina inferior derecha (X_2, Y_2).	<code><area shape="rect" coords="X1,Y1,X2,Y2" href="...></code>	<code>shape="rect"</code> (X_1, Y_1)  (X_2, Y_2)

circle	Área de clic circular. Es necesario especificar la coordenada del centro del círculo (X1, Y1) y la longitud de su radio (R).	<pre><area shape="circle" coords="X1,Y1,R" href=...></pre>	 shape="circle"
poly	Área de clic poligonal. Se define indicando cada uno de sus puntos, los cuales deben ser declarados en orden, siguiendo el camino marcado por el perímetro del polígono.	<pre><area shape="poly" coords=" X1,Y1, X2,Y2, X3,Y3, X4,Y4" href=...></pre>	 shape="rect"

Es importante tener en cuenta que, cuando se definen áreas de clic en este tipo de elementos, los puntos se determinan teniendo en cuenta que el origen del sistema de coordenadas se ubica en la esquina superior izquierda de la imagen en cuestión.



Ejemplo

Si una imagen tiene un ancho de 650 píxeles y un alto de 385 píxeles, la coordenada de la esquina superior izquierda corresponderá al punto (0,0), mientras que la de la esquina inferior derecha será el punto (650, 385).

Un vez que se han definido las áreas de clic del mapa, se hace necesario asociarlas a una imagen, la cual se definirá con su correspondiente etiqueta (``) fuera de la directiva `<map>`. Para indicar que esta imagen debe albergar las zonas de clic definidas en la directiva `<map>`, es necesario añadir, en la etiqueta de dicha imagen, el atributo `usemap`. El valor de este debe ser el mismo que el del atributo `name` del mapa de imágenes, precedido por el símbolo "#". A continuación, se muestra un fragmento de código en el que se puede apreciar la sintaxis de un mapa de imágenes asociado a una imagen:

```
...
<map name="mapa_de_ejemplo">
    <area.../>
    <area.../>
    <area.../>
    <area.../>

...
</map>

...
```



Actividades

21. Busque en Internet alguna imagen en la que aparezca un polígono de 6 lados y utilícela para crear un mapa de imágenes con un área de clic. Dicha área debe coincidir con el perímetro del polígono en cuestión.
-



Ejemplo

El siguiente código HTML corresponde a un mapa de imágenes definido en una imagen (“imagen1.png”). En dicho mapa, se han establecido tres áreas de clic: rectangular, circular y poligonal (triangular).

```
...
<MAP name="mapa">
<AREA shape="rect" coords="0,0,100,200"/>
<AREA shape="circle" coords="350,350,50"/>
<AREA shape="poly" coords="580,290,360,350,560,440"/>
</MAP>
...
<IMG src="imagen1.png" usemap="#mapa"/>
...
```

7.4. Imágenes de fondo

La nueva especificación HTML5 establece que, para colocar imágenes de fondo en las páginas web, es necesario utilizar hojas de estilo CSS. Antes de la aparición de este estándar, existía un atributo (`background`) que se incluía dentro de la directiva `<body>` que permitía insertar imágenes de fondo en los documentos web, cuyo valor correspondía con la ruta en la que se ubicaba el archivo de imagen en cuestión. Su sintaxis era la siguiente:

```
<body background=".."/>
```

Aunque el nuevo estándar desaprueba el uso de este atributo, todavía sigue siendo soportado por la mayoría de los navegadores actuales.



Ejemplo

A continuación, se muestran dos fragmentos de código correspondientes a la inserción de una imagen de fondo (“`imagen.jpg`”) en un documento web, la cual consiste en un círculo blanco. En el primer ejemplo, se utiliza el atributo `background`, mientras que, en el otro, se emplean reglas CSS. El resultado

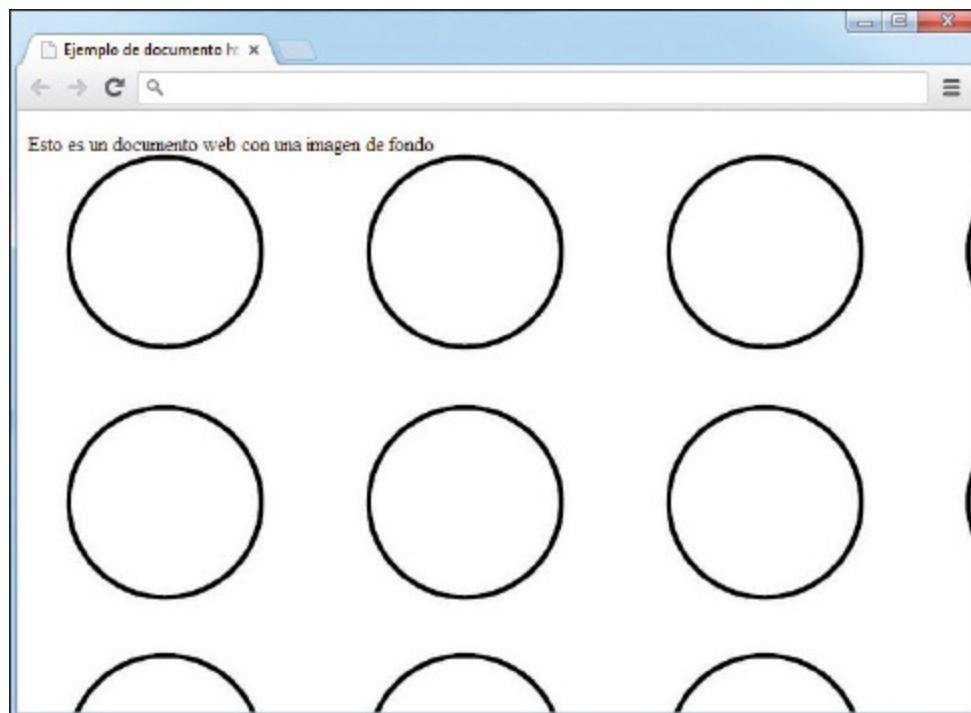
obtenido es idéntico en ambos métodos.

- Código con atributo background:

```
...  
<body background="imagen.jpg">  
...
```

- Código con CSS:

```
...  
<body style="background-image:url(imagen.jpg)">  
...
```



Ejemplo de documento web con imagen de fondo

Como se puede observar en el ejemplo anterior, cuando se insertan imágenes de fondo en los documentos web, estas aparecen con repetición a lo largo de los ejes X e Y del área de visualización del documento. Este comportamiento puede ser modificado fácilmente con CSS.

8. Listas

Otros elementos muy utilizados en HTML son las listas. En este apartado se van a estudiar la manera de implementar y configurar los distintos tipos de listas que existen en este estándar.

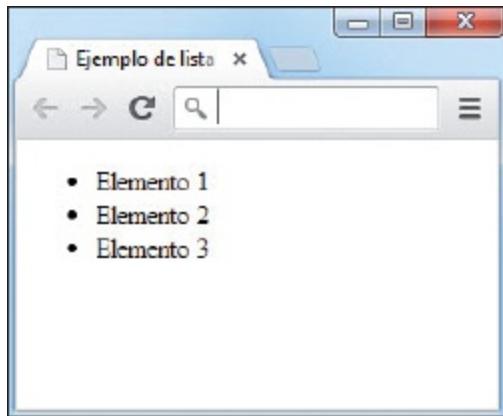
8.1. Características y tipos de listas

En HTML, las listas consisten en una sucesión de elementos precedidos por una viñeta (listas no ordenadas) o un número (listas ordenadas). A continuación, se enumeran los principales tipos de listas que se pueden implementar en HTML.

Listas no ordenadas

Las listas no ordenadas son aquellas que no tienen ordenación y cuyos elementos van precedidos por algún símbolo o imagen. Los elementos de una lista de este tipo van dentro de la etiqueta `` y, por cada punto que se desee añadir, es necesario utilizar la etiqueta ``. A continuación, se muestra un fragmento de código correspondiente a un ejemplo de lista no ordenada:

```
...
<ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
</ul>
...
```



Ejemplo de visualización de una lista no ordenada

Para modificar el aspecto de las viñetas que acompañan a los elementos de las listas no ordenadas, se puede hacer uso del atributo `type`. Este atributo se puede incluir dentro de las directivas `` y puede tener los siguientes valores:

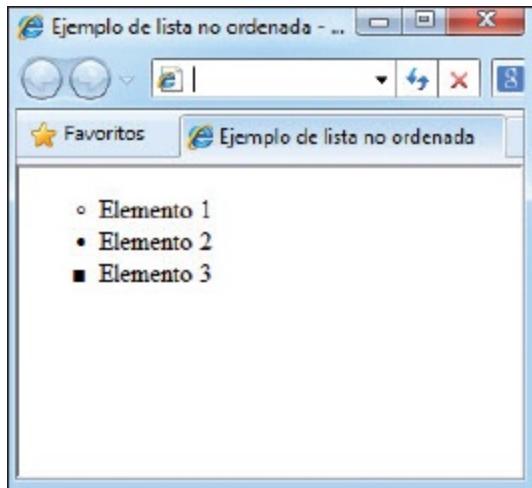
- `disc`: círculo de color negro.
- `circle`: círculo con fondo blanco.
- `square`: cuadrado.



Ejemplo

A continuación, se muestra un fragmento de código HTML correspondiente a una lista no ordenada. En cada uno de sus elementos se ha establecido un tipo de viñeta diferente:

```
...
<ul>
    <li type="circle">Elemento 1</li>
    <li type="disc">Elemento 2</li>
    <li type="square">Elemento 3</li>
</ul>
...
```



Ejemplo de visualización de una lista no ordenada en el navegador Internet Explorer



Actividades

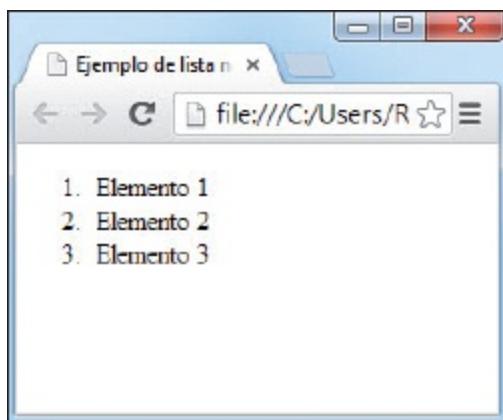
22. Investigue en Internet acerca de la manera de modificar el aspecto de las listas no ordenadas utilizando hojas de estilo CSS. Una vez hecho esto, escriba un código que permita obtener el mismo resultado que el mostrado en el ejemplo anterior, pero sin utilizar en ningún caso el atributo `type`.
-

Ordenación de listas

Las listas ordenadas se definen con la directiva ``, dentro de la cual se van definiendo los diferentes puntos con etiquetas `` (igual que en las listas no ordenadas).

En este tipo de listas, los símbolos que precederán a cada uno de los elementos son de carácter numérico y se irán generando automáticamente conforme se añadan puntos. A continuación, se muestra un ejemplo muy sencillo de lista ordenada:

```
...
<ol>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
</ol>
...
```



Ejemplo de visualización de una lista ordenada

En este tipo de listas, se puede establecer fácilmente el número inicial a partir del cual se van a contabilizar los elementos de misma. Esto se consigue gracias al atributo `value`. Este atributo se puede incluir dentro de las etiquetas `` y los siguientes puntos que se escriban se generarán automáticamente por orden, partiendo del número indicado en dicho atributo.



Actividades

23. Modifique la lista del ejemplo anterior para que sus elementos comiencen a contabilizarse a partir del número 2.

El tipo de numeración de las listas ordenadas es otro aspecto que puede ser modificado con HTML. Para ello, se puede utilizar el atributo `type`, el cual puede tener los siguientes valores:

- 1: numeración estándar: 1, 2, 3...

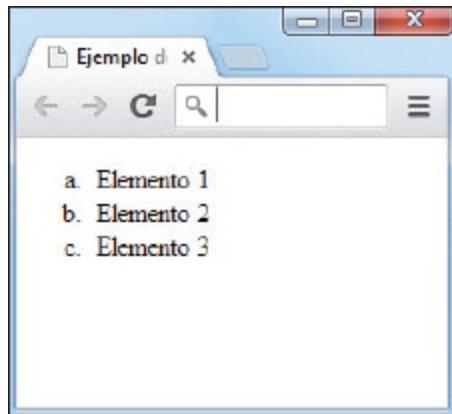
- a: numeración alfabética en minúsculas: a, b, c...
- A: numeración alfabética en mayúsculas: A, B, C...
- i: numeración en números romanos y en minúsculas: i, ii, iii...
- I: numeración en números romanos y en mayúsculas: I, II, III...



Ejemplo

A continuación, se muestra un fragmento de código HTML correspondiente a una lista ordenada sencilla en la que se ha establecido el tipo de numeración alfabética:

```
...
<ol>
  <li type="a">Elemento 1</li>
  <li type="a">Elemento 1</li>
  <li type="a">Elemento 1</li>
</ol>
...
```



Ejemplo de lista ordenada con numeración alfabética

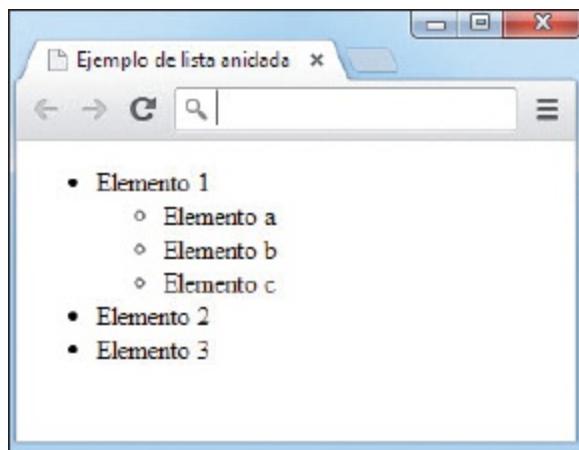


Actividades

-
24. Modifique el código de la lista anterior para hacer que sus elementos empiecen a contabilizarse a partir del orden 2. La numeración de los elementos debe visualizarse en números romanos.
-

8.2. Anidamiento en listas

En HTML también es posible establecer listas anidadas, es decir, listas que contengan otras dentro de ellas. Para implementarlas, basta con tener en cuenta que una lista puede establecerse como si se tratara de un elemento de otra lista.



Ejemplo de lista anidada

Desde el punto de vista de la programación en HTML, se puede afirmar que, en este ejemplo, la lista no ordenada principal contiene 4 elementos: 3 elementos de lista convencionales (Elemento 1, Elemento 2 y Elemento 3) y otra lista no ordenada, la cual está situada en segundo lugar (después de Elemento 1) y está constituida por otros tres elementos: Elemento a, Elemento b y Elemento c.



Nota

Las listas ordenadas también se pueden establecer con anidamientos.

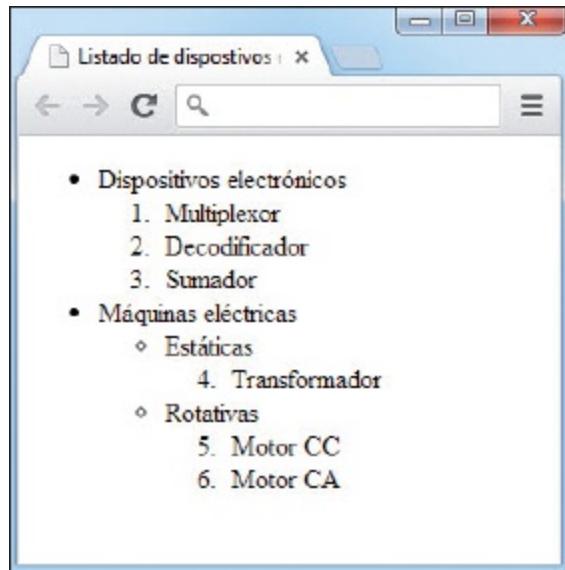
A continuación, se muestra el fragmento de código HTML correspondiente a esta lista anidada:

```
...
<ul>
    <li>Elemento 1</li>
    <ul>
        <li>Elemento a</li>
        <li>Elemento b</li>
        <li>Elemento c</li>
    </ul>
    <li>Elemento 2</li>
    <li>Elemento 3</li>
</ul>
...
```



Aplicación práctica

Se dispone a crear un documento HTML en el que desea incluir una lista anidada que debe visualizarse de la siguiente manera:



Ejemplo de lista anidada

Escriba el código HTML del documento correspondiente teniendo en cuenta las siguientes consideraciones:

- El título del documento será: “Listado de dispositivos eléctricos/electrónicos”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.

SOLUCIÓN

```
<!DOCTYPE html>
<html>
    <head>
        <title>Listado de dispositivos
        eléctricos/electrónicos</title>
        <meta charset="UTF-8"/>
    </head>
    <body>
        <ul>
            <li>Dispositivos electrónicos</li>
            <ol>
                <li>Multiplexor</li>
                <li>Decodificador</li>
                <li>Sumador</li>
            </ol>
            <li>Máquinas eléctricas</li>
            <ul>
                <li>Estáticas</li>
                <ol>
                    <li value="4">Transformador</li>
                </ol>
                <li>Rotativas</li>
                <ol>
                    <li value="5">Motor CC</li>
                    <li>Motor CA</li>
                </ol>
            </ul>
        </ul>
    </body>
</html>
```

8.3. Otros tipos de listas: listas de definición

Las listas de definiciones son otro tipo de listas que se pueden utilizar cuando se desean incluir definiciones, es decir, cuando se quiere mostrar un concepto y, justo debajo, su definición. Estos elementos tienen sus propias etiquetas específicas, las cuales se describen a continuación.

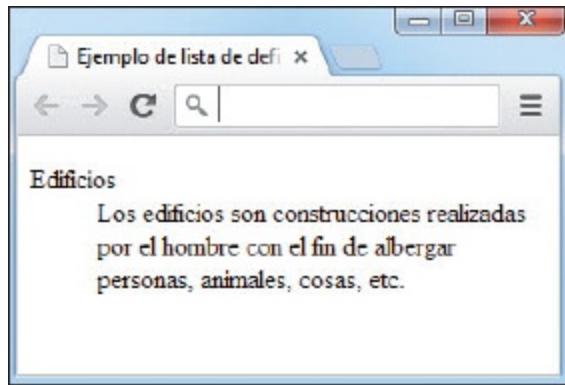
- <dl>: dentro de esta etiqueta se incluye todo el contenido del bloque concepto-definición.
- <dt>: esta etiqueta alberga el concepto que se va a definir.
- <dd>: en esta directiva se incluye la descripción del concepto en cuestión.



Ejemplo

A continuación, se muestra un fragmento de código correspondiente a un ejemplo sencillo de lista de definición:

```
...
<dl>
    <dt>Edificios</dt>
    <dd>Los edificios son construcciones realizadas
        por el hombre con el fin de albergar personas,
        animales, cosas, etc.</dd>
</dl>
...
```



Ejemplo de lista de definición

9. Tablas

Las tablas son muy utilizadas en HTML para estructurar datos y presentar información de manera ordenada y relacionada. En este apartado, se estudiará la manera implementar y configurar este tipo de elementos en los documentos web.

9.1. Estructura básica

En HTML, una tabla básica puede definirse como un conjunto de filas donde cada una de ellas alberga un determinado número de celdas, dentro de las cuales se insertan los datos que se desean estructurar.

En definitiva, una tabla simple puede ser definida según tres etiquetas:

- `<table>`: contenedor principal de la tabla. Dentro de ella, se insertan las diferentes filas y columnas que la constituyen.
- `<tr>`: etiqueta correspondiente a cada una de las filas de la tabla. Cada fila contiene un determinado número de celdas.
- `<td>`: cada celda de la tabla es definida con esta directiva. Dentro de ella se escribe el contenido que se desea incluir en la misma.

Partes de una tabla



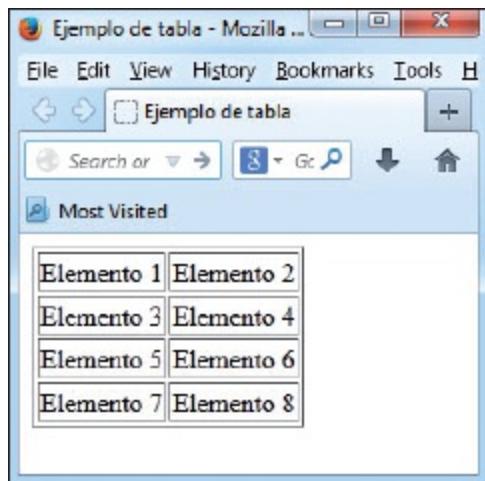
Cuando se inserta una tabla en HTML, el contenido de la misma debe definirse dentro de la etiqueta `<table>`. El orden de construcción de las tablas se establece de la forma: fila-celdas de la fila, es decir, se van definiendo cada una de las filas de la tabla (con la directiva `<tr>`) y dentro de ellas se van insertando las diferentes celdas (con la etiqueta `<td>`) que pertenecen a cada una de dichas filas.



Ejemplo

A continuación se muestra un fragmento de código HTML correspondiente a una tabla básica constituida por 4 filas y 2 columnas:

```
...
<table border=1>
  <tr>
    <td>Elemento 1</td>
    <td>Elemento 2</td>
  </tr>
  <tr>
    <td>Elemento 3</td>
    <td>Elemento 4</td>
  </tr>
  <tr>
    <td>Elemento 5</td>
    <td>Elemento 6</td>
  </tr>
  <tr>
    <td>Elemento 7</td>
    <td>Elemento 8</td>
  </tr>
</table>
...
```



Ejemplo de visualización de una tabla constituida por cuatro filas y dos columnas

Por defecto, las tablas que se implementan en HTML se visualizan sin bordes. Esto

puede ser modificado con CSS o bien añadiendo el atributo border dentro de la etiqueta <table> (ver ejemplo anterior).

9.2. Formato de tablas: bordes, alineación, tamaño, etc.

Para dar formato a las tablas de los documentos web, la nueva especificación HTML5 recomienda usar hojas de estilo CSS. No obstante, también existen una serie de atributos que permiten modificar la apariencia de estos elementos. Estos son:

- **align**: este atributo especifica la alineación de la tabla en el documento. Sus posibles valores son: `left` (izquierda), `center` (centro) y `right` (derecha). Este atributo no es soportado por la nueva especificación HTML5, pero es reconocido por la mayoría de los navegadores actuales.
- **border**: con este atributo, se puede habilitar la visualización del borde que delimita la tabla y sus celdas. Si se indica el valor 1, dichos bordes serán visibles. Si su valor se deja vacío ("") o bien se omite este atributo, la tabla se visualizará sin bordes (únicamente aparecerá el contenido de las celdas). Este atributo no es soportado por la especificación HTML5, pero, actualmente, es reconocido por la mayoría de los navegadores más utilizados.
- **width**: con este atributo se puede indicar el ancho de la tabla (en píxeles).
- **cellpadding**: este atributo sirve para modificar el espacio (en píxeles) que existe entre las celdas de la tabla y el contenido que albergan. Aunque este atributo no es soportado por HTML5, sí que es reconocido por la mayoría de los navegadores.
- **cellspacing**: con este atributo se puede modificar el espacio (en píxeles) que hay entre las celdas de una tabla. Este atributo tampoco es soportado por la nueva especificación HTML5, pero sí que sigue siendo reconocido por la mayoría de los navegadores actuales.



Actividades

25. Elabore un documento web que contenga una tabla que muestre algunos de sus datos personales (nombre, dirección, teléfono, domicilio, etc.). Para ello, debe tener en cuenta las siguientes consideraciones:
 - Los bordes que delimitan la tabla y sus celdas deben visualizarse.
 - El ancho de la tabla será de 600 píxeles.

- La tabla debe aparecer centrada en el documento.
26. Investigue en Internet acerca de las reglas CSS que permiten modificar el estilo de las tablas HTML. Vuelva a escribir el documento de la actividad anterior utilizando estas propiedades y sin hacer uso de cualquiera de los atributos vistos en este apartado.
-

9.3. Formato de contenido de celdas

En las etiquetas <td> también se pueden incluir atributos que permitan modificar el estilo de las celdas y su contenido. A continuación, se muestran algunos de ellos:

- **align:** especifica la alineación del contenido de la celda. Puede tener los siguientes valores: `left` (izquierda), `right` (derecha) y `center` (centrado).
- **bgcolor:** especifica el color de fondo de la celda.
- **height:** con este atributo se puede establecer el alto (en píxeles) de la celda en cuestión.
- **width:** este atributo se puede usar para modificar el ancho (en píxeles) de la celda correspondiente.



Recuerde

Cada celda de una tabla es definida con la directiva <td>. Dentro de ella, se escribe el contenido que se desea incluir en la misma.

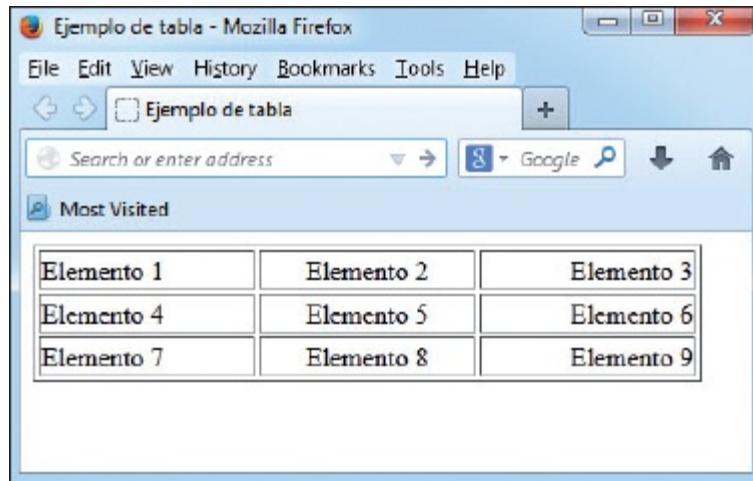
Los atributos `align` y `bgcolor` no son soportados por la nueva especificación HTML5. No obstante, sí que todavía siguen siendo reconocidos por la mayoría de los navegadores actuales.



Ejemplo

A continuación, se muestra un fragmento de código correspondiente a una tabla en la que se ha modificado la alineación del contenido de sus celdas de la siguiente manera:

- Celdas de la columna de la izquierda: contenido alineado a la izquierda.
- Celdas de la columna de la derecha: contenido alineado a la derecha.
- Celdas de la columna central: contenido centrado.



Ejemplo de tabla con diferentes alineaciones en el contenido de sus celdas

9.4. Agrupamiento de filas y columnas

En HTML, existe un conjunto de etiquetas que permiten establecer ciertas agrupaciones en las filas y columnas de las tablas para así poder ejercer un mayor control sobre la visualización de estos elementos.

Agrupaciones de filas

Las etiquetas que permiten agrupar filas de una tabla son:

- <thead>: para agrupar las filas de la cabecera de una tabla.
- <tbody>: para agrupar las filas que pertenecen al cuerpo de una tabla.
- <tfoot>: para agrupar las filas del pie de una tabla.

El fragmento de código que se muestra a continuación corresponde a una tabla en la que se han utilizado estas etiquetas para establecer agrupaciones en sus filas con el fin de aplicar fácilmente estilos CSS comunes para los contenidos de un mismo grupo. Como se puede observar, gracias a la utilización de etiquetas, se pueden establecer estilos que afecten a los contenidos encerrados en cada una de las agrupaciones (sin necesidad de modificar el aspecto de cada una de las celdas por separado):

```
...
<table border=1 width=500>
    <thead style="font-weight:bold;text-decoration:underline">
        <tr>
            <td>Elemento</td>
            <td>Modelo</td>
            <td>Rendimiento (%)</td>
            <td>Precio (€)</td>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>Modelo A</td>
            <td>AA123</td>
            <td>50</td>
            <td>500</td>
        </tr>
        <tr>
            <td>Modelo B</td>
            <td>BB345</td>
            <td>60</td>
            <td>600</td>
        </tr>
        <tr>
            <td>Modelo C</td>
            <td>CC678</td>
            <td>90</td>
            <td>800</td>
        </tr>
    </tbody>
    <tfoot style="font-weight:bold">
        <tr>
            <td>Total</td>
            <td></td>
            <td></td>
            <td></td>
        </tr>
    </tfoot>
</table>
...

```

Elemento	Modelo	Rendimiento (%)	Precio (€)
Modelo A	AA123	50	500
Modelo B	BB345	60	600
Modelo C	CC678	90	800
Total			1900

Ejemplo de tabla con filas agrupadas



Actividades

27. Escriba el código HTML correspondiente a una tabla que sea idéntica a la anterior, pero sin utilizar etiquetas de agrupación de filas. ¿Qué conclusiones saca al respecto? ¿Qué método considera más apropiado? Razone su respuesta.
-

Agrupaciones de columnas

En el caso en el que se deseen establecer agrupaciones de columnas que sirvan para aplicarles estilos y formatos, se hace necesario utilizar las siguientes directivas:

- `<col>`: se refiere a una columna determinada.
- `<colgroup>`: se refiere a un grupo de columnas.

En resumen, para definir un grupo de columnas, hay que utilizar la etiqueta `<colgroup>`, dentro de la cual se pueden ir definiendo las diferentes columnas con la directiva `<col>`. Con el uso del atributo `span` de la directiva `<colgroup>`, se puede especificar el número de columnas que va a contener dicho grupo.

Los grupos de columnas deben definirse justo después de la apertura de la directiva `<table>` y antes de incluir las diferentes filas.



Ejemplo

A continuación, se muestra un ejemplo de uso de estas etiquetas en una tabla constituida por dos filas y cinco columnas:

```
...
<table border=1>
    <colgroup span=3 width=100></colgroup>
    <colgroup>
        <col width=75></col>
        <col width =100></col>
    </colgroup>
    <tr>
        <td> Columna 1</td>
        <td>Columna 2</td>
        <td> Columna 3</td>
        <td> Columna 4</td>
        <td> Columna 5</td>
    </tr>
    <tr>
        <td>Columna 1</td>
        <td> Columna 2</td>
        <td> Columna 3</td>
        <td> Columna 4</td>
        <td> Columna 5</td>
    </tr>
</table>
```

Columna 1	Columna 2	Columna 3	Columna 4	Columna 5
Columna 1	Columna 2	Columna 3	Columna 4	Columna 5

Ejemplo de tabla con agrupaciones en sus columnas



Actividades

28. Modifique el código del ejemplo anterior para establecer negrita en el contenido de las celdas de la columna central. Utilice para ello reglas CSS. Para establecer negrita en el contenido de una etiqueta usando hojas de estilo se puede añadir la siguiente línea dentro del correspondiente atributo `style: font-weight:bold;`.
-

9.5. Tablas anidadas

En HTML también es posible establecer anidamientos de tablas, es decir, insertar tablas unas dentro de otras.

Para anidar una tabla dentro de otra, simplemente basta con introducir la estructura de la primera dentro de una de las celdas de la tabla principal.

Es importante tener en cuenta que no hay límite respecto al número de tablas que se pueden anidar dentro de una determinada tabla, aunque siempre es recomendable utilizar el menor número de tablas que sea posible para que el código resultante no se haga excesivamente enrevesado y complejo.



Ejemplo

A continuación, se muestra un ejemplo de código correspondiente a una tabla

constituida por dos filas y dos columnas. Como se puede ver, en la primera celda de la segunda fila se ha insertado otra tabla formada por tres filas y dos columnas:

```
...
<table border="1">
  <tr>
    <td>Celda de la tabla principal</td>
    <td>Celda de la tabla principal</td>
  </tr>
  <tr>
    <td>
      <table border="1">
        <tr>
          <td>Celda de la tabla secundaria</td>
          <td>Celda de la tabla secundaria</td>
        </tr>
        <tr>
          <td>Celda de la tabla secundaria</td>
          <td>Celda de la tabla secundaria</td>
        </tr>
        <tr>
          <td>Celda de la tabla secundaria</td>
          <td>Celda de la tabla secundaria</td>
        </tr>
      </table>
    </td>
    <td>Celda de la tabla principal</td>
  </tr>
</table>
```

Ejemplo de tabla anidada



Actividades

29. Modifique el código del ejemplo para anidar la tabla que tiene dos filas y dos columnas, dentro de la celda inferior izquierda de la secundaria.
-

9.6. Buenas prácticas en el uso de tablas

Aunque antes se solían utilizar también para maquetar los contenidos de las páginas web, las tablas deben usarse únicamente en el caso en el que se deseen tabular datos (texto, imágenes, enlaces, etc.). El uso de hojas de estilo CSS para posicionar los diferentes elementos que forman parte de un sitio web proporciona una serie de ventajas en relación con la utilización de tablas para tal fin. Estas son:

- Se logra separar el contenido de la presentación, ya que, al utilizar tablas, se están manteniendo, en un mismo archivo, ambas informaciones.
- La utilización de hojas de estilo CSS minimiza considerablemente la cantidad de etiquetas HTML necesarias para maquetar una web, lo cual permitirá reducir el tiempo de carga de misma en el navegador.
- Al tener separado el contenido de la presentación, la página será mejor rastreada por los buscadores.
- El tiempo necesario para maquetar un sitio web completo utilizando tablas es mucho mayor que con CSS.
- Tener separado el contenido de la presentación hace posible que se pueda

modificar rápidamente el diseño de la web sin necesidad de alterar la estructura de la página.



Importante

La nueva especificación HTML5 recomienda el uso de CSS para dar estilo a los diferentes elementos que forman parte de los documentos web.

10. Marcos (*frames*)

Los marcos o *frames* son herramientas que permiten establecer vistas de uno o varios documentos HTML en uno solo. En este apartado, se estudiará la manera de crear y configurar este tipo de elementos en las páginas web.

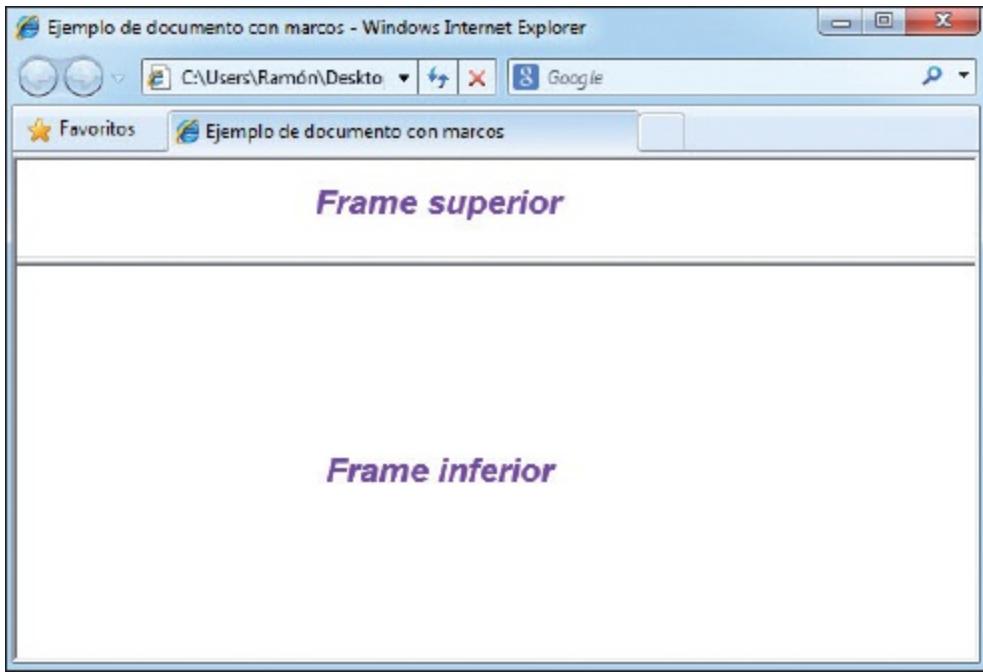
10.1. Creación de marcos

Existen dos tipos fundamentales de marcos: los marcos fijos o *frames* y los marcos incrustados o *iframes*. Estos últimos se estudiarán al final del presente apartado.

Los documentos HTML que contienen *frames* no presentan etiqueta `<body>`. En su lugar, cuentan con la etiqueta `<frameset>`, que se utiliza para definir las diferentes áreas en las que se subdividirá el documento correspondiente.

Por otro lado, existe la directiva `<frame>`. Esta etiqueta se usa para definir el contenido de cada uno de los marcos que se han establecido con `<frameset>`.

A continuación, se muestra un ejemplo de visualización de un documento HTML en el que se han creado dos marcos: uno superior y otro inferior.



Ejemplo de documento web con marcos

Cuando se utilizan marcos, también es importante hacer uso de la directiva `<noframes>`, la cual se suele situar justo después del cierre del *frame* principal y cuyo contenido se visualizará en el caso en el que el navegador no soporte el uso de marcos o bien halla sido configurado para no utilizarlos.

A continuación, se muestra un fragmento de código que permite comprender la manera de implementar marcos en HTML:

```
...
<frameset ...>
  <frame...></frame>
  <frame...></frame>
  ...
</frameset>
<noframes>
  ...
</noframes>
...
```

10.2. Ventajas e inconvenientes en el uso de marcos. Soporte de

navegadores

El uso de marcos en los documentos HTML puede resultar ventajoso o beneficioso. A continuación, se enumeran algunas de las principales ventajas y desventajas que supone la utilización de estos elementos en las páginas web.

Ventajas del uso de frames

El uso de *frames* en los documentos web proporciona las siguientes ventajas fundamentales:

- La navegación se hace más rápida después de la primera carga.
- Es desarrollo de sitio puede ser más rápido. Por ejemplo, si se desarrolla un documento web que consiste únicamente en la barra de navegación principal de la web, ya no es necesario volver a construirla en los demás documentos del sitio.
- Se pueden establecer partes de la web que estén siempre fijas, lo cual resulta ventajoso en muchos casos.

Inconvenientes de usar frames

El uso de marcos fijos o *frames* proporciona también una serie de inconvenientes que es importante tener en cuenta:

- Eliminan espacio útil de la pantalla debido a que el lugar ocupado por los *frames* es siempre fijo.
- Para muchos usuarios, la presencia de estas partes fijas resulta limitador en relación a su movilidad a través la web.
- La nueva especificación HTML5 no soporta estos elementos (excepto los *iframes* o *frames* incrustados). No obstante, sí que siguen siendo visualizados correctamente por la mayoría de los navegadores actuales.
- Los marcadores o favoritos no funcionan correctamente en ciertos casos.
- Es posible que el botón “atrás” del navegador no se comporte como se espera que lo haga.



Nota

La nueva especificación HTML5 no soporta el uso de marcos o *frames*. En su lugar, se recomienda el uso de CSS para posicionar y dar estilo a las diferentes partes de los documentos que, antes de la aparición de este estándar, se solían establecer mediante el uso de marcos (barra de navegación, menús laterales, pie de página etc.).

10.3. Formateado de marcos

Los diferentes atributos específicos que soportan las directivas `<frame>` y `<frameset>` permiten una gran variedad de opciones de configuración para estos elementos.

Los principales atributos que soporta la directiva `<frameset>` son:

- `cols`: especifica el tamaño de cada una de las divisiones verticales a establecer.
- `rows`: este atributo indica el tamaño de cada una de las divisiones horizontales.

Cada valor indicado en estos atributos debe separarse por comas y se puede dar en: % y píxeles. También se puede emplear el carácter “*”, que indica que el tamaño de esa división corresponderá el resto del espacio disponible (del área de navegación) para albergar ese marco.

A continuación, se enumeran los principales atributos soportados por la etiqueta `<frame>`:

- `name`: su valor corresponde con el nombre del marco.
- `src`: contiene la ruta del documento HTML que va a albergar el marco.
- `frameborder`: puede tener los valores 1 y 0, que indican, respectivamente, si el *frame* tendrá o no borde asociado.
- `marginwidth`: indica el espacio (en píxeles) que habrá entre el contenido del *frame* y sus márgenes izquierdo y derecho.
- `marginheight`: indica el espacio (en píxeles) que habrá entre el contenido del *frame* y sus márgenes superior e inferior.
- `scrolling`: sirve para indicar si se permite o no el uso de la barra de *scroll* o deslizamiento para desplazarse por el *frame*. Sus posibles valores son: *no* (*scroll* deshabilitado) y *yes* (*scroll* habilitado).
- `noresize`: si se añade esta atributo (sin valor), se está indicando que el *frame* no puede redimensionarse por el usuario (al hacer clic y arrastrar el borde que

los delimita).

- `frameborder`: indica si el *frame* tendrá o no un borde asociado. Puede tener los siguientes valores: 0 (sin borde) y 1 (con borde).

Aplicación práctica

Desea crear un documento web (“index.html”) con marcos que debe tener las siguientes características:

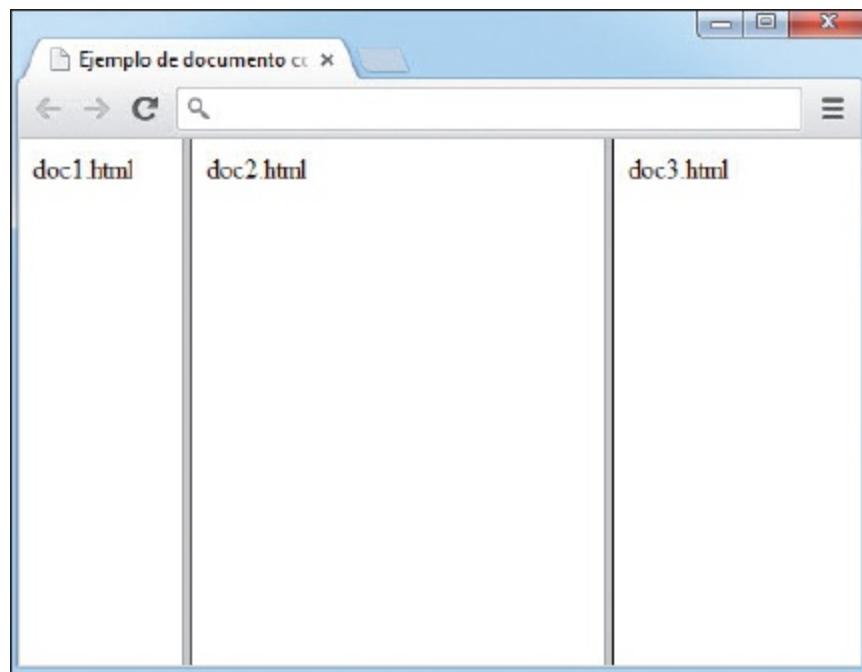
- Se van a establecer tres divisiones verticales, cuyas áreas deben albergar, de izquierda a derecha, los siguientes documentos: “doc1.html”, “doc2.html” y “doc3.html”.
- El nombre de cada marco corresponderá con el nombre del documento HTML que alberga (sin extensión).
- El tamaño de las divisiones verticales será (de izquierda a derecha) de: 20%, 50%, 30%.
- Los marcos no podrán redimensionarse por el usuario.
- El uso de la barra de *scroll* estará deshabilitado únicamente en el *frame* central.

Escriba el código HTML del documento teniendo en cuenta las siguientes consideraciones:

- Todos los documentos HTML se encuentran ubicados en el mismo directorio.
- En el caso en el que el navegador no soporte *frames*, deberá visualizarse un párrafo con el siguiente texto: “Su navegador no soporta marcos. Revise su configuración o pruebe con otro navegador”.
- El título del documento será: “Ejemplo de documento con tres marcos verticales”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.

Solución

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de documento con tres marcos verticales</title>
    <meta charset="UTF-8"/>
  </head>
  <frameset cols="20%,50%,30%">
    <frame name="doc1" src="doc1.html" noresize></frame>
    <frame name="doc2" src="doc2.html" noresize scrolling="no"></frame>
    <frame name="doc3" src="doc3.html" noresize></frame>
  </frameset>
  <noframes>
    <p>Su navegador no soporta marcos. Revise su configuración o pruebe con otro navegador.</p>
  </noframes>
</html>
```



Ejemplo de documento web con tres marcos verticales

10.4. Enlaces entre contenidos de marcos

Una de las funciones principales de los *frames* es la que permite modificar el contenido de uno de los marcos a partir de las acciones ejecutadas en otro diferente.

Por ejemplo, al seleccionar un determinado enlace contenido en un *frame*, el contenido que hay en otro cambia, es decir, se carga un nuevo documento HTML distinto al que inicialmente había.

Para conseguir esto, es necesario añadir, en el atributo `target` del enlace, el nombre del marco en el que se cargará el documento al que dirige dicho vínculo.



Ejemplo

A continuación, se muestra el código de un enlace que apunta a “`doc4.html`” e indica que, al pulsarlo, dicho documento se cargará en un marco cuyo atributo `name` es `doc2`:

```
...
<a href="doc4.html" target="doc2">Cargar página</a>
...
```



Actividades

30. Vuelva a fijarse en el caso planteado en la aplicación práctica anterior y cree un enlace en el documento “`doc1.html`” que direccione a la página de Google (`www.google.es`), que deberá cargarse en el marco central.

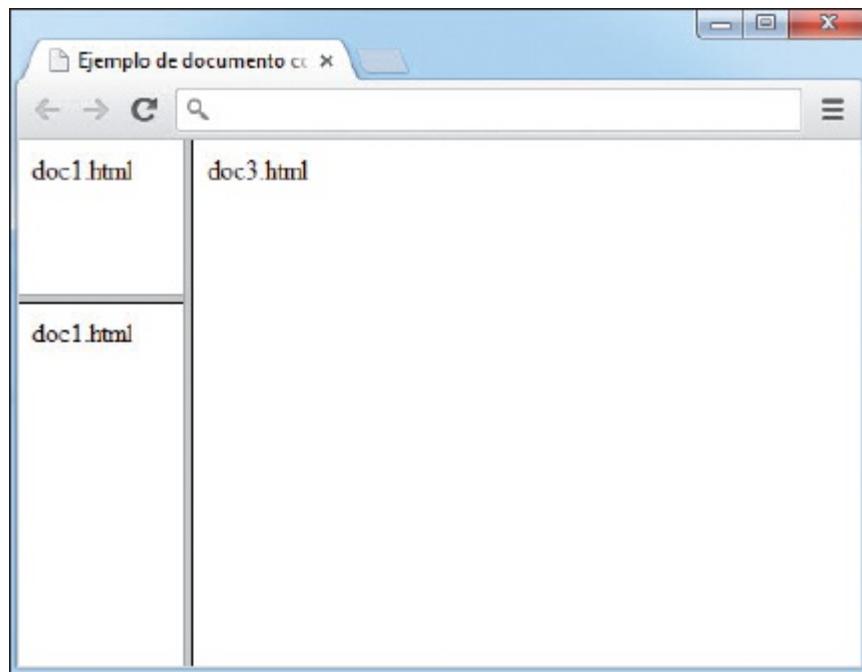
10.5. Marcos anidados

Gracias al anidamiento de marcos se pueden combinar divisiones horizontales y

verticales de estos. Para ello, basta con incluir el nuevo marco dentro de la etiqueta <frameset> del *frame* principal.

A continuación, se muestra un fragmento de código correspondiente a un documento en el que se han establecido dos divisiones verticales (que ocupan, respectivamente el 20 y el 80% del ancho total de la pantalla) y en el que se ha anidado un marco dentro del área más pequeña. Este marco establece dos divisiones horizontales, las cuales ocupan, respectivamente, el 30 y el 70% del alto total de dicha área:

```
...
<frameset cols="20%, 80%">
  <frameset rows="30%,70%">
    <frame name="doc1" src="doc1.html"></frame>
    <frame name="doc2" src="doc1.html"></frame>
  </frameset>
  <frame name="doc3" src="doc3.html"></frame>
</frameset>
...
```



Ejemplo de anidamiento de marcos

10.6. Marcos incrustados (*iframes*)

Los *iframes*, también conocidos como marcos incrustados, consisten en áreas flotantes dentro de las cuales se pueden cargar documentos web. La etiqueta que se utiliza para crear estos elementos es `<iframe>`, que soporta principalmente los siguientes atributos:

- `name`: aquí se indica el nombre del *iframe*.
- `src`: en este atributo se escribe la ruta en la que se localiza el documento local o la URL de la página web a visualizar en el *iframe*.
- `width`: en este atributo se indica el ancho del *iframe* en píxeles.
- `height`: se indica el alto del *iframe* en píxeles.
- `sandbox`: aquí se pueden especificar restricciones de seguridad en el acceso a los contenidos de la web que contiene el *iframe*.
- `scrolling`: este atributo sirve para indicar si el *iframe* va a tener *scroll* o no. Sus posibles valores son *yes* (con *scroll*) y *no* (sin *scroll*).



Recuerde

A excepción de la directiva `<iframe>`, las demás etiquetas que se utilizan para establecer marcos no son soportadas por HTML 5.



Aplicación práctica

Desea crear un documento web que contenga únicamente un *iframe* con las siguientes características:

- Dentro se visualizará la página: <http://www.exlibric.com>.
- Sus dimensiones serán:
 - Ancho: 600 píxeles.
 - Alto: 360 píxeles.
- El marco tendrá *scroll*.

Escriba el código de este documento teniendo en cuenta las siguientes

consideraciones:

- **El título del documento será: “Documento con marco incrustado”.**
- **El documento debe ser compatible con los caracteres propios de Europa Occidental.**

SOLUCIÓN

```
<!DOCTYPE html>
<html>
  <head>
    <title>Documento con marco incrustado</title>
    <meta charset="UTF-8"/>
  </head>
  <body>
    <iframe src="http://www.exlibric.com" width="600"
      height="360" scrolling="Yes"></iframe>
  </body>
</html>
```



Ejemplo de página con marco flotante

11. Formularios

Los formularios son secciones interactivas de una página web que los autores insertan en la mismas para permitir que el usuario pueda enviar información al servidor donde esté alojado el sitio.

En este apartado, se va a estudiar únicamente la manera de construir y configurar formularios en los documentos web, ya que el procesamiento de estos por el servidor es un tema que va más allá del aprendizaje en HTML.

11.1. Descripción general y uso de formularios. Procesamiento de formularios

El uso de formularios ha sido siempre un método muy utilizado para recoger

información introducida por el usuario de la WWW. Una vez que el visitante ha rellenado/marcado los campos correspondientes (y pulsa el botón de envío), dicha información es enviada y procesada en el servidor a través de aplicaciones externas a HTML, como pueden ser: ASP, PHP, JSP, etc. Como se ha comentado antes, el aprendizaje del procesamiento de formularios por medio de estas aplicaciones externas no es un propósito de este manual, aunque la creación de los elementos básicos que pueden formar parte de ellos sí que es un tema importante que está directamente relacionado con el lenguaje HTML.



PHP es un lenguaje de programación del lado del servidor que permite, entre otras muchas cosas, procesar la información enviada a través de formularios.

Todo formulario se crea a partir de la directiva `<form>`, dentro de la cual se van insertando los diferentes elementos que formen parte de él (cajas de texto, listas desplegables, botones, etc.). Esta etiqueta presenta dos atributos que deben incluirse obligatoriamente:

- `action`: este atributo se utiliza para definir el tipo de acción que se va a llevar a cabo una vez que se pulse el botón de envío. Por lo general, contiene la ruta en la que se aloja el documento encargado de procesar los datos introducidos en el formulario.
- `method`: en este atributo, se define la manera en la que se van a enviar los datos introducidos, pudiendo ser: `GET` y `POST` (este último es el más usual).

También existen otros atributosopcionales:

- `enctype`: para describir el método que se va a utilizar para codificar el contenido del formulario. Sus posibles valores son: `application/x-www-form-urlencoded`, `multipart/form-data` y `text/plain`.
- `name`: para identificar el formulario mediante un nombre.
- `target`: especifica el lugar donde se van a visualizar los resultados del envío del formulario. Sus posibles valores son: `_blank`, `_self`, `_parent` y `_top`.
- `accept-charset`: aquí se especifica la codificación de caracteres que es aceptada por el servidor que va a procesar el formulario.

11.2. Elementos de un formulario: texto, botones, etc.

La directiva `<form>` actúa como si se tratase de un contenedor que alberga un conjunto de elementos que permiten al usuario marcar o introducir datos. Existen tres etiquetas que sirven para crear elementos de formulario. Estas son `<input>`, `<textarea>` y `<select>`.

`<input>`

Esta directiva sirve para crear botones y casillas de texto. Su sintaxis básica es la siguiente:

```
<input type="..." value="..." name="..."/>
```

El atributo `name` sirve para identificar el elemento cuando su información es enviada por el método correspondiente. El atributo `value` sirve para indicar un valor predeterminado para el elemento en cuestión, es decir, el valor que está escrito o marcado antes de que el usuario manipule el formulario. Con el atributo `type` se puede indicar el tipo de elemento que se desea insertar, pudiendo tener los siguientes valores:

- `checkbox`: consiste en una casilla de selección, la cual podrá estar seleccionada o no. El atributo `value` contiene el valor que se enviará en el caso de que la casilla esté seleccionada.



Casilla de selección

Ejemplo de `checkbox`

- `hidden`: este campo consiste en un almacén interno de datos cuya información no puede ser modificada por el usuario, es decir, es un campo oculto cuyo valor es enviado siempre junto con el resto de los datos.
- `file`: este elemento permite el envío de ficheros a través del formulario y consiste en un campo de texto junto al cual se dispone automáticamente un botón “Examinar...”, que permite seleccionar manualmente, desde el explorador del

sistema operativo que corresponda, el archivo a enviar.

Nota: Cuando se utilicen este tipo de elementos, es necesario incluir, en el atributo `enctype` de la directiva `<form>`, el valor `multipart/form-data`.

A screenshot of a web browser showing a file input field. The field has a placeholder text "No se ha seleccionado ningún archivo" and a button labeled "Seleccionar archivo".

Ejemplo de elemento `file`

- `image`: corresponde a un botón de envío personalizado que consiste en una imagen definida por el atributo `src`.
- `password`: consiste en una casilla de texto cuyos caracteres escritos aparecen como asteriscos (o cualquier otro símbolo, lo cual depende del navegador) para ocultar el texto que se escribe.

A screenshot of a web browser showing a password input field. The label "Contraseña:" is followed by a text input field containing masked text (asterisks).

Ejemplo de elemento `password`

- `radio`: elemento similar a `checkbox`, pero asigna el mismo valor de `name` a varios elementos. Esto significa que, cuando existen varios elementos de radio que tienen el mismo valor en su atributo `name`, solo uno de ellos puede estar marcado a la vez. Al igual que ocurre con `checkbox`, el atributo `value` de estos elementos contiene el valor que se enviará en el caso en el que se seleccionen.

A screenshot of a web browser showing a radio button element. It consists of a small circle and the word "Casilla".

Ejemplo de elemento `radio`

- `reset`: este botón sirve para borrar todos los datos introducidos por el usuario, restaurando el formulario a su estado inicial.

A screenshot of a web browser showing a "Borrar" (Delete) button.

Ejemplo de botón `reset`

- `submit`: botón de envío del formulario. El texto que aparece escrito sobre el botón puede definirse en su atributo `value`.



Ejemplo de botón de envío de formulario

- `text`: casilla de texto que sirve para escribir en ella. El tamaño de la casilla en caracteres puede definirse con el atributo `size`. También es posible limitar la cantidad de texto que puede escribirse (en caracteres) con el atributo `maxlength`. Estos dos atributos también son aplicables al elemento `password`.

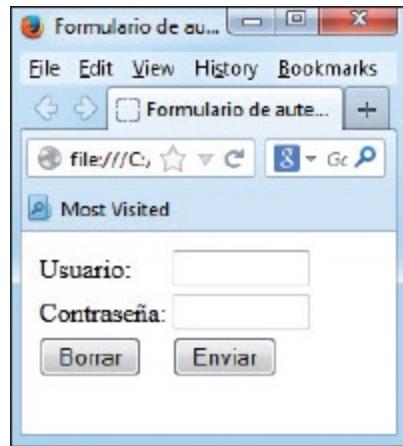
Campo de texto:	<input type="text" value="Escribe aquí el texto"/>
-----------------	--

Ejemplo de casilla de texto

Aplicación práctica. Creación de un formulario (I)

Desea crear un documento (“index.html”) que contenga un formulario que sirva para que un usuario se autentifique en la web mediante sus datos de usuario y contraseña. Escriba el código del documento teniendo en cuenta las siguientes consideraciones:

- En formulario se enviará por el método *POST* a un documento llamado “envio.php” ubicado en el mismo directorio que “index.html”.
- El tamaño de las cajas de texto de usuario y contraseña será de 10 caracteres, siendo este también el límite de caracteres que puede escribirse en ellos.
- El nombre de las cajas de texto de usuario y contraseña será, respectivamente: *user* y *pass*.
- El formulario dispondrá de dos botones: uno de *reset* (`name=borrar`) y otro de *submit* (`name=enviar`).
- El título del documento será: “Formulario de autenticación”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.
- El formulario deberá visualizarse como se muestra en la siguiente imagen (consejo: utilice una tabla para colocar los textos, cajas y botones):



Visualización de “index.html” en el navegador Mozilla Firefox

Solución

```
<!DOCTYPE html>
<html>
    <head>
        <title>Formulario de autentificación</title>
        <meta charset="UTF-8"/>
    </head>
    <body>
        <form method="post" action="envio.php">
            <table>
                <tr>
                    <td>Usuario: </td>
                    <td><input type="text" value="" name="user" size="10" maxlength="10"/></td>
                </tr>
                <tr>
                    <td>Contraseña: </td>
                    <td><input type="password" value="" name="pass" size="10" maxlength="10"/></td>
                </tr>
                <tr>
                    <td><input type="reset" value="Borrar" name="borrar"/></td>
                    <td><input type="submit" value="Enviar" name="envio"/></td>
                </tr>
            </table>
        </form>
    </body>
</html>
```



Actividades

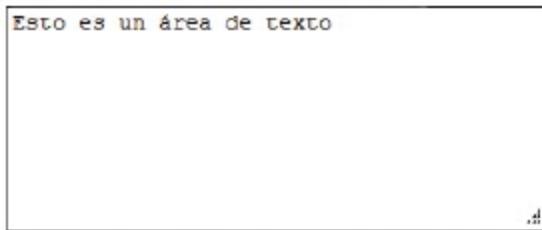
31. ¿Qué modificaciones podría hacer en el código del documento anterior si deseara añadir dos elementos radio que sirvan para indicar el sexo del usuario que se está autenticando?

<textarea>

Este elemento es similar a *text* con la diferencia de que, en <textarea>, el usuario puede escribir textos más extensos, es decir, de varias filas y columnas. Sus principales atributos son:

- `name`: identifica el <textarea>.
- `rows`: indica el número de filas que va a ocupar la caja de texto.
- `cols`: indica el número de columnas que va a ocupar la caja de texto.

Para indicar un texto inicial que se desea que aparezca en la caja de texto, es necesario escribirlo entre la apertura y el cierre de esta directiva (a diferencia de los elementos *text*, en los que se utiliza el atributo `value`).



Ejemplo de área de texto



Actividades

32. Identifique y corrija los errores del siguiente fragmento de código correspondiente a un área de texto de un formulario:

```
<input type="textare" cols="20" rows="5">Esto es un  
textarea</input>
```

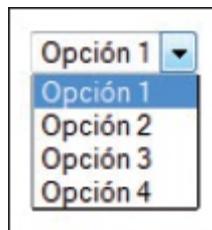
<select>

Estos elementos consisten en listas desplegables de opciones que son mostradas al hacer clic en un pequeño botón adjunto a los mismos. Las diferentes opciones que van a formar parte de estas listas son definidas, dentro de <select>, usando directivas <option>, por lo que la sintaxis básica de estos elementos es:

```
...
<select name="">
    <option value="">...</option>
    <option value="">...</option>
    <option value="">...</option>
    ...
</select>
...
```

Al igual que en el resto de elementos de formulario, el atributo `name` de esta directiva permite identificar al elemento dentro del formulario. Por otro lado, con el atributo `value` de la etiqueta <option>, se indica el valor que será enviado al seleccionar la opción correspondiente. Para especificar el texto de cada una de las opciones, basta con escribirlo entre la apertura y cierre de la correspondiente directiva <option>.

También es posible añadir, en alguna de las opciones, el atributo `selected` (sin valor), el cual indica que dicho elemento será el que se vea antes de desplegar la lista.



Ejemplo de lista desplegable

Aplicación práctica. Creación de un formulario (II)

Se dispone a crear un documento HTML (“index.html”) en el que desea incluir un formulario constituido por una lista desplegable (`name="lista"`), un área de texto (`name="descripcion"`) y un botón de envío (`name="envio"`). Escriba el código del documento teniendo en cuenta las siguientes consideraciones:

- El formulario se enviará por el método *POST* a un documento llamado “envio.php” ubicado en el mismo directorio que “index.html”.
- La lista desplegable contendrá las siguientes opciones: Deporte (`value="d"`), Vídeojuegos (`value="v"`), Moda (`value="mo"`), Música (`value="mu"`). Este último es el que se visualizará antes de desplegar la lista.
- La caja de texto tendrá una extensión de 30 columnas y 10 filas.
- El título del documento será: “Ejemplo de formulario”.
- El documento debe ser compatible con los caracteres propios de Europa Occidental.
- El formulario deberá visualizarse como se muestra en la siguiente imagen (consejo: utilice una tabla para colocar los diferentes elementos):

The screenshot shows a Mozilla Firefox browser window titled "Formulario de ejemplo - Mozilla Firefox". The window contains a form with the following elements:

- A dropdown menu labeled "Selecciona tu preferencia:" with the value "Música" selected.
- A large text area labeled "Háblanos sobre tí:".
- A "Enviar" button at the bottom of the form.

Ejemplo de formulario con lista desplegable y área de texto

Solución

```

<!DOCTYPE html>
<html>
    <head>
        <title>Formulario de ejemplo</title>
        <meta charset="UTF-8"/>
    </head>
    <body>
        <form method="post" action="envio.php"
            enctype="multipart/form-data">
            <table>
                <tr>
                    <td>Selecciona tu preferencia:</td>
                    <td>
                        <select name="lista">
                            <option value="d">Deporte</option>
                            <option value="v">Videojuegos</option>
                            <option value="mo">Moda</option>
                            <option value="mu" selected>Música</option>
                        </select>
                    </td>
                </tr>
                <tr>
                    <td>Háblanos sobre ti:</td>
                    <td><textarea rows="10" cols="30"></textarea></td>
                </tr>
                <tr>
                    <td></td>
                    <td><input type="submit" name="envio"
                        value="Enviar"/></td>
                </tr>
            </table>
        </form>
    </body>
</html>

```

11.3. Formateado de formularios: atajos de teclado, orden de edición,

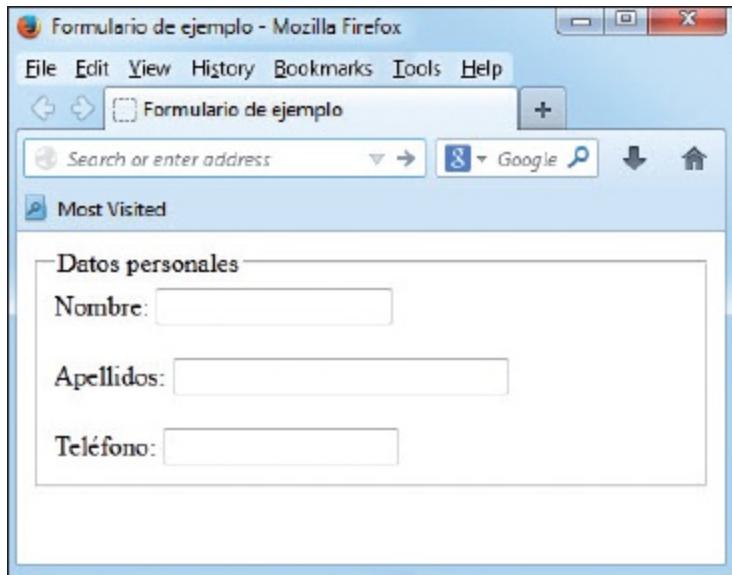
grupos, etiquetas, etc.

A continuación, se presentan algunas consideraciones y características adicionales relacionadas con la visualización y funcionalidad de los formularios que se insertan en los documentos web:

- El orden en el que se declaren los elementos dentro del código del documento va a influir directamente en el orden de aparición de los mismos dentro del área de navegación. No obstante, la aplicación de ciertas propiedades y estilos CSS sobre dichos elementos puede hacer que la disposición de los mismos varíe radicalmente.
- Los diferentes elementos que constituyen los formularios pueden ser agrupados con la directiva `<fieldset>`. Dentro de esta, puede incluirse la etiqueta `<legend>` para dar nombre a dicha agrupación. Ejemplo:

```
...
<form ....>
  <fieldset>
    <legend>Datos personales</legend>
    ...
    <fieldset>
  </form>
  ...

```



Ejemplo de formulario con sus elementos agrupados

- Al igual que en los enlaces, los elementos de un formulario pueden asociarse a alguna tecla de acceso rápido, utilizando el atributo `acceskey`.



Recuerde

El atributo `acceskey` sirve para definir un atajo de teclado que servirá para acceder directamente al elemento en cuestión. Al pulsar la tecla [Alt] junto con la definida en este atributo, el navegador dirigirá al usuario al lugar donde se encuentre dicho elemento dentro de la web.



Actividades

33. Escriba el código de un documento web en el que se visualice un formulario lo más parecido posible al presentado en la imagen anterior (formulario agrupado). Asocie las teclas `[n]`, `[a]` y `[t]` a cada una de las cajas de texto (de arriba abajo).

12. Elementos específicos para tecnologías móviles

Cuando se diseña un sitio web, cada vez es más necesario tener en cuenta el hecho de que debe estar optimizado también para dispositivos móviles. En este apartado se van a enumerar algunas nociones básicas a tener en cuenta en el diseño y desarrollo web orientado hacia estos dispositivos.

12.1. Selección del lenguaje marcas para tecnologías móviles

En la actualidad, la inmensa mayoría de los dispositivos móviles soportan casi todas las etiquetas correspondientes a la nueva especificación HTML5. No obstante, sí que es importante tener en cuenta una serie de recomendaciones que permitirán mejorar la optimización de las páginas web para los dispositivos móviles. Algunas de ellas son:

- Diseñar las páginas del sitio de manera que se amolden a diferentes resoluciones y tamaños de pantalla. Lo ideal es establecer diseños líquidos para los distintos elementos que van a formar parte de las páginas web, de manera que estos se adapten al área de visualización disponible en cada caso.



Las pantallas de los dispositivos móviles suelen tener resoluciones y tamaños de pantalla mucho menores que los de los monitores de los ordenadores de sobremesa.

- Utilizar etiquetas específicas de HTML5, ya que generalmente los navegadores de estos dispositivos las soportan.

- Utilizar etiquetas <meta> específicas relacionadas con la visualización del documento en los dispositivos móviles. Algunas de ellas son:
 - <meta name="MobileOptimized" content="width"/>: al incluir esta línea en la cabecera de la página, el navegador adaptará la anchura de la web a la pantalla del dispositivo que la visualice.
 - <meta name="HandheldFriendly" content="true"/>: esta línea informa al navegador acerca de que la web ha sido optimizada para dispositivos móviles, por lo que no se debe escalar su contenido al visualizarlo en este tipo de pantallas.
 - <meta name="viewport" content="width=device-width, initial-scale=1">: esta línea de código informa al navegador acerca de que la página está optimizada para dispositivos móviles, en la que también se puede especificar la manera en la que el navegador debe interpretar y renderizar el documento. Esta etiqueta sirve para indicar diversas informaciones al navegador, como la anchura a la que se tiene que ver el documento, la escala inicial, si permite hacer zoom, etc.

12.2. Hojas de estilo en dispositivos móviles

La última versión de CSS (CSS3) incorpora las denominadas *Media Queries*, que consisten en una serie de condiciones y requisitos que permiten filtrar estilos CSS dependiendo de un rango de resolución de pantalla determinado.

Las *Media Queries* se pueden incluir dentro de los archivos “.css” con la siguiente sintaxis:

```
@media screen and (max-width:...) {  
    ...  
}
```

De esta manera, los estilos CSS añadidos dentro de la *Media Query* ({ ... }) se aplicarán a los documentos web cuando la resolución de pantalla no sea superior al valor indicado en la propiedad `max-width`.



Ejemplo

En el siguiente ejemplo, el estilo `p {width:100%;}`, solo aplicará cuando la resolución de pantalla sea inferior a 500 píxeles:

```
...
@media screen and (max-width: 500px) {
  p {width:100%;}
}
...
```

Una *Media Query* también se puede usar para cargar un archivo “.css” únicamente bajo determinadas resoluciones, en cuyo caso es necesario añadirla dentro de la cabecera del documento web que corresponda (`<head>`), usando la directiva `<link>`. A continuación, se muestra un ejemplo de uso de una *Media Query* para este cometido:

```
...
<link rel="stylesheet" type="text/css" media="only screen
and (min-device-width: 200px) and (max-device-width: 600px)"
href="mobile.css" />
...
```

En este ejemplo, el archivo “mobile.css” se cargará en aquellos casos en los que la pantalla del dispositivo tenga un ancho comprendido entre 200 y 600 píxeles.

13. Elementos en desuso (deprecated)

Existe una extensa colección de elementos que desaparecen definitivamente en la nueva especificación HTML5. En este apartado, se describen algunos de los más importantes.

13.1. Texto parpadeante

Los textos parpadeantes se solían usar para establecer palabras o frases destacadas, las cuales se visualizaban con un efecto de parpadeo bastante llamativo. Para ello, se solía utilizar la etiqueta no estándar `<blink>`, dentro de la cual se escribía el texto que se deseaba mostrar con este efecto. Por ejemplo:

```
...
<blink>Este es el texto que parpadea</blink>
...
```



Sabía que...

A diferencia de otros elementos no soportados por la nueva especificación HTML5, la directiva `<blink>` ya no es reconocida por la mayoría de los navegadores actuales.

13.2. Marquesinas

Existía una etiqueta, la cual no pertenecía al ningún estándar HTML, que servía para insertar marquesinas en los documentos web: `<marquee>` y dentro de ella se escribía el texto que se deseaba establecer como marquesina.

Dicha etiqueta permitía establecer textos que se desplazaban de un lado a otro de la pantalla y disponía de ciertos atributos que permitían configurar y modificar este comportamiento.

Al principio, la etiqueta `<marquee>` únicamente era soportada por el navegador *Internet Explorer*, lo cual hacía que el uso de este elemento fuera totalmente desaconsejable. No obstante, en la actualidad, la mayoría de los navegadores interpretan correctamente esta etiqueta.



Consejo

Debido a que esta etiqueta no es soportada por la nueva especificación HTML5, es aconsejable que, en su lugar, se utilicen imágenes en formato GIF o bien animaciones *JavaScript*.

13.3. Alineaciones

La etiqueta `<center>` servía para delimitar un bloque de texto que se deseaba que apareciera centrado. A continuación, se muestra un ejemplo de uso de esta etiqueta:

```
...
```

```
<center>Este es el texto centrado.</center>
```

```
...
```

Gracias a las hojas de estilo CSS, se pueden establecer diferentes alineaciones en los textos de los documentos web. Precisamente el uso de hojas de estilo es lo que recomienda la nueva especificación HTML5 para establecer cualquier alineación en los textos.



Nota

La etiqueta `<center>` todavía es reconocida por la mayoría de los navegadores actuales.

13.4. Otros elementos en desuso

Existen muchos otros elementos que han desaparecido en la nueva versión del estándar. A continuación, se enumeran brevemente algunos de los más importantes:

- `<frame>` y `<frameset>`: como se ha visto anteriormente, estas etiquetas servían para incluir marcos en los documentos web.

- <applet>: permitía introducir programas *Java* (llamados *applets*) dentro de una página web.
- : servía para establecer diferentes características en los textos que contenían (color, fuente, tamaño, etc.).
- <basefont>: el cometido de esta etiqueta era establecer el tamaño por defecto de la letra de los documentos web.



Recuerde

La nueva especificación HTML5 no soporta el uso de marcos o *frames*. En su lugar, recomienda el uso de CSS para posicionar y dar estilo a las diferentes partes de los documentos que, antes de la aparición de este estándar, se solían establecer mediante el uso de marcos (barra de navegación, menús laterales, pie de página etc.).

14. Resumen

HTML es el lenguaje de marcas a partir del cual se desarrollan los documentos web. Cuando una página se visualiza en el navegador, realmente lo que se está viendo es una traducción visual del código HTML (y CSS) que el desarrollador ha implementado en el documento.

Para crear una página web, es imprescindible conocer el lenguaje HTML, que, gracias a su sencillez, puede ser dominado en muy poco tiempo. No obstante, también es importante saber cómo aplicar estilos utilizando CSS, ya que proporciona infinidad de posibilidades (algunas de ellas imposibles de conseguir con la única utilización de etiquetas HTML) relacionadas con la apariencia de los elementos que constituyen los documentos web.

En HTML, existe un gran número de etiquetas diferentes y algunas de ellas deben estar presentes en todos los documentos web, como son <DOCTYPE>, <html>, <body>, <title>, etc.

Otras directivas permiten establecer estilos a los textos de los documentos web, aunque no se recomienda su utilización. Es su lugar, la nueva especificación HTML5 aconseja utilizar hojas de estilo CSS.

Con HTML también es posible manejar y configurar otras muchas etiquetas, que permiten enriquecer los documentos web con multitud de elementos. Algunos de los más importantes son: enlaces, párrafos de texto, imágenes, tablas, formularios, etc.



Ejercicios de repaso y autoevaluación

1. El lenguaje HTML es un lenguaje...

- a. ... interpretado.
- b. ... compilado.
- c. ... similar a Java, C++, etc.
- d. Todas las respuestas anteriores son correctas.

2. Complete el siguiente texto.

La _____ línea de código que hay que escribir en un documento HTML es <!DOCTYPE html>, que consiste en un _____ especial que indica al navegador que el documento a interpretar está escrito en _____.

3. De las siguientes afirmaciones, diga cuál es verdadera o falsa.

La directiva <head> debe incluirse en cualquier documento escrito en HTML.

- Verdadero
- Falso

Para definir el juego de caracteres a utilizar en la visualización de un documento web, es necesario utilizar la directiva <link>.

- Verdadero
- Falso

<aside> es una directiva nueva en HTML5.

- Verdadero
- Falso

4. En HTML, los colores pueden expresarse...

- a. ... en RGB.
- b. ... en CMYK.

- c. ... con su denominación en inglés.
- d. Las respuestas a. y c. son correctas.

5. Complete el siguiente texto.

Para establecer importancia en los textos de los documentos web se pueden utilizar las etiquetas de _____. De estas, la que más importancia establece en el texto es _____.

6. ¿Qué son los estilos lógicos y físicos? ¿Para qué se utilizan?

7. ¿Cuál de las siguientes etiquetas se utiliza para dar énfasis a los textos de los documentos web?

- a. .
- b. .
- c. <sub>.
- d. .

8. ¿En qué se diferencian fundamentalmente los enlaces absolutos y los relativos?

9. Complete el siguiente texto.

El formato _____ es el más utilizado a la hora de insertar _____ en las páginas web. Esto se debe fundamentalmente a la excelente relación calidad-_____ que presentan los archivos de imagen de este tipo.

10. Señale la etiqueta que es necesario utilizar cuando se implementan mapas de imágenes en HTML.

- a. <map>.
- b. <area>.

- c. .
- d. Todas las respuestas anteriores son correctas.

11. De las siguientes afirmaciones, diga cuál es verdadera o falsa.

HTML5 recomienda utilizar el atributo `background` para establecer imágenes de fondo en los documentos web.

- Verdadero
- Falso

Para crear listas ordenadas es necesario hacer uso de la directiva ``.

- Verdadero
- Falso

Las listas no ordenadas no soportan anidamientos.

- Verdadero
- Falso

12. Complete el siguiente texto.

Cuando se inserta una tabla en HTML, el contenido de la misma debe definirse dentro de la etiqueta _____. El orden de construcción de estos elementos se establece de la forma: fila-_____ de la fila, es decir, se van definiendo cada una de las filas de la tabla (con la directiva `<tr>`) y dentro de ellas se van insertando las diferentes _____ (con la etiqueta _____) que pertenecen a cada una de dichas filas.

13. Señale la directiva que sirve para establecer contenidos que aparecerán cuando se estén utilizando marcos y estos no se puedan visualizar en el navegador.

- a. <noframes>.
- b. <frameset>.
- c. <frame>.
- d. <iframe>.

14. Defina brevemente el concepto de formulario web.

15. De las etiquetas que se muestran a continuación, señale aquella que sea soportada por la nueva especificación HTML5.

- a. <iframe>.
 - b. <applet>.
 - c. .
 - d. <center>.
-

Capítulo 4

Hojas de estilo web

1. Introducción

La aparición de las hojas de estilo en cascada (CSS o *Cascading Style Sheet*) supuso uno de los avances más importantes que se han hecho en el ámbito del diseño y desarrollo web profesional.

Las hojas de estilo en cascada son un conjunto de reglas que definen el aspecto de los elementos que forman parte de los documentos web, cuya estructura ha sido definida previamente con HTML.

Como se ha ido viendo a lo largo del presente manual, HTML es un lenguaje sencillo que permite la creación rápida de documentos web con contenidos y estilos realmente variados. No obstante, el mantenimiento y desarrollo de un sitio web utilizando únicamente este lenguaje puede resultar una tarea muy frustrante, sobre todo en el caso de sitios web con muchos documentos y en los que se desee modificar, por ejemplo, el aspecto de uno o varios de ellos, estando estos presentes en la mayoría de los documentos del sitio. Esta tarea puede resultar ser muy sencilla y rápida utilizando hojas de estilo, pero, sin ellas, sería necesario modificar cada uno de los documentos en los que apareciera el elemento cuyo aspecto se deseé cambiar.

En este capítulo, se van a estudiar los fundamentos de las hojas de estilo CSS, aprendiéndose su sintaxis, diversas maneras de incluirlas en los documentos y algunas propiedades básicas que permitirán modificar el aspecto de muchos de los elementos que pueden formar parte de los documentos web.

2. Elementos y estructura de una hoja de estilos

En este apartado, se van a estudiar las maneras de definir y aplicar estilos CSS a los elementos de los documentos web, así como el funcionamiento de las hojas de estilo y la manera de utilizarlas para modificar el formato y la estructura de los documentos web.

2.1. Creación de hojas de estilo

Como ya se ha comentado en capítulos anteriores, existen tres maneras fundamentales de definir estilos CSS en un documento web:

- Mediante el atributo `style`.
- Mediante la etiqueta `<style>`.
- Mediante un archivo “`.css`” externo.

Definición de estilos mediante el atributo `style`

Este método consiste simplemente en definir los estilos CSS utilizando el atributo `style`, que se incluye en el elemento en el que se desean aplicar. Para ello, es necesario tener en cuenta la siguiente sintaxis:

```
...
<etiqueta style="propiedad1:valor1;propiedad2:
valor2...">...</etiqueta>
...
```

Por ejemplo, con el siguiente código se establece color (`color`) rojo (`#ff0000`) y alineación (`text-align`) justificada (`justify`) a un párrafo de un documento web:

```
...
<p style="color:#ff0000;text-align:justify">Esto es un párrafo</p>
...
```



Actividades

1. Cree un documento en el que aparezca un párrafo de color azul (`#0000ff`) y cuyo texto aparezca alineado a la derecha (`right`). Utilice para ello el

atributo style.

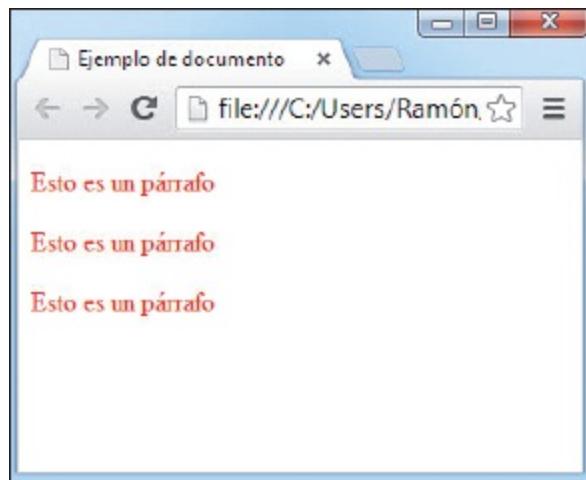
Definición de estilos mediante la directiva <style>

Otra manera de definir estilos CSS en un documento web consiste en incluirlos dentro de la etiqueta <style>, la cual debe ubicarse dentro de la cabecera del documento <head>. A continuación, se muestra la sintaxis de este método:

```
...
<head>
...
<style type="text/css">
    selector1 {
        propiedad1:valor1;
        propiedad2:valor2;
        ...
    }
    selector2 {
        propiedad1:valor1;
        propiedad2:valor2;
        ...
    }
    ...
</style>
...
</head>
...
```

A continuación, se muestra un ejemplo de uso de este método para aplicar los mismos estilos del ejemplo anterior a todos los párrafos presentes en un documento web:

```
...
<head>
...
    <style type="text/css">
        p {
            color:#ff0000;
            text-align:justify;
        }
    </style>
...
</head>
<body>
    <p>Esto es un párrafo</p>
    <p>Esto es un párrafo</p>
    <p>Esto es un párrafo</p>
</body>
...
```



Ejemplo de documento HTML en el que se han utilizado estilos CSS para modificar la alineación y el color del texto de tres párrafos.



Nota

Los selectores CSS indican los elementos sobre los que se van a aplicar los estilos definidos dentro de ellos. Más adelante, se estudiará cómo funcionan y cuáles son los tipos más utilizados.

Definición de estilos mediante un archivo “.css” externo

Este es el método más utilizado para definir los estilos CSS que van a afectar a los elementos de un sitio web completo, el cual consiste en implementarlos en un archivo “.css” aparte. Para enlazar este archivo a cada uno de los documentos web en los que se desee aplicar los estilos definidos en dicho archivo “.css”, es necesario utilizar la directiva `<link>`, tal y como se estudió en capítulos anteriores.



Recuerde

La directiva `<link>` se utiliza, sobre todo, para adjuntar hojas de estilo CSS a los documentos web. Para este caso, la sintaxis de esta etiqueta es la siguiente:

```
<link rel="stylesheet" href="..."/>
```

Con el atributo `rel`, se especifica el tipo de relación que existe entre el documento web en cuestión y el archivo externo que se está incorporando, cuya localización se especifica en el atributo `href`. Si el archivo de estilos se encuentra en el mismo directorio que el documento HTML correspondiente, el valor del atributo `href` coincidirá con el nombre del archivo “.css”.

Por ejemplo, si se deseara vincular una hoja de estilos llamada “estilo.css” a un documento (“index.html”), estando la primera ubicada en una carpeta llamada “css”, la sintaxis de la correspondiente directiva `<link>` sería (teniendo en cuenta que “index.html” se encuentra en la raíz del sitio, al igual que el directorio “css”):

```
...
<head>
...
<link rel="stylesheet" href="css/estilo.css"/>
...
</head>
...
```

Los archivos “.css” son documentos de texto plano (al igual que los documentos “.html”) dentro de los cuales se definen los distintos estilos con la misma sintaxis que cuando se escriben dentro de la directiva `<style>`, pero sin necesidad de añadir esta etiqueta (ni ninguna otra). En definitiva, cuando se definen estilos CSS en archivos externos, únicamente hay que escribir cada uno de los selectores y las propiedades que se especifican en ellos (incluidas entre llaves `{...}`). Por ejemplo, el siguiente código corresponde al contenido de una hoja de estilos en la que se establece un color (#cccccc) al texto de los enlaces y párrafos de los documentos en los que dicha hoja se vincula:

```
a {
    color: #cccccc;
}
p {
    color: #cccccc;
}
```

2.2. Aplicación de estilos. Selectores

Los selectores se utilizan cuando se crean estilos CSS en un archivo externo (.css) o bien cuando se incluyen dentro de la directiva `<style>` (en la cabecera de los documentos web e informan acerca de qué elemento o elementos van a ser los beneficiarios de los estilos que se definen en los mismos).

Aunque existen muchos más (sobre todo en la nueva especificación CSS3), la

mayoría de los sitios web se pueden diseñar utilizando únicamente cuatro tipos de selectores básicos: selectores de etiquetas, selectores descendentes, selectores de clase y selectores id.

Selectores de etiquetas

Cuando se definen estilos CSS para un selector de etiquetas, dichos estilos se aplican a todos los elementos del documento web cuya directiva coincida con el nombre del selector. La sintaxis básica del uso de este tipo de selectores es:

```
...
etiqueta {
    propiedad1: valor1;
    propiedad2: valor2;
    propiedad3: valor3;
}
...
```

Por ejemplo, el siguiente fragmento de código corresponde a un selector de etiquetas (`p`) cuyos estilos (`margin-top:5px;text-decoration:underline`) se aplicarán a todos los párrafos (`<p>`) del documento en cuestión:

```
...
p {
    margin-top:5px;
    text-decoration: underline;
}
...
```

En el caso en el que se deseara aplicar los mismos estilos CSS a varios selectores de etiquetas distintos, bastaría con separar dichos selectores por comas. Por ejemplo, con

el siguiente fragmento de código se aplican dos estilos CSS (`color:red; text-align:center`) a todos los párrafos y cabeceras `<h1>` presentes en el documento (o documentos) en cuestión:

```
...
p, h1 {
    color:red;
    text-align:center;
}
...
```

Selectores descendentes

Estos selectores se utilizan cuando se desean aplicar estilos CSS a elementos que están dentro de otros, siendo su sintaxis la siguiente:

```
...
selector1 selector2 ... selectorn {
    propiedad1: valor1;
    propiedad2: valor2;
    propiedad3: valor3;
}
...
```

Por ejemplo, las propiedades CSS declaradas en el siguiente fragmento de código se aplicarán a todos aquellos enlaces que se encuentren dentro de un párrafo:

```
...
p a {
    color: red;
    text-decoration: none;
}
...
```



Ejemplo

También es posible indicar más de dos selectores descendentes. Por ejemplo, las propiedades CSS declaradas en el siguiente fragmento de código únicamente se aplicarán a los enlaces que se encuentren dentro de párrafos y que estos, a su vez, se ubiquen dentro de una directiva `<aside>`:

```
...
aside p a {
    color: #aabbc;
    text-decoration: underline;
}
...
```



Actividades

2. ¿En qué elementos de un documento HTML se aplicarán las propiedades CSS definidas en el siguiente fragmento de código?

```
body nav a {
    color:#ffffff;
    font-size:1.5em;
}
```

Selectores de clase

Los estilos CSS definidos en un selector de clase se aplican a todos aquellos elementos web que tengan dicha clase asociada en su atributo class (sin el carácter “.”). A continuación, se muestra la sintaxis básica del uso de este tipo de selectores:

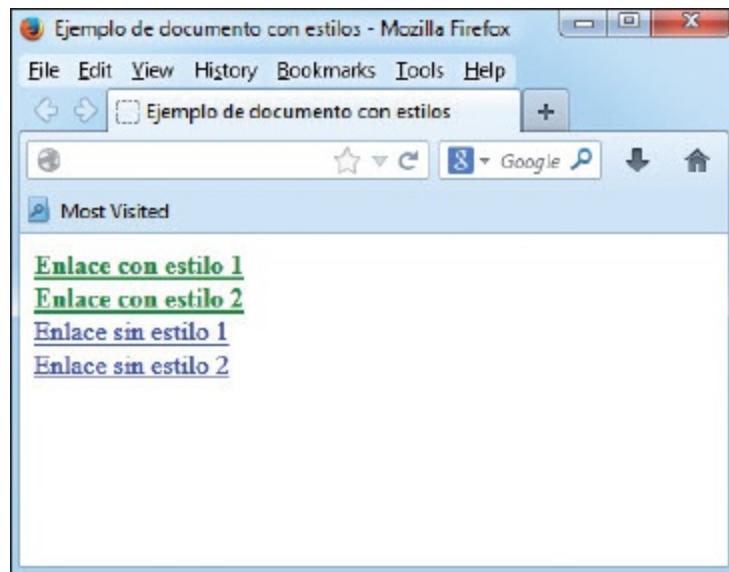
```
...
.clase {
    propiedad1: valor1;
    propiedad2: valor2;
    propiedad3: valor3;
}
...
```

A continuación, se muestra un ejemplo de código correspondiente a un documento HTML en el que se han establecido dos propiedades CSS (`color:green; font-weight:bold;`) que se aplican a los enlaces que tienen asociada la clase `enlace_con_estilo`.

```

<!DOCTYPE html>
<html>
    <head>
        <title>Ejemplo de documento con estilos</title>
        <meta charset="UTF-8"/>
        <style type="text/css">
            .enlace_con_estilo {
                color:green;
                font-weight:bold;
            }
        </style>
    </head>
    <body>
        <a href="documento1.html" class="enlace_con_estilo">Enlace con estilo 1</a><br/>
        <a href="documento1.html" class="enlace_con_estilo">Enlace con estilo 2</a><br/>
        <a href="documento1.html">Enlace sin estilo 1</a><br/>
        <a href="documento1.html">Enlace sin estilo 2</a><br/>
    </body>
</html>

```



Ejemplo de documento web en el que se han establecido estilos CSS en dos de sus enlaces



Ejemplo

Los selectores de clase también se pueden utilizar como selectores descendentes. Por ejemplo, las propiedades CSS definidas a continuación se aplicarán a todos los elementos que tengan asociada la clase “parrafo_estilo” y que, a su vez, se localicen dentro de etiquetas <article>:

```
article .parrafo_estilo {  
    text-align: center;  
    font-size:0.5em;  
}
```

Con CSS también es posible definir estilos que se apliquen a elementos que tengan asociada una clase determinada y que, a su vez, pertenezcan a un tipo de elemento concreto (etiqueta). Para ello, basta con escribir dicha etiqueta justo delante del selector de clase que corresponda. Por ejemplo, la propiedad CSS que se muestra a continuación se aplicará a todos los párrafos que tengan asociado en su atributo `class` el valor “parrafo_negrita”.

```
...  
p.parrafo_negrita {  
    font-weight:bold;  
}  
...
```



Nota

Con CSS también se puede definir más de una clase a un elemento web concreto. Para ello, basta con indicar, en su atributo `class`, cada una de las clases que se

desean aplicar a dicho elemento separadas con espacios. Por ejemplo:

```
<a href="documento1.html" class="clase1 clase2  
clase3">Esto es un enlace con varias clases</a>
```

Selectores id

Este tipo de selectores se utilizan para definir estilos únicos que se aplicarán a un elemento web concreto. La sintaxis básica del uso de este tipo de selectores es:

```
...  
#identificador {  
    propiedad1: valor1;  
    propiedad2: valor2;  
    propiedad3: valor3;  
}  
...
```

Los estilos definidos en un selector `id` se aplican al elemento cuyo valor de su atributo `id` coincida con el nombre del selector (sin el carácter “#”). Por ejemplo, los estilos definidos a continuación se aplicarán al elemento cuyo atributo `id` tenga asignado el valor “identificador”:

```
...
#identificador {
    font-family: "New Century Schoolbook", Times, serif;
    word-spacing: 0.6em;
}
...
```



Nota

Los selectores id también pueden utilizarse como selectores descendentes.

2.3. Herencia de estilos y aplicación en cascada

Las hojas de estilo CSS se denominan hojas de estilo en cascada por varias razones:

- Las propiedades CSS pueden estar escritas en diferentes lugares (dentro de un atributo `style`, en un archivo “.css” externo, etc.) y, dependiendo del sitio donde se encuentren, se aplicarán con más o menos prioridad.
- Cuando un elemento se encuentra ubicado dentro de otro (por ejemplo un párrafo `<p>` dentro de un elemento `<article>`), el elemento hijo (`<p>`) también es afectado por las propiedades definidas para el elemento padre (`<aside>`).
- Un mismo elemento puede ser afectado por propiedades CSS definidas en más un selector (por ejemplo, a un enlace `<a>` con clase “link”, le afectan las propiedades CSS definidas para los selectores `a` y `.link`).
- Si las propiedades CSS que se han especificado no entran en conflicto, el navegador las aplica todas. En caso contrario (por ejemplo si el color de un elemento está definido en varios sitios con colores distintos), existen ciertas reglas que informan acerca de qué estilos son los que finalmente se aplican. Estas son:
 - Las propiedades definidas en un atributo `style` son prioritarias a las

- definidas en una etiqueta <style>.
- Las propiedades definidas en una etiqueta <style> son prioritarias a las definidas en una hoja de estilos externa.
 - Las propiedades definidas en un atributo <style> son prioritarias a las definidas en una hoja de estilos externa.



Actividades

3. En un sitio web se ha definido, con CSS, el color del texto de los enlaces. Esta propiedad ha sido definida en dos lugares distintos:

- En una hoja de estilos externa (color azul).
- En la cabecera (etiqueta <style>) de todos los documentos del sitio (color rojo).

¿De qué color se verán finalmente los enlaces? Razone su respuesta.

2.4. Formateado de páginas mediante estilos

En este apartado, se van a enumerar algunas propiedades CSS que pueden utilizarse para dar formato al texto de los documentos web:

- **font-family**: determina la fuente de letra a utilizar. Se puede especificar más de una, separadas por comas (”,”) por si el navegador no reconoce la indicada en primer lugar. Por ejemplo:

```
a { font-family: Times, serif; }
```

- **font-style**: establece el estilo de la fuente que se use. Puede ser: normal, italic u oblique. Ejemplo:

```
h3 { font-style: italic; }
```

- **font-weight**: determina el grosor o peso de la fuente. Sus posibles valores son: “normal”, “bold” (negrita), “bolder”, “lighter”, “100”, “200”, “300”, “400”, “500”, “600”, “700”, “800”, “900”. Por ejemplo:

```
nav { font-weight: normal; }
```

- **font-size**: define el tamaño de la fuente y admite distintos tipos de valores:
 - Tamaños absolutos: “xx-small”, “x-small”, “small”, “medium”, “large”, “x-large” y “xx-large”.
 - Tamaños relativos: “larger” y “smaller”.
 - Unidades de longitud o porcentajes.
 - Cualquier valor de longitud (absoluto y relativo).
 - Ejemplo:

```
td { font-size: 12em; }
```

- **color**: especifica el color de un elemento. El valor del color puede indicarse por su nombre en inglés o en formato RGB hexadecimal. Por ejemplo:

```
.clase { color: red; }
```

- **word-spacing**: define una cantidad adicional de espacio entre las palabras. Esta cantidad puede expresarse en cualquier unidad de longitud. Por ejemplo:

```
#espaciado { word-spacing: 0.4em; }
```

- **letter-spacing**: es idéntica a la anterior, salvo que el espacio se establece entre las letras. Por ejemplo:

```
H1.esp { letter-spacing: 0.2em; }
```

- **text-decoration**: permite que el texto sea decorado por uno de estos efectos:

- **underline**: subrayado.
- **overline**: subrayado superior.
- **line-through**: tachado.
- **blink**: parpadeo
- **none**: valor por defecto. No se produce ningún efecto.

- **vertical-align**: establece la alineación vertical del texto respecto al elemento contenedor o la línea del elemento. Admite porcentajes y las siguientes palabras clave:

- **baseline**: alinea el elemento con la base del elemento padre.
- **sub**: subíndice.
- **super**: superíndice.
- **top**: alinea la parte superior del elemento con el elemento más alto en la línea.
- **text-top**: alinea la parte inferior del elemento con el elemento más bajo en la línea.
- **middle**: el elemento se coloca en la mitad del elemento padre.
- **bottom**: alinea la parte inferior del elemento con el elemento más bajo en la línea.
- **text-bottom**: alinea las partes inferiores del elemento y la fuente del elemento padre.

Por ejemplo:

```
.text { vertical-align: 40%; }
```

- **text-transformation:** esta propiedad permite establecer, en el texto, diferentes transformaciones, como son:

- **capitalize:** escribe en mayúsculas el primer carácter de cada palabra.
- **uppercase:** escribe en mayúsculas todos los caracteres de cada palabra.
- **lowercase:** uso de letras pequeñas para los caracteres de cada palabra.
- **none:** no hace nada. Valor por defecto.
- Ejemplo:

```
H1 {text-transform: uppercase;}
```

- **text-align:** alinea el texto que alberga un contenedor, por lo que solo es aplicable a elementos contenedores o de bloque. Puede tener los siguientes valores: **left** (izquierda), **right** (derecha), **center** (centrado) y **justify** (justificado). Ejemplo:

```
P {text-align: justify;}
```

- **text-indent:** también se aplica a elementos contenedores y determina la sangría de la primera línea del texto. Ejemplo:

```
P {text-indent: 4px;}
```

- **line-height:** establece espacio entre las líneas de un texto. Si el valor es un número, la altura de línea se calcula a partir de la multiplicación del tamaño de

fuente por ese valor. Los valores en porcentaje son relativos al tamaño de fuente del elemento. No se permiten los valores negativos.



Aplicación práctica

Está desarrollando un sitio web y desea escribir, en un archivo CSS externo, un fragmento de código CSS que establezca negrita y subrayado a todos los enlaces del sitio que tengan asociada la clase “vinculo”. Escriba el fragmento de código CSS que debería escribir para aplicar dichos estilos.

SOLUCIÓN

```
a.vinculo {  
    font-weight:bold;  
    text-decoration:underline;  
}
```

2.5. Estructura de páginas mediante estilos. El modelo de cajas

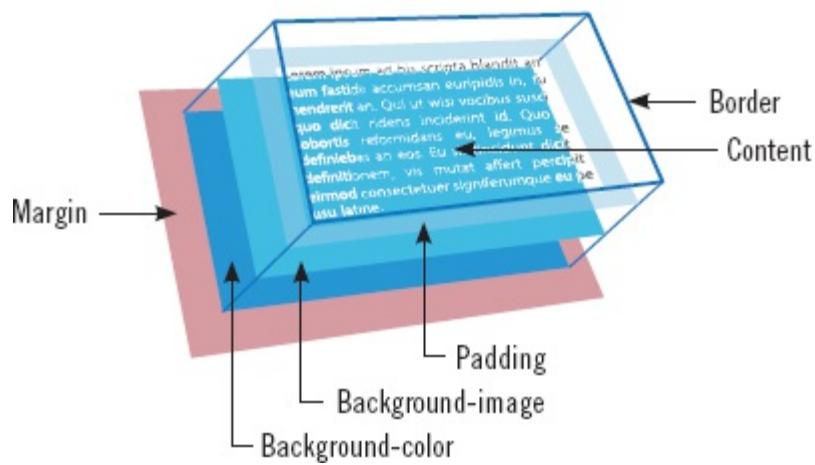
El modelo de cajas (en inglés *box model*) es una de las características más importantes de CSS, ya que condiciona el diseño de la inmensa mayoría de los sitios de la WWW. El *box model* consiste en el comportamiento CSS que hace que todos los elementos incluidos en un documento web presenten una estructura similar a una caja rectangular. Esto hace que sea muy fácil controlar el aspecto de cualquier elemento presente en un documento web, contribuyendo a definir la estructura visual de cualquier página web.

El modelo de cajas de CSS establece que todos los elementos web están constituidos por las siguientes partes:

- **content** (contenido): contenido del elemento. Por ejemplo: las palabras que contiene un párrafo, la imagen contenida dentro de una directiva , etc.
- **padding** (relleno): espacio que hay entre el contenido y un borde que lo cierre.

- **border** (borde): línea que encierra el contenido y el relleno.
- **background-image** (imagen de fondo): imagen que aparece detrás del contenido y el relleno.
- **background-color** (color de fondo): color que aparece detrás del contenido y el relleno.
- **margin** (margen): espacio que existe entre la caja y otros elementos adyacentes.

**Esquema representativo de la estructura de un elemento web,
según el modelo de cajas de CSS**



Nota

Existen muchas propiedades CSS que permiten modificar diversos parámetros relacionados con cada una de estas partes. Para conocerlas, se puede visitar la página <http://www.w3schools.com/css/>.

3. Diseño de estilos para diferentes dispositivos

A la hora de diseñar y desarrollar una página web, es importante tener en cuenta que, una vez que sea subida a la WWW, esta va a ser accesible desde multitud de dispositivos, tales como pc, móviles, tabletas, etc., por lo que su diseño debe orientarse hacia su correcta visualización en pantallas que van a tener tamaños muy diferentes. Para esto, es importante tener en cuenta las siguientes consideraciones (algunas de las

cuales ya se comentaron en capítulos anteriores), relacionadas con la implementación de código CSS que va a afectar a los elementos del sitio web:

- Utilizar tamaños relativos (por ejemplo porcentajes) en cualquier propiedad CSS en la que sea necesario especificar alguna medida. Esto favorecerá a que el tamaño de los elementos se adapte a las dimensiones de la ventana de visualización del dispositivo que corresponda.
- Utilizar las denominadas *Media Queries* que incorpora la nueva especificación CSS3.



Recuerde

La última versión de CSS (CSS3) incorpora las denominadas *Media Queries*, que consisten en una serie de condiciones y requisitos que permiten filtrar estilos CSS dependiendo de un rango de resolución de pantalla determinado.



Aplicación práctica

Acaba de desarrollar un sitio web y, una vez que los aloja en un servidor remoto, se dispone a comprobar su visualización desde su teléfono móvil. Al hacerlo, se da cuenta de que el contenido principal de la página se desborda por los laterales de la pantalla, siendo necesario desplazarse de un lado a otro. El contenedor principal de la página tiene asociadas las siguientes propiedades CSS:

```
#contenedor {  
    color:#000000;  
    text-align:center;  
    text-decoration:none;  
    width:1200px;  
    height: 100%;  
}
```

¿Cree que alguna de estas propiedades CSS puede afectar negativamente a la visualización de la web en su teléfono móvil? Razone su respuesta.

Nota: las propiedades CSS width y height se utilizan para establecer, respectivamente, el ancho y alto de los elementos web.

SOLUCIÓN

Como se puede apreciar, existe una propiedad (`width`) asociada al contenedor que establece un tamaño fijo en el ancho de este elemento. Esto significa que dicho contenedor (a no ser que existan otras propiedades CSS que la anulen) siempre se visualizará con un ancho de 1200 píxeles, un tamaño muy superior a la resolución de pantalla de muchos dispositivos móviles.

4. Tipos de hojas de estilos: estáticas y dinámicas. Las pseudo-clases

A lo largo de este capítulo, se ha estudiado la manera de aplicar estilos CSS de manera estática, es decir, el aspecto que presentan los elementos a los que se les aplica es siempre el mismo e independiente de cualquier otro factor externo.

No obstante, en CSS existen las denominadas pseudo-clases, las cuales aportan algo de dinamismo a los estilos que se pueden aplicar a los elementos web, ya que permiten aplicar estilos únicamente cuando se cumplan ciertas condiciones.

Las pseudo-clases se escriben justo después del selector en el que se quieren incluir, según la siguiente sintaxis:

```
...
    selector:pseudo-clase {
        propiedad1: valor1;
        propiedad2: valor2;
        propiedad3: valor3;
    ...
}
```

A continuación, se enumeran algunas de las pseudo-clases más utilizadas en CSS:

- `:first-child`: hace referencia al primer elemento de un tipo determinado que está contenido dentro de otro.
- `:link`: permite establecer estilos en los vínculos que todavía no han sido visitados.
- `:visited`: permite establecer estilos en los enlaces que sí han sido visitados.
- `:hover`: permite especificar el estilo de un elemento cuando el ratón se sitúa sobre él.
- `:active`: permite establecer el aspecto de un elemento cuando se hace clic sobre él.
- `:focus`: permite establecer el estilo de un elemento cuando este tiene el foco.
- `:lang`: permite especificar el aspecto de los elementos de un idioma determinado.



Actividades

4. Escriba el fragmento de código CSS que es necesario para establecer subrayado en los enlaces de una página web cuando, sobre estos, se posiciona el cursor del ratón.
-

5. Buenas prácticas en el uso de hojas de estilo

A continuación, se enumeran algunos consejos que es muy recomendable tener en cuenta cuando se aplican estilos CSS a los elementos de cualquier sitio web:

- **Separar el contenido del diseño:** como se ha visto anteriormente, el código CSS puede escribirse en un archivo externo, enlazado en la cabecera del documento. Esto permite separar completamente el contenido (HTML) de la presentación (CSS).
- **Aprovechar los prefijos de proveedor:** a veces, los navegadores ofrecen compatibilidad con estándares que todavía no se han aprobado. Para usarlos, existen unos prefijos que, añadidos a la propiedad que corresponda, permiten visualizarla de manera confiable en el navegador de que se trate.
- **Utilizar preprocesadores:** este tipo de aplicaciones permiten escribir código con mayor rapidez y posibilidades, adaptándolo posteriormente a un formato óptimo para todo tipo de navegadores. Algunos ejemplos de preprocesadores son: *Sass*, *Stylus* y *Less*.
- **Depurar el código CSS:** existen herramientas que permiten depurar las hojas de estilo que se utilicen en un sitio web, eliminando (entre otras cosas) aquellos fragmentos de código que no se utilicen.
- **Evitar repeticiones:** también es importante evitar redundancias en las hojas de estilo. Esto permitirá, entre otras cosas, hacer que la carga del documento sea más liviana.



Recuerde

El método más utilizado para definir los estilos CSS que van a afectar a los elementos de un sitio web consiste en implementarlos en un archivo “.css” externo.

6. Resumen

El aprendizaje de CSS es un paso fundamental para poder diseñar sitios web atractivos y competentes, ya que ofrece infinidad de posibilidades de diseño, muchas de las cuales son imposibles de lograr con la única utilización del lenguaje HTML.

Fundamentalmente, existen tres maneras diferentes de incluir estilos CSS en un

documento web. Estas son:

- Mediante el atributo `style`.
- Mediante la etiqueta `<style>`.
- Mediante un archivo “**.css**” externo.

Para añadir estilos CSS con los dos últimos métodos se utilizan los denominados selectores, los cuales consisten en indicadores que informan acerca del elemento o elementos que van a ser los beneficiarios de los estilos que establezcan. Existen cuatro tipos de selectores básicos:

- Selectores de etiquetas.
- Selectores descendentes.
- Selectores de clase.
- Selectores id.

También es importante conocer las denominadas pseudo-clases, que permiten dotar de cierto dinamismo a los estilos CSS que se pueden definir para los elementos de un sitio web.



Ejercicios de repaso y autoevaluación

1. ¿En qué consisten las hojas de estilo en cascada?

2. ¿Cómo se pueden insertar estilos CSS en un documento web?

- a. Mediante el atributo `style`.
- b. Mediante la etiqueta `<style>`.
- c. Mediante un archivo “`.css`” externo.
- d. Todas las respuestas anteriores son correctas.

3. Complete el siguiente texto.

Una manera de definir estilos CSS en un documento web consiste en incluirlos dentro de la etiqueta _____, la cual debe ubicarse dentro de la _____ del documento (_____).

4. ¿Qué son los selectores CSS?

5. Corrija los errores que detecte en el siguiente fragmento de código CSS.

```
p {  
    color: abcabc;  
    text-decoration:underline;  
}
```

6. Cuando se desean aplicar estilos CSS a ciertos elementos web que se encuentran ubicados dentro de otros, se utilizan los selectores...

- a. ... de etiquetas.
- b. ... descendentes.
- c. ... de clase.

d. ... de id.

7. Cuando se escribe un selector de clase, es necesario escribir, justo delante del selector, el carácter...

- a. ... “.”.
- b. ... “&”.
- c. ... “#”.

d. Ninguna de las respuestas anteriores es correcta.

8. Complete el siguiente texto.

Con CSS es posible definir estilos que se apliquen a elementos que tengan asociada una clase determinada y que, a su vez, pertenezcan a un tipo de _____ concreto (etiqueta). Para ello, basta con escribir dicha _____ justo delante del selector de _____ que corresponda.

9. ¿Para qué se utilizan los selectores id?

10. ¿Con qué prioridad se aplican los estilos CSS en el caso en el que existan conflictos entre ellos?

11. Para establecer negrita en los textos de los documentos web se puede utilizar la propiedad...

- a. ... font-family.
- b. ... font-style.
- c. ... font-weight.
- d. ... text-decoration.

12. Escriba la propiedad CSS (con su valor) que hay que usar para subrayar textos.

13. Complete el siguiente texto.

El modelo de cajas (en inglés *box model*) es una de las características más importantes de CSS, ya que condiciona el _____ de la inmensa mayoría de los sitios de la WWW. El *box model* consiste en el comportamiento CSS que hace que todos los _____ incluidos en un documento web presenten una estructura similar a una _____.

14. ¿Por qué se dice que las pseudo-clases aportan dinamismo al aspecto de los elementos de las páginas web?

15. ¿Qué pseudo-clase se utiliza para definir estilos CSS en elementos cuando el cursor del ratón se posiciona sobre ellos?

- a. :visited.
 - b. :hover.
 - c. :link.
 - d. :focus.
-

Bibliografía

Monografías

- ÁLVAREZ García, A.: *HTML 5*. Madrid: Anaya Multimedia. 2010.

Textos electrónicos, bases de datos y programas informáticos

- Albergue de Náufragos, lenguajes de marcas, de:
[<http://www.alberguedenafragos.com.ar>](http://www.alberguedenafragos.com.ar).
- Antonio Berciano Alonso: enlaces, de: [<http://platea.pntic.mec.es/~abercian/>](http://platea.pntic.mec.es/~abercian/).
- Codex Exempla: sintaxis de XML, de: [<http://www.codexexempla.org>](http://www.codexexempla.org).
- Curso Web 2.0 online: Páginas Web: definición y estructura, de:
[<http://cursoweb20.net/>](http://cursoweb20.net).
- David Valverde: “8 razones para no maquetar con tablas”, de:
[<http://www.davidvalverde.com/>](http://www.davidvalverde.com).
- Desarrollo Web: “¿Qué es el DOM?”, de: [<http://www.desarrolloweb.com/>](http://www.desarrolloweb.com).
- El viaje del navegante: “El documento XML”, de:
[<http://elviajedelnavegante.blogspot.com.es>](http://elviajedelnavegante.blogspot.com.es).
- El otro lado: colores HTML, de: [<http://www.elotrolado.net>](http://www.elotrolado.net).
- Haz una web: separadores, etiqueta hr, de: [<http://html.hazunaweb.com>](http://html.hazunaweb.com).
- Hieluki blog: “¿Qué es y para qué sirve H264?”, de: [<http://hieluki.com/blog/>](http://hieluki.com/blog).
- Html Quick: tablas en HTML, de: [<http://www.htmlquick.com/>](http://www.htmlquick.com).
- iNegocios: “¿Qué es una página web?”, de: [<http://www.wevxs.com/inegocios/>](http://www.wevxs.com/inegocios).
- I.E.S. Gaviota: “El lenguaje HTML”, de:
[<http://www.juntadeandalucia.es/averroes/~04000134/index.htm>](http://www.juntadeandalucia.es/averroes/~04000134/index.htm).
- Kioskea: formularios HTML, de: [<http://es.kioskea.net>](http://es.kioskea.net).

- | Libros Web: “Breve historia de HTML”, de: <<http://librosweb.es/>>.
- | La Webera: principios de diseño web, de: <<http://www.lawebera.es/>>.
- | Mc Libre: estilos en cascada, de: <<http://www.mclibre.org>>.
- | Premio Fundación Telefónica de Innovación Educativa: enlaces HTML, de: <<http://premioeducacion.fundaciontelefonica.com/>>.
- | Rafael Barzana Llana. Universidad de Murcia: “¿Qué son los lenguajes de marcado o de marcas?”, de: <<http://www.um.es/docencia/barzana/>>.
- | Universidad de Cádiz: estilos físicos y lógicos, de: <<http://www2.uca.es>>.
- | Universitat Pompeu Fabra: “Diseño web centrado en el usuario: usabilidad y arquitectura de la información”, de: <<http://www.upf.edu/>>.
- | Web Taller: “El proceso de diseño de un sitio Web”, de: <<http://www.webtaller.com>>.