

TEMA 2. LENGUAJES DE MARCADO GENERALES

2.0. Introducción

Los lenguajes de marcas, o lenguajes de marcado, consisten en una manera de codificar un documento de manera que, además del texto, se incluyen marcas o etiquetas que proporcionan información adicional sobre cómo es la estructura del texto o cómo debe presentarse el texto en pantalla.

El lenguaje de marcado más conocido es HTML, aunque no es el único.

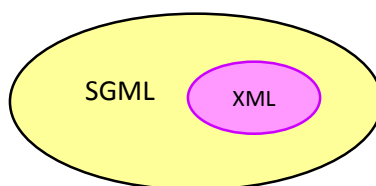
2.1. Origen de los lenguajes de marcado generales: SGML y XML

Los lenguajes de marcas son anteriores a los ordenadores. Desde la aparición de la imprenta, era habitual incluir anotaciones o instrucciones en los márgenes de los documentos que se enviaban a las imprentas con instrucciones tipográficas y de presentación para que el editor supiera, por ejemplo, que determinado texto tenía que ir en negrita o cursiva o que era una nota a pie de página. A medida que la industria editorial evoluciona, estas marcas se van estandarizando.

Esta idea de separar el contenido de la presentación fue lo que luego se trasladó al ámbito informático y se aplica de manera muy extendida en el mundo del diseño web.

Aunque existieron proyectos anteriores (como el Gencode), se considera que el investigador Charles F. Goldfarb es el padre de este tipo de lenguajes. Goldfarb participó en la década de los 60 en un proyecto auspiciado por IBM y en el que se desarrolló el lenguaje GML (Generalized Markup Language). Posteriormente, dirigió el comité que elaboró SGML (Standard Generalized Markup Language, estándar ISO 8879:1986 que especifica una sintaxis para las marcas que se incluyen en los textos además de establecer qué etiquetas se permiten y dónde se deben utilizar).

Posteriormente, W3C desarrolló un estándar para solucionar las limitaciones de SGML: XML (eXtensible Markup Language). XML tomó muchas de las ventajas de SGML y eliminó muchos de sus inconvenientes. Desde su publicación, se ha popularizado rápidamente siendo un lenguaje muy utilizado para el intercambio de información entre aplicaciones.



2.2. Características generales de los lenguajes de marcado

La característica fundamental y común a los lenguajes de marcas consiste en mezclar dentro del mismo documento:

- El texto, es decir, el contenido del documento que es la información que realmente se desea mostrar o suministrar al usuario.
- Las etiquetas. Elementos que se incluyen en el documento (delimitadas de alguna forma, generalmente encerrándolas entre signos <>) y que contienen instrucciones de cómo debe mostrarse (o interpretarse) el contenido que las sigue.

2.3. Estructura general de un documento con lenguaje de marcado

En los lenguajes de marcado es importante distinguir entre:

- **Estructura lógica.** Está compuesta por las partes que lo componen. Aunque la estructura puede variar según el lenguaje que estemos estudiando, en casi todos los lenguajes encontramos dos grandes secciones perfectamente diferenciadas:
 - **Cabecera** (encabezado, prólogo,...). En muchas ocasiones, esta sección es opcional y de incluirla puede contener datos interesantes del documento que no forman parte del contenido.
 - **Cuerpo.** En esta sección se incluye el contenido propiamente dicho del documento. Por ejemplo, el texto que debe mostrar el navegador o la información que se quiere enviar. Generalmente, esta sección es obligatoria.
- **Estructura física.** La estructura física determina la apariencia que presentará el documento, definiendo aspectos tales como la posición que ocupa cada elemento dentro del documento (cabecera, cuerpo,...), la alineación, la tipografía, etc.

2.3.1. Metadatos e instrucciones de proceso

Los metadatos son “los datos de los datos” (Por ejemplo, en un email serían metadatos la fecha y hora de envío, lugar de conexión, etc.).

En el contexto de un documento web, usamos los metadatos para incluir cualquier información relevante sobre la propia página. La función principal de estos metadatos es refinar las consultas a buscadores.

Los siguientes son ejemplos de metadatos incluidos en una página web:

```
<meta http-equiv="Content-Type" content="application/xhtml+xml; charset=utf-8" />
```

```
<meta name="description" content="Gijón te brinda la posibilidad de formarte en Asturias con cursos, masters gratuitos y de pago homologados y de calidad" />
```

2.3.2. Codificación de caracteres. Caracteres especiales (escape)

Una consideración importante a la hora de incluir nuestros textos en las páginas web es la inserción de caracteres especiales:

- Los caracteres <, > y " que utiliza HTML como parte de sus etiquetas no se pueden utilizar libremente.
- Los caracteres propios de idiomas que no sean el inglés (como la ñ del idioma castellano), pueden presentar ciertos problemas a la hora de visualizar la página donde estén incluidos.

La solución a este problema consiste en utilizar el escape de caracteres que no es otra cosa que sustituir estos caracteres reservados por ciertas expresiones (a las que se denomina entidades):

Caracteres especiales	
entidad	Carácter
<	<
>	>

Caracteres alfabeto español	
Entidad	Carácter
ñ	ñ
Ñ	Ñ

&amp;	&
&quot;	"
&nbsp;	(espacio en blanco)
&apos;	'

&aacute;	á
&eacute;	é
&iacute;	í
&oacute;	ó
&uacute;	ú
&Aacute;	Á
&Eacute;	É
&Iacute;	Í
&Oacute;	Ó
&Uacute;	Ú
&euro;	€

NOTA. En HTML y XML es muy común incluir sentencias charset al principio del documento, en la parte del encabezado, donde se indique el código del conjunto de caracteres que se va a emplear para poder escribir directamente el carácter en cuestión y no tener que recurrir continuamente a las entidades anteriores.

2.3.3. Etiquetas o marcas

Ya hemos dicho anteriormente que los lenguajes de marcado se denominan así precisamente por utilizar marcas o etiquetas. Estas etiquetas son fragmentos de texto escrito de una forma especial que permiten definir las diferentes instrucciones propias del lenguaje en cuestión.

Generalmente, estas etiquetas se escriben entre algunos signos especiales (por ejemplo, en HTML, las etiquetas van encerradas entre los signos < y >).

En el siguiente ejemplo podemos ver dos etiquetas HTML: una la etiqueta font y otra la etiqueta b. Además vemos que se utiliza una etiqueta para indicar el comienzo del texto afectado por esta etiqueta y otra etiqueta para indicar el final del texto afectado por dicha etiqueta.

```
<font color="red"><b>Este texto se imprimirá en rojo</b></font>
```

2.3.4. Atributos

En ocasiones, la etiqueta por sí sola no contiene suficiente información y es necesario añadir algún elemento que nos permita agregar información adicional para que la etiqueta quede completamente definida. Estos elementos es lo que llamamos **atributos**. Esto nos permite utilizar la misma etiqueta para diferentes propósitos.

Cada etiqueta tiene sus propios atributos, es decir, no todos los atributos se pueden utilizar en todas las etiquetas.

2.3.5. Elementos

En muchas ocasiones se utiliza el término “*Elemento*” como sinónimo de “*Etiqueta*”. Sin embargo, esto no es del todo correcto ya que un elemento es mucho más que una etiqueta. Un elemento suele estar formado por:

- Una etiqueta de apertura.
- Cero, uno o más atributos.
- Texto afectado por la etiqueta (opcional, puede haber etiquetas que no encierren ningún texto).

- Una etiqueta de cierre.

<code><etiqueta</code>	<code>atributo1="valor1"> atributo2="valor2"</code>	Texto	<code></etiqueta></code>
Etiqueta apertura	atributos	Texto afectado por la etiqueta	Etiqueta de cierre

En el ejemplo siguiente podemos ver un elemento en el que podemos apreciar todos sus componentes:

```
<a href="http://www.asata.com">Página asata </a>
```

```
<p>Este es el primer párrafo</p>
<p>Este es el segundo párrafo</p>
```

En este ejemplo tenemos dos elementos (los dos párrafos) y cuatro etiquetas (las dos que abren el párrafo, `<p>` y las dos que indican el final de párrafo `</p>`

Estos elementos a los que hacemos referencia se pueden clasificar en dos grupos:

- Elementos en línea (inline elements). Sólo ocupan el espacio de la línea necesario para mostrar sus contenidos.
- Elementos de bloque (block elements). Este tipo de elementos siempre empiezan en una línea nueva y ocupan la línea completa.

2.3.6. Comentarios

Habitualmente, los programadores suelen incluir texto dentro de su código que permiten incluir notas y explicaciones sobre cómo se ha creado el código o qué hace determinada instrucción. Estos textos son lo que se denominan comentarios.

Al igual que otros lenguajes de programación, los lenguajes de marcado permiten incluir comentarios que añaden información adicional al código y que no se mostrarán por pantalla, siendo invisibles para el usuario final.

Los comentarios se suelen introducir de una manera específica para que el navegador sepa qué parte de un documento es código y cuál otra son comentarios y debe de omitirlos. Por ejemplo, en HTML los comentarios tienen una sintaxis como la siguiente:

```
<!-- texto del comentario -->
```

```
<!-- Inicio del menú -->
```

Aunque el usuario no visualiza los comentarios en pantalla, si accede al código de la página web sí que los verá por lo que deberemos de tener sumo cuidado de no introducir datos confidenciales o sensibles dentro de los mismos.

2.4. Documentos válidos y bien formados. Esquemas

Los documentos bien formados aseguran que se cumplen las reglas de sintaxis del lenguaje que se está utilizando y que no existe ninguna incoherencia en el uso del lenguaje en cuestión.

Para conseguir un documento bien formado es importante cumplir los estándares del lenguaje que se está utilizando en cada momento.

Una de las innovaciones de SGML fue incorporar el concepto de DTD¹. El DTD nos permite definir la estructura de un documento y qué etiquetas, atributos y valores se pueden emplear en el mismo.

Se considera que un documento es válido si cumple las especificaciones que le imponga un DTD o un esquema (documento creado en un lenguaje en concreto y cuya función es la misma del DTD).

Existen validadores que comprueban si el documento está bien formado y que se ajusta a las especificaciones del lenguaje utilizado.

¹ Document Type Definition

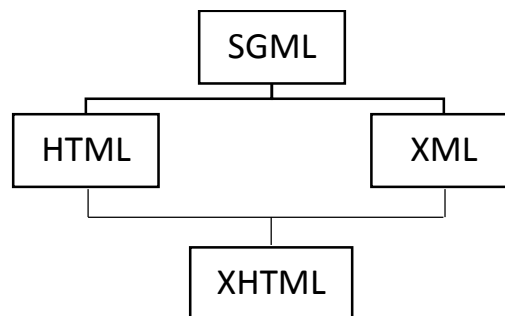
Contenido

TEMA 3. LENGUAJES DE MARCADO PARA PRESENTACIÓN DE PÁGINAS WEB	2
3.0. INTRODUCCIÓN	2
3.1. HISTORIA DE HTML Y XHTML. DIFERENCIAS ENTRE VERSIONES.....	2
3.2. ESTRUCTURA DE UN DOCUMENTO.....	3
3.2.1. INTRODUCCIÓN	3
3.2.2. ESTRUCTURA DE UN DOCUMENTO HTML	3
3.2.3. VERSIONES.....	4
3.2.4. LA ETIQUETA <HTML>	4
3.2.5. CABECERA.....	5
3.2.6. ELEMENTOS DEL CUERPO DEL DOCUMENTO	7
3.3. COLOR	10
3.3.1. CODIFICACIÓN DE COLORES.....	13
3.3.2. COLORES TIPO	14
3.3.3. COLORES SEGUROS	15
3.3.4. ARMONÍAS DE COLORES	16
3.4. TEXTO	17
3.4.1. ENCABEZADOS. JERARQUÍA Y ESTRUCTURA DEL CONTENIDO DE UN DOCUMENTO.....	18
3.4.2. PÁRRAFOS	19
3.4.3. ALINEACIÓN, ESPACIADO Y SANGRADO DE TEXTO.....	19
3.4.4. CARACTERÍSTICAS DE LETRA: TIPOS, TAMAÑOS Y COLORES	20
3.4.5. SEPARADORES DE TEXTO	21
3.4.6. ETIQUETAS ESPECÍFICAS PARA EL MARCADO DE TEXTO. ESTILOS LÓGICOS	21

TEMA 3. LENGUAJES DE MARCADO PARA PRESENTACIÓN DE PÁGINAS WEB

3.0. INTRODUCCIÓN

Una vez que hemos visto las generalidades sobre los lenguajes de marcas, vamos a centrarnos en dos de los lenguajes más utilizados para el diseño de páginas web: **HTML** (Hypertext Markup Language, Lenguaje de Marcado de Hipertexto) y **XHTML** (eXtensible HTML).



3.1. HISTORIA DE HTML Y XHTML. DIFERENCIAS ENTRE VERSIONES

Los orígenes de HTML se remontan a 1980, de la mano de Tim Berners-Lee¹, ingeniero del CERN², quien propuso un sistema de *hipertexto*³ para compartir documentos.

El primer documento formal publicado sobre la estructura del HTML se publicó en el año 1991 con el nombre de **HTML Tags**⁴ y que consistía en un documento de texto donde se explican las etiquetas que se usan en HTML.

En 1993 el IETF⁵ propone convertir HTML en un estándar. Sin embargo, no es hasta 1995 cuando este mismo organismo consigue publicar HTML 2.0 que es el primer estándar oficial.

A partir de 1996, W3C⁶ pasa a ser el organismo encargado de la publicación de los estándares HTML, entre los que se incluyen: HTML 3.2 (1997), HTML 4.0 (1998) o HTML 4.01 (1999).

A partir de HTML 4.01, W3C abandonó la estandarización de HTML y se centró en el estandarizado de XHTML. Por esta razón en el año 2004 algunas empresas como Apple, Opera, Google o Mozilla, se unen y crean el grupo WHATWG⁷ con el fin de buscar un nuevo estándar para HTML. El 22 de enero de 2008 este grupo publica el primer borrador de HTML 5.0. Ante la fuerza de las empresas que constituyen WHATWG y la publicación de los borradores de HTML 5.0, W3C retoma la estandarización de HTML publicando en octubre de 2014 el nuevo estándar HTML5.

Como hemos comentado anteriormente, una vez publicado HTML 4.01, W3C paralizó sus labores de estandarización de HTML y se centró en desarrollar el estándar XHTML publicando la versión XHTML 1.0 a principios del 2000 y su revisión en agosto del 2002. Actualmente ya se han publicado los borradores de XHTML 1.1 y de XHTML 2.0.

¹ Científico de Ciencias de la computación británico, considerado el padre de la web

² Conseil Européen pour la Recherche Nucléaire, Organización Europea para la Investigación Nuclear

³ Texto que contiene enlaces a otros documentos

⁴ Puedes ver este documento en el siguiente enlace: <https://www.w3.org/History/19921103-hypertext/hypertext/WWW/MarkUp/Tags.html>

⁵ Internet Engineering Task Force, Grupo de trabajo de ingeniería de internet

⁶ World Wide Web Consortium

⁷ Web Hypertext Application Technology Working Group

3.2. ESTRUCTURA DE UN DOCUMENTO

3.2.1. INTRODUCCIÓN

Lo primero que saber a la hora de crear un documento HTML es que no necesitamos ningún software específico, basta simplemente con un editor de texto plano (como por ejemplo el bloc de notas que se incluye con Windows). Aunque con el bloc de notas sería suficiente, hay algunas aplicaciones que nos facilitarán enormemente la edición de nuestro documento HTML y que son gratuitas como pueden ser NotePad++ o SublimeText. Y, por supuesto, también existe multitud de software con licencia que nos sirve para este propósito (como Dreamweaver).

Además, deberemos de tener en cuenta que los documentos que generemos con nuestro código HTML deberán de tener la extensión .htm o .html

A la hora de teclear nuestro código HTML deberemos tener en cuenta algunas reglas:

- Las etiquetas van encerradas entre los signos < y >.
- Generalmente, toda etiqueta abierta debe de ser cerrada. Para abrir una etiqueta teclearemos <nombre_etiqueta> y para cerrarla </nombre_etiqueta>
- Es indiferente que las etiquetas se tecleen en mayúsculas o minúsculas.
- Los comentarios que se quieran añadir en una página web para ayudar en la interpretación del código se hará mediante la siguiente sintaxis:

```
<!-- texto del comentario -->
```

3.2.2. ESTRUCTURA DE UN DOCUMENTO HTML

La estructura de un documento en HTML incorpora varias partes que a continuación nombramos brevemente y que iremos desgranando a lo largo del presente tema:

- **Versión.** Es la primera línea de cualquier documento HTML y en ella especificaremos la versión de HTML en la que está escrita nuestro documento.
- A continuación, todos los documentos deben comenzar con la etiqueta <HTML> que indica que comienza nuestro código HTML y al final del documento deberá haber una etiqueta </HTML> que indica que nuestro código HTML se ha terminado. Todo el resto de nuestro código deberá ir entre estas dos etiquetas.

Dentro de las etiquetas <HTML> y </HTML> se incluirán dos grandes secciones:

- **Head o Cabecera.** Esta sección contendrá información que no se visualiza en la página web pero que es interesante para el navegador. Dentro de esta sección se incluye el título de la página que se visualizará en la barra de título del navegador.

La cabecera aparecerá identificada por las etiquetas <HEAD> que marcará el inicio de la cabecera y </HEAD> que marcará el final de la cabecera.

- **Body o Cuerpo.** Es la parte fundamental de nuestra página. En esta sección irá incluido todo el contenido que queramos visualice el usuario además de todo el código HTML que determinará cómo será la presentación de ese contenido. El body estará limitado por las etiquetas <BODY> que indicará el inicio del body y </BODY> que indicará el final del body.

3.2.3. VERSIONES

La primera línea de cualquier documento HTML será una sentencia destinada a identificar la versión de HTML en la que está escrito nuestro documento. Esto es importante porque permitirá al navegador cargar todas las funciones de ese estándar.

En el caso de HTML5 deberá de ser:

```
<!DOCTYPE html>
```

IMPORTANTE. La línea anterior deberá ser la primera del documento sin espacios ni líneas que la precedan.

3.2.4. LA ETIQUETA <HTML>

Después de la línea donde se indica la versión, aparecerá una etiqueta <HTML>. Utilizaremos la etiqueta <HTML> para indicar que empieza nuestro código y la etiqueta </HTML> para indicar que se ha terminado el código y se ha llegado al final del documento.

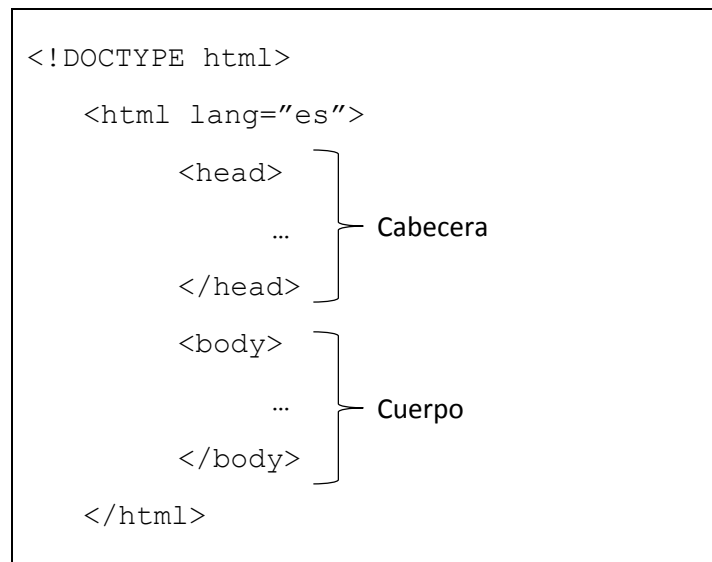
Podríamos prescindir de esta etiqueta pero se aconseja no hacerlo, para asegurarnos la compatibilidad con todos los navegadores.

Podemos usar la etiqueta <HTML> sin más pero existen una serie de atributos que podemos añadirle para definir algunas características generales de nuestro documento. Uno de los atributos que más habitualmente veremos junto a nuestra etiqueta <HTML> es aquél que nos permite el idioma humano del contenido de nuestro documento⁸.

```
<html lang="es">  
  
...  
  
</html>
```

Dentro de las etiquetas <html> </html> se distinguirán dos grandes secciones: cabecera y body.

⁸ Podemos consultar una lista de códigos de lenguajes en el siguiente enlace: http://www.w3schools.com/tags/ref_language_codes.asp



3.2.5. CABECERA

Como se dijo anteriormente, esta sección contiene información que, generalmente, el usuario no visualizará en la página (salvo el título que será visible en la barra de título del navegador), pero que servirá para proporcionar información adicional al navegador.

Alguna de la información que se suele incluir dentro de la cabecera será:

- El título de la página.
- El código de caracteres que se va a emplear.
- Incluir referencias a archivos externos como los que contienen los estilos CSS y/o código JavaScript.

A continuación vamos a describir algunas de las etiquetas más comunes en la cabecera.

3.2.5.1. ETIQUETA <META>

La primera etiqueta que nos solemos encontrar en la cabecera es la etiqueta <meta>. Esta etiqueta se utilizará para proporcionar información general sobre el documento. El usuario no verá la información contenida en esta etiqueta en la ventana de su navegador, pero la información que contiene es importante para los motores de búsqueda y algunos dispositivos.

La etiqueta <meta> puede estar acompañada por múltiples atributos. A continuación describimos brevemente los más habituales:

- **charset**. Nos permite definir la codificación que se va a utilizar en nuestra página (en nuestro caso casi siempre será utf-8). Sólo puede haber una etiqueta meta con el atributo charset dentro de la página.
- **name**. Este atributo nos permite definir un metadato. El atributo name nos permite definir el metadato que vamos a crear y el atributo **content** nos permite asignar valor a ese metadato.

Los metadatos más utilizados son:

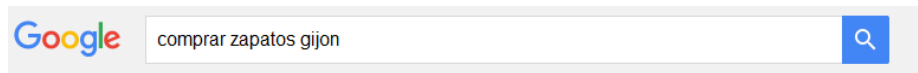
- **author**. Este metadato contendrá el nombre del autor de la página.
- **description**. Este metadato será un mensaje que nos permitirá dar una pequeña descripción de cuál es el contenido de nuestra página. Este valor será

tenido en cuenta por los motores de búsqueda. Cuando el usuario realice una búsqueda desde un buscador (como google) este texto aparecerá debajo del enlace de la página que lo contiene.

Por ejemplo, la página web de una zapatería contiene el siguiente código:

```
<meta name="description" content="Tienda de zapatos y calzado online. Compra zapatos online en Merkal para toda la familia. Descubre todo nuestro catálogo de zapatos online." />
```

Cuando un usuario teclea en un buscador como google *comprar zapatos Gijón*



El buscador muestra todas las páginas que ha encontrado que cumplen con los requisitos del usuario. En la línea en la que se muestra nuestra zapatería veremos lo siguiente:

Comprar Calzado en Gijón - Ofertas de Calzado barato

[www.tiendeo.com](#) > [Tiendeo en Gijón](#) > [Ofertas en Gijón](#) > [Moda](#) ▼

Encuentra las mejores ofertas de Calzado en Gijón y todas las promociones de tus tiendas favoritas. ¡Ahorra con Tiendeo!

Merkal Calzados - Comprar Zapatos online en Merkal

[www.merkal.com/](#) ▼

Tienda de zapatos y calzado online. Compra zapatos online en Merkal para toda la familia. Descubre todo nuestro catálogo de zapatos online.

Vemos que el texto que aparece debajo del enlace, es el que figura dentro del apartado description del código fuente de la página de Merkal.

- **keywords.** Esta etiqueta nos permitirá definir una serie de palabras clave que serán tenidas en cuenta por los buscadores.

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8" />
    <meta name="description" content="Ejercicio HTML5" />
    <meta name="keywords" content="HTML5, CSS3" />
</head>
<body>

</body>
</html>
```

Nota. Vemos que en el ejemplo anterior las líneas meta terminan con los signos `/>`. En HTML5 no es necesario cerrar etiquetas simples con la barra al final, pero se recomienda su uso por razones de compatibilidad.

3.2.5.2. ETIQUETA <TITLE>

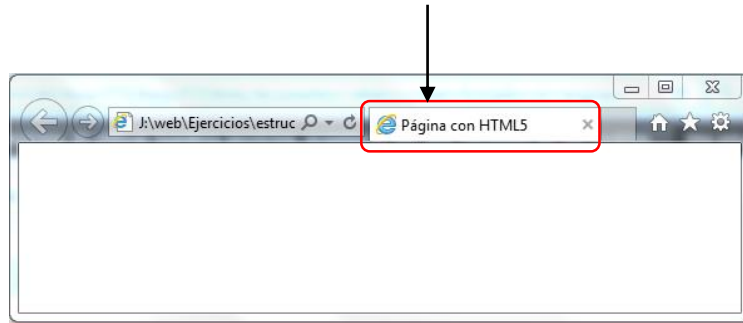
Esta etiqueta es la única de la sección HEAD que el usuario visualizará en su navegador.

La etiqueta <TITLE> tiene el siguiente formato:

```
<TITLE>Título de la página</TITLE>
```

El texto tecleado entre las etiquetas <TITLE> aparecerá en la parte superior de la ventana del navegador cuando el usuario esté visualizando la página:

```
<TITLE>Página con HTML5</TITLE>
```



3.2.5.3. ETIQUETA <LINK>

La etiqueta <link> permite cargar un recurso externo y relacionarlo con la página web actual. Uno de los usos más frecuentes de esta etiqueta es para enlazar la página actual con el archivo que contenga las hojas de estilo del documento.

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

Para declararla correctamente, en HTML5 sólo necesitaremos los atributos *rel* y *href*.

- **rel** nos permitirá indicar la relación entre el archivo en el que estamos y el archivo que incorporamos.
- **href**. Nos permite indicar qué archivos queremos vincular con nuestro documento y dónde se encuentra ubicado dicho archivo.

En el ejemplo de arriba, estamos vinculando un archivo del tipo hoja de estilos css que se llama estilos y que se encuentra ubicado en la misma carpeta que la página web en la que estamos poniendo el link (ya que no indicamos ninguna ruta, solo estamos poniendo el nombre del archivo).

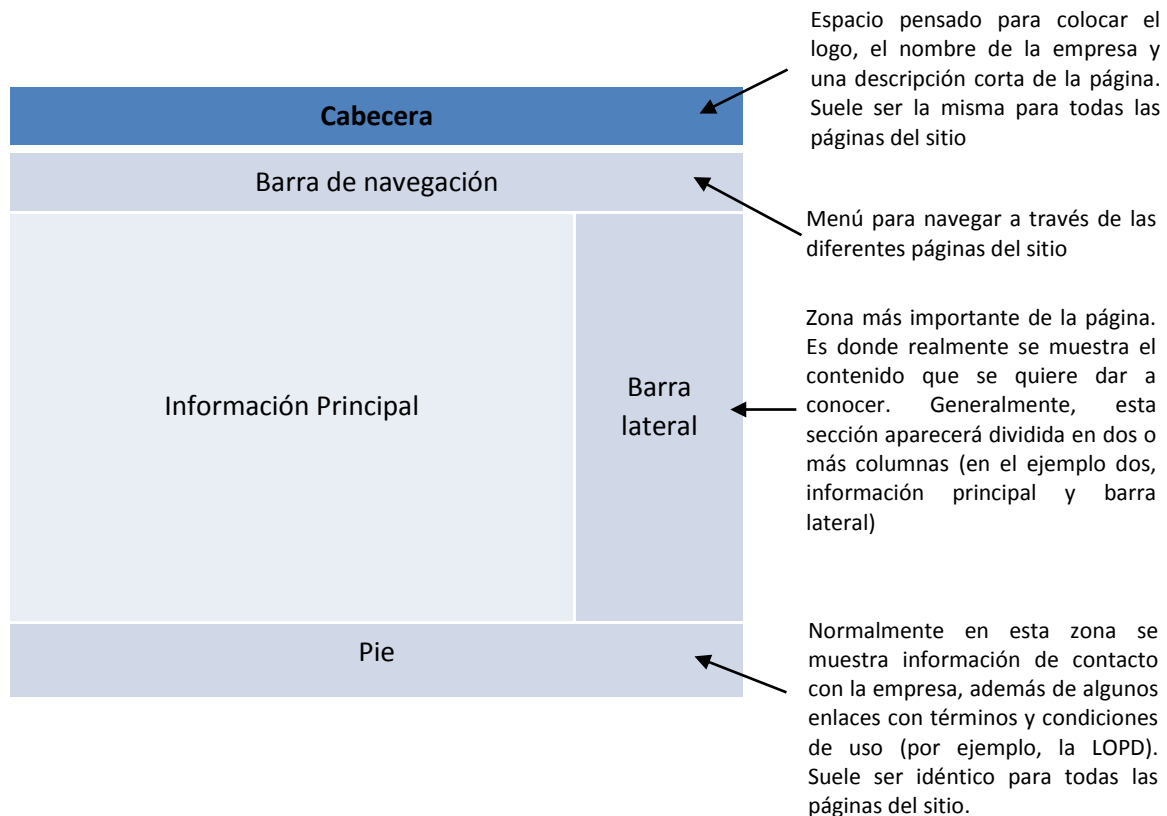
3.2.6. ELEMENTOS DEL CUERPO DEL DOCUMENTO

El cuerpo del documento (body) será la parte fundamental de nuestras páginas ya que en el incluiremos toda la información que queremos que el usuario vea en su navegador cuando abra nuestra página.

Para definir el cuerpo de nuestras páginas, utilizaremos las etiquetas <body> de apertura y </body> de cierre. Entre estas dos etiquetas incluiremos todo el contenido de nuestra página (textos, imágenes, tablas, enlaces, etc.)

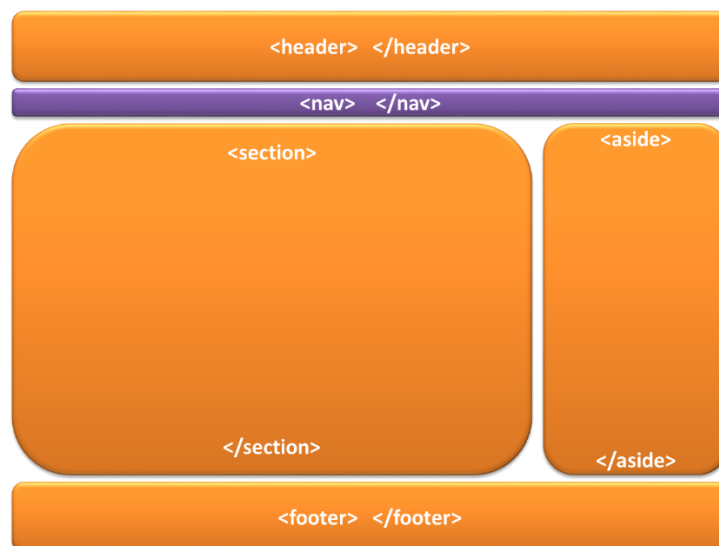
3.2.6.1. ESTRUCTURA BÁSICA DE UNA PÁGINA WEB

Hoy en día cualquier página web suele presentar una estructura similar y parecida a la siguiente:



3.2.6.2. ESTRUCTURA MÁS ACEPTADA POR HTML5

HTML5 acepta la estructura anterior y proporciona elementos para diferenciar y declarar cada una de las partes especificadas en el punto anterior.



- **HEADER.** Es la cabecera de nuestra página web. Generalmente contiene el logo, el nombre, el título y subtítulos de la página y, en ocasiones, alguna breve descripción del sitio.

Lo normal es que esta sección sea la misma para todas páginas de un mismo sitio.

Nota. No debemos confundir HEADER con HEAD.

- **NAV.** Este elemento contiene el menú de navegación para permitir al usuario moverse a través de las diferentes páginas que componen el sitio web.
- **SECTION.** Este elemento contiene la información relevante de la página, generalmente la información que queremos suministrar al usuario que está visitando nuestro sitio.

En una misma página web puede haber varias SECTION.

- **ASIDE.** Este elemento permite crear las barras laterales que mostrábamos en el apartado 3.2.6.1. Generalmente, esta parte de la página contendrá publicidad, enlaces a redes sociales (como Facebook, twitter), en ocasiones también puede contener enlaces a otras páginas de interés externas a la propia empresa, etc.

En una página web puede haber varios ASIDE y pueden estar tanto a izquierda como derecha de la página.

- **FOOTER.** Es el pie de página por lo que aparecerá en la parte inferior de la página. Generalmente suele aparecer el autor, la empresa, términos y condiciones, etc.

Lo normal es que esta sección sea la misma para todas páginas de un mismo sitio.

3.2.6.3. EL ELEMENTO <ARTICLE>

En ocasiones la información que colocamos en un <section> no conforma un bloque de contenido sino que se trata de información diferente. Para dividir el contenido de una section y agrupar la información por características similares utilizaremos el elemento <article>.

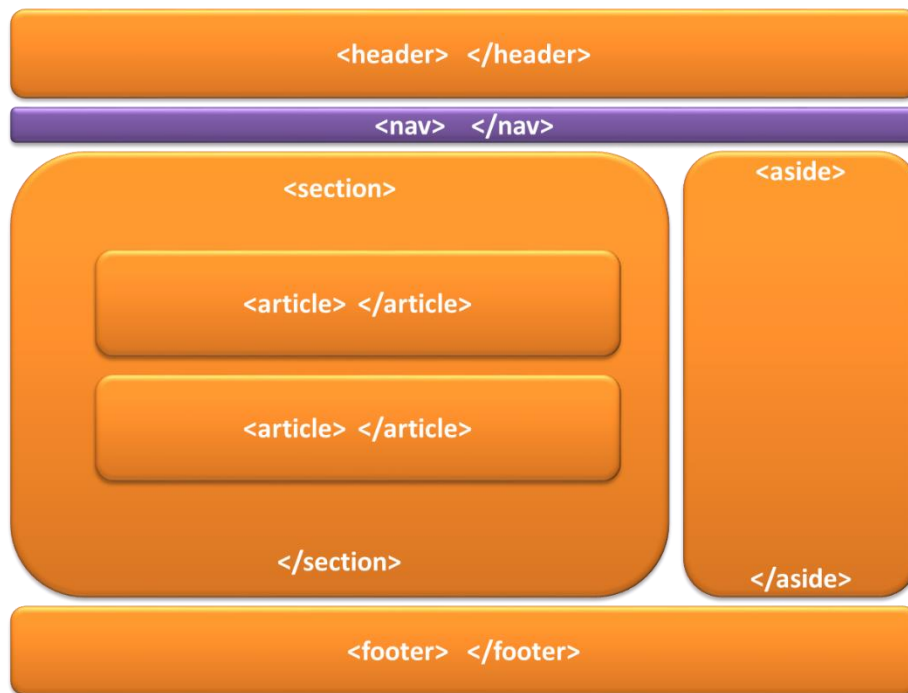
En una página web pueden existir varios articles dentro de un section (incluso independientes del mismo).

Dentro del <article> podemos proporcionar otro <header> y otro <footer>.

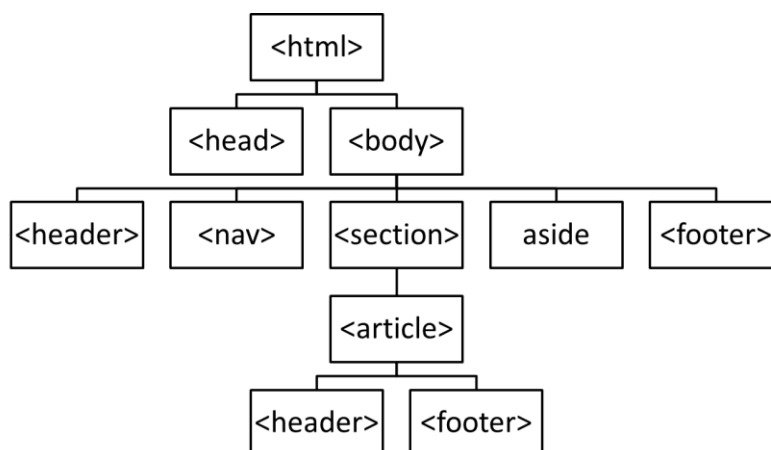
Las diferencias entre <section>, <article> y <aside> son algo complejas y es fácil confundir una con otra de principio. Como "*truco*" para distinguir cuando debemos usar cada una podemos aplicar el siguiente criterio:

- Si la información que queremos incluir no tuviera sentido por sí sola sino que adquiere el sentido al incluirlo con el resto de lo comentado en la página, usaremos un <section>.
- Si la información que queremos incluir tiene sentido por sí sola y al colocarla en una página diferente tuviera significado sin necesidad del resto, podemos usar un <article>.
- Si la información a incluir es algo que no tiene mucho que ver con el resto de información mostrada en la página y lo único que pretende es ayudar, complementar o proporcionar recursos adicionales, entonces deberíamos ponerla en un <aside>.

La nueva estructura del <body> incorporando el elemento <article> quedaría como sigue:



Si la estructura anterior, la representáramos en forma jerarquizada, obtendríamos el siguiente árbol:



3.3. COLOR

El color resulta **vital** en el diseño web.

Existen estudios que demuestran que la gente no sólo compra un producto por el producto en sí, sino que influyen los colores que lo acompañan (<http://www.lavanguardia.com/estilos-de-vida/20120113/54244233858/los-colores-nos-condicionan-al-comprar.html>, estudios sobre compra de detergentes y cremas faciales). Es más, cuando una persona opta por un producto, el 90% de esa decisión se basa en el color.



Además el color aumenta el reconocimiento de una marca hasta en un 80%. El reconocimiento de marca está directamente relacionado con la confianza del consumidor hacia esa marca.

80% INCREMENTO
RECONOCIMIENTO DE MARCA

20%

Existe un término denominado la “*psicología del color*” que analiza cómo afecta cada color a las personas:

Rojo	Sentimientos positivos	Emoción, pasión, amor, energía
	Sentimientos negativos	Violencia, guerra, peligro o fuego
	Puedes usarlo	Páginas de comida, deportes, servicios de emergencia o salud
	Debes evitarlo	Páginas de artículos de lujo, naturaleza, páginas profesionales No conviene abusar de éste color, puede llegar a ser cargante
Amarillo	Sentimientos positivos	Felicidad, alegría, optimismo, juventud, competitividad
	Sentimientos negativos	Cobardía, engaño, envidia, vulnerabilidad
	Puedes usarlo	Puede dar buenos resultados en botones de llamada a la acción
	Debes evitarlo	No conviene abusar de este color ya que puede resultar agresivo
Naranja	Sentimientos positivos	Energía, felicidad, diversión, emoción, entusiasmo, calor, ambición y cautela
	Sentimientos negativos	Debilidad, cobardía, celos, desconfianza, enfermedad, miedo, odio
	Puedes usarlo	Páginas de tiendas, automoción, tecnología, entretenimiento, comida y cuidado de niños. Atraer la atención sobre nuestras llamadas a la acción (como botones para suscripción, comprar, etc.)
	Debes evitarlo	No conviene abusar de éste color, puede llegar a ser cargante
Verde	Sentimientos positivos	Salud, bienestar, crecimiento, fertilidad, energía, masculinidad, naturaleza, calma, paz, buena suerte, dinero y generosidad
	Sentimientos negativos	Envidia, enfermedad, cobardía, enfado, rencor
	Puedes usarlo	Páginas de ciencia, turismo, medicina, recursos humanos, medioambiente y sostenibilidad
	Debes evitarlo	Páginas de artículos de lujo, tecnología o contenido dedicado a adolescentes.
Azul	Sentimientos positivos	Verdad, fuerza, productividad, seguridad, fidelidad, confianza, lealtad, seriedad, eficacia y calidad
	Sentimientos negativos	Frialdad, distanciamiento, tristeza y apatía
	Puedes usarlo	Páginas sobre temas de salud, tecnología, medicina, ciencia, políticos y servicios públicos. Muchos bancos y grandes empresas lo usan porque es signo de seriedad.

	Debes evitarlo	Puede reducir el apetito por lo que deberíamos evitarlo en aquellas páginas que tengan que ver con la comida
Morado	Sentimientos positivos	Realeza, autoridad, poder, riqueza, creatividad, imaginación, sofisticación, prosperidad, misterio, sabiduría y respeto
	Sentimientos negativos	Obsesión, crueldad, injusticia e intolerancia
	Puedes usarlo	Dar sensación de lujo y riqueza Páginas de productos de belleza, astrología, masaje, yoga, salud, espiritualidad y contenido relacionado con chicas adolescentes y marcas femeninas
	Debes evitarlo	Tiene un efecto relajante por lo que no es muy buena idea utilizarlo en páginas donde se quiera dar sensación de dinamismo
Rosa	Sentimientos positivos	Sofisticación, sinceridad, ternura, sensibilidad, sentimentalidad
	Sentimientos negativos	Nadie asocia sentimientos negativos al color rosa, si acaso, cursilería
	Puedes usarlo	Para productos o páginas dirigido a mujeres y chicas adolescentes
	Debes evitarlo	El fucsia puede resultar demasiado chillón y algunos rosas suaves pueden resultar demasiado dulces para páginas donde se quiera dar sensación de vitalidad.
Marrón	Sentimientos positivos	Tierra, naturaleza, solidez, estabilidad, equilibrio, madurez, estabilidad y amistad
	Sentimientos negativos	Aspereza, decadencia, penitencia extrema
	Puedes usarlo	Estimula el apetito, lo que lo hace ideal para páginas con contenido alimenticio. Páginas con contenido animal, veterinario, finanzas Es muy utilizado para fondos.
	Debes evitarlo	No se recomienda cuando lo que queremos es llamar la atención. No debería usarse nunca en llamadas a la acción (como por ejemplo, botones).
Negro	Sentimientos positivos	Sofisticación, elegancia, autoridad, poder, estabilidad, fuerza, formalidad e inteligencia
	Sentimientos negativos	Muerte, misterio, maldad y rebelión
	Puedes usarlo	Para páginas de productos de lujo, moda, marketing y cosméticos
	Debes evitarlo	Puede ser cargante y hacer que la gente se sienta incómoda o asustada por lo que deberemos tener mucho cuidado con su uso.
Blanco	Sentimientos positivos	Pureza, limpieza, felicidad, sinceridad y seguridad
	Sentimientos negativos	Debilidad, vulnerabilidad, cobardía, falta de vitalidad.
	Puedes usarlo	Páginas relacionadas con la industria de la salud, páginas de tecnología y ciencias. Combinado con negro, oro, plata o gris produce muy buenos resultados para artículos de lujo.

	Debes evitarlo	Dependiendo de los colores que lo acompañan, podemos utilizarlo en casi todo tipo de páginas.
Gris	Sentimientos positivos	Formalidad, profesionalidad, sofisticación, neutralidad, equilibrio, atemporalidad.
	Sentimientos negativos	Represión, limitación, miedo, derrota, fracaso.
	Puedes usarlo	Páginas profesionales o de artículos de lujo.
	Debes evitarlo	Cuando se quiera llamar la atención de la gente.

De la tabla anterior, deducimos que antes de elegir los colores que vamos a utilizar en nuestra página web debemos tener en cuenta varios aspectos:

- Qué ofrecemos. Debemos usar colores relacionados con el producto o servicio que queramos ofrecer.
- A quién está dirigido. Dependiendo del sector al que dirijamos nuestra página (jóvenes o adultos, hombres o mujeres, etc.), habrá una serie de colores más apropiados que otros.

3.3.1. CODIFICACIÓN DE COLORES

Existen diferentes modelos de color, es decir diferentes formas de calcular el color: HSV (Hue-Saturation-Value, Tono-Saturación-Valor), CMYK (Cyan-Magenta-Yellow-black, Cyan-Magenta-Amarillo-Negro), RGB (Red-Green-Blue, Rojo-Verde-Azul), Pantone, etc.

El modelo más adecuado para representar imágenes en monitores de computadora es el RGB.

El modelo RGB (Rojo, Verde, Azul) se basa en mezclar estos tres colores para conseguir el círculo cromático completo.

HTML cuenta con diferentes formas de especificar los colores:

- **Nombre del color.** Cuando optemos por esta opción, deberemos utilizar el nombre del color en inglés, por ejemplo, white, red, black, etc. HTML cuenta con 147 colores predefinidos con su nombre.
- **Colores en formato RGB** (Red, Green, Blue). Cada uno de estos tres colores toma un valor entre 0 y 255, donde 0 significa ausencia total de ese color y 255 representa la máxima intensidad de ese color. Con este sistema, podemos representar $256 \times 256 \times 256 = 16.777.216$ colores diferentes.

rgb(0,0,0) = negro	rgb(255,0,0) = rojo	rgb(0,255,0) = verde
rgb(0,0,255) = azul	rgb(255,255,0) = amarillo	rgb(128,0,128) = morado
rgb(128,128,128) = gris	rgb(255,0,255) = rosa	rgb(255,255,255) = blanco

- **Colores en formato hexadecimal.** Es la forma más utilizada para especificar los colores en HTML. Al igual que en el formato anterior, para indicar un color se especificará la cantidad de Rojo, Verde y Azul necesario para formar ese color solo que el valor se expresará en formato hexadecimal, utilizando el valor 00 para representar ausencia total de ese color y FF para representar la máxima intensidad de ese color.

La sintaxis para especificar el color será:

#RRGGBB

#000000 = negro	#FF0000 = rojo	#00FF00 = verde	#0000FF = azul
#FFFF00 = amarillo	#800080 = morado	#808080 = gris	#FFFFFF = blanco

Nota. Cuando un color lleva letras en su sintaxis no importa si éstas se escriben en mayúsculas o minúsculas. Sin embargo, lo que sí es muy importante es colocar siempre el carácter de almohadilla #

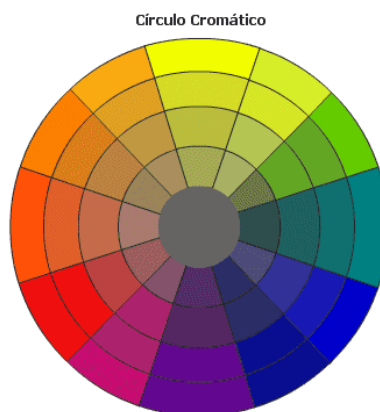
3.3.2. COLORES TIPO

Como hemos comentado en el apartado anterior, el modelo más utilizado para el diseño web es el sistema RGB. Este sistema parte de tres **colores primarios**: Rojo, Verde y Azul.

A partir de los colores primarios podemos obtener el resto de colores del círculo cromático y dependiendo de las tonalidades hacer diferentes clasificaciones.

Mezclando directamente dos colores primarios en iguales cantidades obtenemos los **colores secundarios**: cian, magenta y amarillo. Entre estos tres, existe una gran gama de colores que se obtienen de variar la cantidad de cada color primario para componerlos.

A base de mezclar colores vecinos entre sí obtendremos el círculo cromático que nos ayudará a clasificar todos los colores y a determinar qué combinaciones son adecuadas y cuáles no.



A partir del círculo cromático podemos clasificar los colores de diferentes formas, entre las que a continuación citaremos algunas.

3.3.2.1. Colores cálidos y fríos

Los colores cálidos provocan un efecto estimulante, dan sensación de dinamismo, vitalidad y alegría. Sin embargo, los colores fríos producen un efecto relajante dando sensación de frialdad o distanciamiento.

Colores cálidos



Amarillo - Naranja – Rojo – Púrpura (en menor medida)

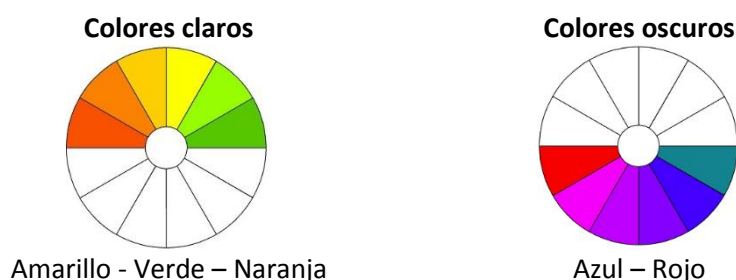
Colores fríos



Verde – Azul - Violeta

3.3.2.2. Colores claros y oscuros

Los colores claros inspiran limpieza, juventud, liviandad, descanso. Por el contrario los colores oscuros inspiran seriedad, madurez, calma. Encierran el espacio y lo hacen parecer



3.3.2.3. Colores pastel y colores apagados o sucios

Los colores pastel se obtienen disminuyendo la luminosidad en todo el ciclo cromático mientras que los colores apagados se obtienen aumentando la luminosidad en todo el círculo cromático.

Los colores pastel sugieren frescura, naturalidad, luz, vida mientras que los colores apagados expresan seriedad, oscuridad, muerte.



3.3.2.4. Escala de grises

El blanco y el negro no se consideran colores y no aparecen en el círculo cromático. El negro sería la ausencia de total de color y el blanco sería la presencia de todos los colores.

Sin embargo, si combinamos blanco y negro se obtiene el gris y dependiendo de la cantidad de blanco y negro que utilicemos, obtendremos diferentes escalas de gris. Por esto que tenemos un círculo especial que llamaremos "círculo cromático en escala grises"

Los grises son colores en los que la cantidad de rojo, verde y azul es la misma. Es decir, si expresamos un color gris en formato hexadecimal tendría un formato como el siguiente #RRGGBB (donde RR = GG = BB).

Aunque suelen ser colores algo tristes se utilizan para expresar seriedad, elegancia, atemporalidad,...

3.3.3. COLORES SEGUROS

La expresión colores seguros para la web (web-safe colors), viene desde los orígenes de Internet. En aquella época los monitores a color sólo podían reproducir 256 colores diferentes.

Como hemos comentado anteriormente, con el sistema de representación de colores utilizado hoy en día podemos representar hasta 16.777.216 colores diferentes.

Si contamos con un monitor antiguo que sólo disponga de los 256 colores, y queremos visualizar una página web que utilice un código de color que no esté disponible, el navegador intentará reproducirlo mezclando píxeles de colores. Esto generará un efecto de tramado (dithering) que no resulta muy estético.

Para evitar estos tramados se creó la "paleta de colores seguros para la web". Esta paleta es una lista de 216 colores inmunes al tramado lo que hace que se muestren siempre igual, por lo que, al utilizar colores seguros, nos aseguramos que los visitantes ven los colores de la página tal y como se ha diseñado independientemente del monitor, del Sistema Operativo o del navegador que esté utilizando.

A la hora de definir un color seguro con el formato hexadecimal, los tonos empleados para cada color son 00, 33, 66, 99, CC y FF.

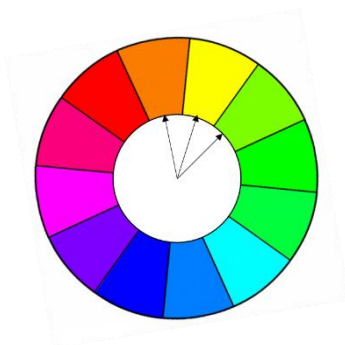
Actualmente, el empleo de los colores seguros ha decaído, dado que la existencia de monitores con 256 colores se ha decrementado en gran medida.

Existen multitud de páginas donde se puede consultar la paleta de colores seguros, como por ejemplo http://html.net/tutorials/html/lesson7_216websafecolourchart.php.

3.3.4. ARMONÍAS DE COLORES

A la hora de combinar colores para nuestra página, existen diferentes formas de hacerlo para que los colores elegidos resulten armónicos.

3.3.4.1. Armonía monocromática o de análogos

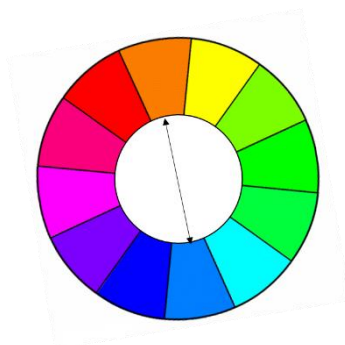


Se trata de elegir un color y sus análogos (es decir, diferentes tonos de ese color). Por ejemplo, Amarillo / Naranja / Verde

Para ello, elegimos un color del círculo cromático y seleccionaríamos los colores que están a izquierda y derecha del color seleccionado.

Este tipo de armonía se puede utilizar en temas asociados a la naturaleza.

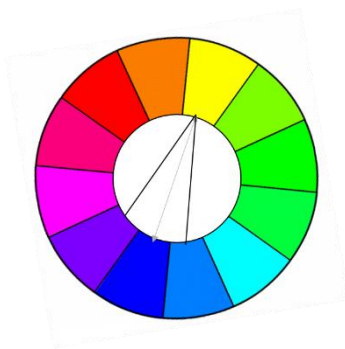
3.3.4.2. Armonía de colores complementarios



Dos colores son complementarios si se encuentran en puntos opuestos del círculo cromático. Por ejemplo, Azul / Naranja.

Un color parece más intenso cuando se asocia a su complementario, por lo que este tipo de contrastes resulta idóneo cuando se quiere llamar la atención.

3.3.4.3. Armonía de complementarios divididos

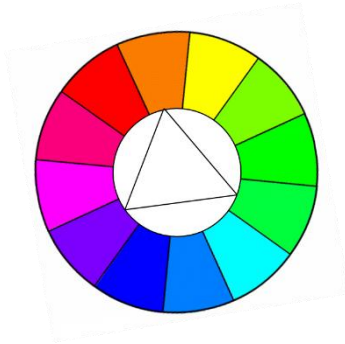


En este caso seleccionamos un color y en lugar de utilizar su complementario, utilizaremos los situados a izquierda y/o

derecha del complementario. Por ejemplo, Amarillo / Violeta rojizo / Violeta azulado.

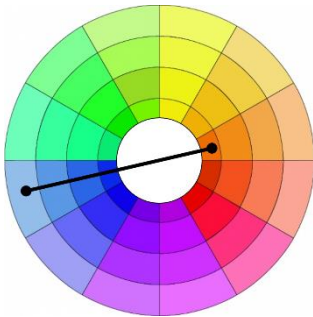
Esto hace que el contraste no sea tan marcado.

3.3.4.4. Armonía de tríadas o armonía de tres colores



Consiste en seleccionar tres colores que se encuentren equidistantes en el círculo cromático. Por ejemplo, Naranja / Violeta / Verde.

3.3.4.5. Mezcla brillante-tenue



Consiste en elegir un color brillante puro y una variante tenue de su complementario.

NOTA. Existen sitios web y aplicaciones que dado un color, nos dicen sus armónicos, sus complementarios, etc. Por ejemplo, Kuler de Adobe

3.4. TEXTO

El contenido más importante de una página web son los textos que la integran. Esto hace que la redacción, la disposición y la presentación de los mismos sean extremadamente importantes.

- A la hora de redactar un texto para una página web deberemos ser breves, claros y concisos.
- Deberemos ser muy cuidadosos con las faltas de ortografía y utilizar de manera correcta las mayúsculas y las minúsculas.
- Utilizar títulos y subtítulos que nos permitan esquematizar el contenido de la página.
- Deberemos procurar que cada párrafo solo contenga una idea.
- Es conveniente que aquellas palabras clave de cada párrafo estén destacadas de alguna manera (generalmente mediante el uso de negritas o itálica, se desaconseja utilizar el subrayado).
- Si usamos un solo color para todo el texto, éste resultará monótono. Se aconseja utilizar un color diferente para los títulos y subtítulos aunque sin pasar de dos o tres colores.
- También es importante que el color del texto contraste sobre el fondo sobre el que se encuentra.
- No utilizar efectos como texto parpadeante, marquesita, etc.

- Usar un máximo de tres fuentes diferentes.
- Usar un espaciado apropiado entre líneas, párrafos, títulos y subtítulos, etc.

Para introducir el texto en una página web, bastará con escribir el texto dentro de las etiquetas <body>.

3.4.1. ENCABEZADOS. JERARQUÍA Y ESTRUCTURA DEL CONTENIDO DE UN DOCUMENTO

Generalmente, el texto que incluimos dentro del <body> estará dividido en títulos, subtítulos y párrafos.

3.4.1.1. Etiquetas <h>

Para escribir estos títulos y subtítulos contamos con las etiquetas <h>. Tenemos 6 etiquetas <h>, desde la <h1> hasta la <h6>. El número indica la importancia del título, siendo <h1> el título de mayor importancia y <h6> el título menos importante.

```
<h1>Este texto será el título principal</h1>
<h2>Este texto será un subtítulo</h2>
```

Cada etiqueta <h> proporciona formatos de fuente diferentes: un tipo de fuente, un tamaño de letra, negritas, etc. Este formato dependerá del navegador con el que se esté visualizando la página web, lo que hace el navegador es formatear el párrafo según la etiqueta <h> que se encuentre para que el usuario pueda identificar que se trata de un titular y de la importancia de ese titular.

Además, el uso de etiquetas <h> implica la separación del texto en párrafos de manera que, lo que se escribe dentro de cada par de etiquetas <h> </h> será un párrafo independiente del resto.

En html5 los motores de búsqueda usan estas etiquetas <h> para optimizar los resultados de su búsqueda por lo que es sumamente importante utilizarlas de una manera correcta y respetar su jerarquía, es decir, primero declarar la etiqueta <h1> para los títulos, luego <h2> para los subtítulos y así sucesivamente.

Las etiquetas <h> son para colocar titulares en páginas web, no se deben utilizar para formatear texto normal. Si queremos que un texto normal aparezca más grande o en negritas o con un tipo de fuente diferente, utilizaremos los métodos adecuados y que veremos más adelante pero no las etiquetas <h>.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="Ejercicio titulos" />
    <meta name="keywords" content="HTML5, titulos y subtítulos" />
    <title>Página con Encabezados</title>
  </head>
  <body>
    <h1>Encabezado de nivel 1</h1>
    <h2>Encabezado de nivel 2</h2>
    <h3>Encabezado de nivel 3</h3>
    <h4>Encabezado de nivel 4</h4>
    <h5>Encabezado de nivel 5</h5>
    <h6>Encabezado de nivel 6</h6>
    Aquí irá el texto normal.
  </body>
</html>
```

3.4.2. PÁRRAFOS

Un párrafo es un fragmento de texto que se separa del siguiente por un punto y aparte.

HTML no reconoce la tecla de retorno de carro como marca de final de párrafo.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="Ejercicio párrafos" />
    <meta name="keywords" content="HTML5, párrafos" />
    <title>Párrafos</title>
  </head>
  <body>
    <h1>Ejercicio con párrafos</h1>
    <h2>Primera parte</h2>
    Este es el primer párrafo.
    Este es el segundo párrafo.
    Este es el tercer párrafo.
  </body>
</html>
```

Para definir un párrafo deberemos utilizar la etiqueta `<p>` donde queramos marcar el inicio de párrafo y `</p>` donde queramos indicar el final de párrafo.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8" />
    <meta name="description" content="Ejercicio párrafos" />
    <meta name="keywords" content="HTML5, párrafos" />
    <title>Párrafos</title>
  </head>
  <body>
    <h1>Ejercicio con párrafos</h1>
    <h2>Primera parte</h2>
    <p>Este es el primer párrafo. </p>
    <p>Este es el segundo párrafo. </p>
    <p>Este es el tercer párrafo. </p>
  </body>
</html>
```

NOTA. La etiqueta `</p>` no sería estrictamente obligatoria pero se considera mejor práctica emplearla, además de que el no utilizarla puede hacer que algún navegador provoque errores.

3.4.3. ALINEACIÓN, ESPACIADO Y SANGRADO DE TEXTO

Los párrafos admiten algunos formatos típicos de este tipo de estructuras como son la alineación, el espaciado entre líneas o las sangrías.

3.4.3.1. Alineación

La alineación nos permitirá definir si el texto debe aparecer a la izquierda, derecha, centro o justificado en la ventana del navegador. Por defecto, el texto de un párrafo aparecerá alineado a la izquierda.

Para definir la alineación, utilizaremos el parámetro `align="alineación"` donde alineación podrá tomar los valores `left`, `right`, `center` o `justify` según que queramos alinear a la izquierda, derecha, centrar o justificar respectivamente.

El parámetro align puede aparecer acompañando a multitud de etiquetas, como por ejemplo, <p>, <h1>, , etc.

```
<h1 align="center">Este título irá centrado</h1>
<p align="right">Este párrafo irá alineado a la derecha</p>
```

3.4.3.2. Espaciado

HTML considera espacios en blanco los propios espacios en blanco, los tabuladores y el Intro que utilizamos para indicar que queremos una nueva línea. Esto hace que cuando utilizamos uno de los caracteres anteriores, HTML lo sustituye por un espacio en blanco en vez de provocar el efecto habitual que tendría ese carácter. Además, si utilizamos varios espacios en blanco seguidos, el navegador mostrará un único espacio en blanco prescindiendo de los demás que se hayan incluido.

Por esta razón, HTML nos proporciona varios recursos que podemos utilizar cuando queremos separar dos palabras por más de un espacio o cuando queramos incluir una nueva línea en el párrafo.

Nuevas líneas

Para definir un salto de línea, podemos utilizar la etiqueta
. Utilizaremos
 si deseamos crear una nueva línea sin iniciar un nuevo párrafo.

La etiqueta
 es una etiqueta vacía, es decir, no delimita ningún texto. Por lo que lo correcto sería abrirla y cerrarla de forma seguida:
</br>. En este caso, HTML permite utilizar un atajo que consiste en utilizar la etiqueta
 (o
)

Espacios en blanco

Cuando lo que queremos es incluir más de un espacio en blanco entre dos palabras, sustituiremos cada espacio en blanco adicional por el código (Ojo, es obligatorio incluir el signo & del inicio y el ; del final).

3.4.3.3. Sangrado de texto

La sangría es una especie de margen que se fija a ambos lados del texto.

Para sangrar texto utilizaremos la etiqueta <blockquote> para marcar a partir de qué momento queremos que se inicie el sangrado del texto y </blockquote> para indicar dónde queremos que termine el sangrado.

El uso de la etiqueta <blockquote> hace que el texto que se teclea a continuación aparezca en un nuevo párrafo.

```
Este texto estará en los márgenes habituales del texto
<blockquote>Este párrafo aparecerá sangrado</blockquote>
Este párrafo volverá a aparecer en los márgenes normales del documento
```

3.4.4. CARACTERÍSTICAS DE LETRA: TIPOS, TAMAÑOS Y COLORES

En una página web se incluyen multitud de componentes de los cuales quizás el más utilizado es el texto. En un apartado anterior de este mismo tema dábamos algunos puntos que deberíamos de tener en cuenta a la hora de redactar los textos para nuestras páginas web. En este apartado no ocuparemos de cómo proveer a nuestros textos de un buen aspecto para que resulten atractivos.

Para definir propiedades de la letra como son el tipo, el tamaño o el color podemos utilizar la etiqueta .

La sintaxis de la etiqueta font será la siguiente:

```
<font face="nombre_fuente" size="tamaño" color="color">
```

- nombre_fuente será el nombre de la fuente que queramos utilizar: Arial, Impact, Tahoma, etc.
- size. Será el tamaño que queramos asignar a nuestro texto. HTML proporcionar 7 tamaños del 1 al 7.
- color. Especificará el color que se quiere aplicar al texto que aparece a continuación. Se puede especificar el color con cualquiera de los métodos de codificación que hemos visto en el apartado 3.3.1 (código rgb, código hexadecimal o nombre del color)

Al igual que en otros casos, es necesaria la etiqueta .

NOTA. Esta etiqueta se considera **Desaprobada**, por lo que, en lo posible, debe evitarse su uso.

3.4.5. SEPARADORES DE TEXTO

3.4.5.1. Etiqueta <hr>

La etiqueta <hr> no tiene nada que ver con las etiquetas <h> del apartado anterior. <hr> crea una línea horizontal en una página HTML.

El elemento <hr> se emplea para separar un párrafo de otro mediante el uso de una línea horizontal.

No es necesaria etiqueta </hr> para marcar el final de la línea.

Si solo especificamos la etiqueta <hr> sin ningún atributo, el navegador mostrará una línea horizontal que ocupará todo el ancho de la pantalla y a la que acompañará una sombra.

La sintaxis de la etiqueta <hr> será:

```
<hr align=left|right|center noshade size=n width=n>
```

- Align. Indica la alineación de la regla en el caso de que no ocupara todo el ancho de la pantalla del navegador.
- Noshade. Indica al navegador que no muestre la sombra de la línea.
- Size=n Indicará la altura de la línea expresada en píxeles.
- Width= n o n% Indicará el ancho de la línea. Este ancho se puede indicar en píxeles o en el porcentaje de pantalla que queremos que ocupe

Nota. No conviene abusar de estas líneas horizontales ya que pueden dificultar la lectura del contenido de la página.

3.4.6. ETIQUETAS ESPECÍFICAS PARA EL MARCADO DE TEXTO. ESTILOS LÓGICOS

Llamamos estilos a los diferentes efectos que aplicamos al texto para cambiar su aspecto.

En HTML existen dos tipos de estilos:

- **Estilos físicos.** Los estilos físicos son aquellas etiquetas que definen cómo se debe mostrar en pantalla el texto que encierran.
- **Estilos lógicos.** Los estilos lógicos están pensados para indicar al navegador cual es el significado del texto que delimitan. Por ejemplo, cuando encerramos un texto entre las etiquetas estamos indicando al navegador que ese texto tiene una gran importancia. Según el estilo lógico que se utilice, el texto se mostrará con un efecto.

	Etiqueta HTML	Ejemplo	Efecto
Estilos físicos		El texto se visualizará en negrita	
	<i>	El texto se visualizará en cursiva	
	<u>	El texto se visualizará subrayado	
Estilos lógicos	<address>	Especifica direcciones de correo electrónico	
	<blockquote>	Introduce citas textuales o texto destacado	
	<cite>	Indica el título de una película o el título de un libro	
		Indica texto con énfasis	
		Indica texto resaltado	

```
<h1>Estilos físicos</h1>
<b>Este texto estará en Negrita</b> <br />
<i>Este texto estará en Cursiva</i> <br />
<u>Este texto estará Subrayado</u> <br />

<h1>Estilos lógicos</h1>
<address>Este texto es una dirección de correo </address><br />
<blockquote>Este texto es una cita</blockquote><br />
<cite>Esta etiqueta contendrá el título de una película</cite><br />
<em>Esta etiqueta enfatiza el texto </em><br />
<strong>Esta etiqueta indica texto importante</strong><br />
```



Visualización en el navegador

Estilos físicos

Este texto estará en Negrita

Este texto estará en Cursiva

Este texto estará Subrayado

Estilos lógicos

Este texto es una dirección de correo

Este texto es una cita

Esta etiqueta contendrá el título de una película

Esta etiqueta enfatiza el texto

Esta etiqueta indica texto importante

Hay estilos físicos y estilos lógicos que producen el mismo efecto. Además, un mismo texto puede verse afectado por varios estilos físicos y/o lógicos a la vez.

3.7. LISTAS

En ocasiones necesitaremos mostrar varios elementos en forma de lista. HTML proporciona varias posibilidades para representar una lista.

HTML permite definir tres tipos de listas:

- Listas no ordenadas
- Listas ordenadas
- Listas de definición

3.7.1. CARACTERÍSTICAS

Las **listas no ordenadas** presentan una secuencia de elementos sin importar el orden en el que se aparecen. Cada elemento que forma parte de estas listas aparecerá identificado por algún tipo de viñeta sencilla.

Las **listas ordenadas** también presentan una secuencia de elementos pero en este caso los elementos mostrados seguirán un orden determinado. Cada elemento que forma parte de la lista aparecerá numerado.

Las **listas de definición** muestran un conjunto de términos y su definición (parecido a un diccionario). Son quizás las menos utilizadas en las páginas HTML.

Ejemplos								
Listas no ordenadas	Listas ordenadas	Listas de definición						
Los cinco continentes son: <ul style="list-style-type: none">• Europa• Asia• África• América• Oceanía	Helado de fruta: <ol style="list-style-type: none">1. Pelar y triturar la fruta2. En un bol montar la nata con el azúcar3. Añadir la fruta y batir4. Meter al congelador	<table><tr><td>París</td><td>Capital de Francia</td></tr><tr><td>Madrid</td><td>Capital de España</td></tr><tr><td>Roma</td><td>Capital de Italia</td></tr></table>	París	Capital de Francia	Madrid	Capital de España	Roma	Capital de Italia
París	Capital de Francia							
Madrid	Capital de España							
Roma	Capital de Italia							

3.7.2. ORDENACIÓN DE LISTAS

3.7.2.1. Listas no ordenadas

Como comentamos anteriormente, las listas no ordenadas son aquellas en las que los elementos que se enumeran no tienen por qué seguir un orden específico.

Una lista no ordenada irá encerrada entre las etiquetas `` de apertura de lista y la etiqueta `` de cierre de lista. Cada elemento de la lista irá encerrado entre las etiquetas `` y ``. (Cada etiqueta `` provoca un salto de línea por lo que no es necesario identificar cada elemento de la lista como un párrafo con la etiqueta `<p>`)

Código	Visualización
<pre> <p>Los cinco continentes son:</p> Europa Asia Africa América Oceanía </pre>	

Por defecto, el carácter de viñeta que mostrará el navegador al principio de cada elemento será un pequeño círculo negro. Pero este carácter se puede cambiar, mediante el atributo `style="list-style-type: tipo"` dentro de la etiqueta ``.

Valor tipo	Descripción
disc	Los elementos de la lista se marcarán con pequeños círculos negros (esta es la opción por defecto)
circle	Los elementos de la lista se marcarán con círculos
square	Los elementos de la lista se marcarán con cuadrados
none	No se mostrará ninguna marca delante de los elementos de la lista

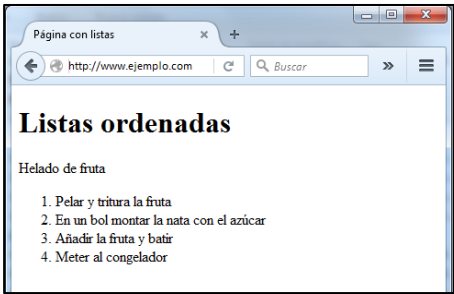
Ejemplo	Visualización
<pre> <ul style="list-style-type:disc"> Europa Asia Africa América Oceanía </pre>	<ul style="list-style-type: none"> Europa Asia África América Oceanía
<pre> <ul style="list-style-type:circle"> Europa Asia Africa América Oceanía </pre>	<ul style="list-style-type: none"> Europa Asia África América Oceanía
<pre> <ul style="list-style-type:square"> Europa Asia Africa América Oceanía </pre>	<ul style="list-style-type: none"> Europa Asia África América Oceanía
<pre> <ul style="list-style-type:none"> Europa Asia Africa América Oceanía </pre>	<ul style="list-style-type: none"> Europa Asia África América Oceanía

	
-------	--

3.7.2.2. Listas ordenadas

Las listas ordenadas presentan una serie de elementos pero en este caso la enumeración implica un orden.

Una lista ordenada irá encerrada entre las etiquetas de apertura de lista y la etiqueta de cierre de lista. Cada elemento de la lista irá encerrado entre las etiquetas y .

Código	Visualización
<pre><p>Helado de fruta</p> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	

Por defecto, el tipo de numeración con el que los elementos serán numerados será 1., 2., 3., etc. aunque esta numeración se podrá cambiar con el atributo `type` dentro de la etiqueta .

Valor type	Descripción
Type="1"	Los elementos de la lista serán numerados con números (por defecto)
type="A"	Los elementos de la lista serán numerados con letras mayúsculas
type="a"	Los elementos de la lista serán numerados con letras minúsculas
type="I"	Los elementos de la lista serán numerados con números romanos en mayúsculas
type="i"	Los elementos de la lista serán numerados con números romanos en minúsculas

Ejemplo	Visualización
<pre><ol type="1"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	<ol style="list-style-type: none"> 1. Pelar y tritura la fruta 2. Montar la nata con el azúcar 3. Añadir la fruta y batir 4. Meter al congelador
<pre><ol type="A"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	<ol style="list-style-type: none"> A. Pelar y tritura la fruta B. Montar la nata con el azúcar C. Añadir la fruta y batir D. Meter al congelador

Ejemplo	Visualización
<pre><ol type="a"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	<ol style="list-style-type: none"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador
<pre><ol type="I"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	<ol style="list-style-type: none"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador
<pre><ol type="i"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	<ol style="list-style-type: none"> Pelar y tritura la fruta Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador

Mediante el atributo `start` podemos especificar un valor para que la lista empiece a partir de él.

```
<ol start="25">
```


En una lista numerada también podemos especificar que la numeración se haga en sentido descendente. Para ello utilizaremos el atributo `reversed` (este atributo es nuevo de HTML5).

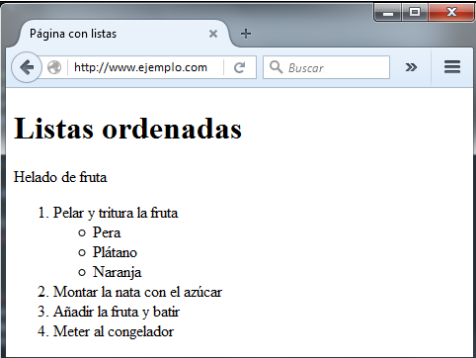
```
<ol reversed>
```

3.7.3. ANIDAMIENTO EN LISTAS

Una lista anidada no es más que una lista que está dentro de otra lista.

Podemos anidar una lista dentro de otra del mismo o diferente tipo.

Listas anidadas del mismo tipo	
<pre><ul style="list-style-type:circle"> Europa Asia Africa América <ul style="list-style-type:square"> América del Norte América del Sur Oceanía </pre>	
Listas anidadas de diferente tipo	

<pre> Pelar y tritura la fruta Pera Plátano Naranja Montar la nata con el azúcar Añadir la fruta y batir Meter al congelador </pre>	
--	--

Una forma bastante efectiva de crear listas anidadas es partiendo desde fuera hacia dentro. Es decir, comenzar creando la lista de primer nivel (la más externa), a continuación crearemos la de segundo nivel y así sucesivamente.

3.7.4. OTROS TIPOS DE LISTAS: LISTAS DE DEFINICIÓN

Una lista de definición es una lista de términos junto con su definición.

Para crear una lista de definición utilizaremos la etiqueta `<dl>` de apertura para marcar el comienzo de la lista y la etiqueta `</dl>` de cierre para indicar el final de la lista.

Dentro de la lista, nos encontraremos pares de elementos, uno será el término y otro la definición. Los términos los delimitaremos mediante etiquetas `<dt>` de apertura y cierre `</dt>` y las definiciones las delimitaremos mediante etiquetas `<dd>` de apertura y cierre `</dd>`.

Código	Visualización
<pre> <dl> <dt>París</dt> <dd>Capital de Francia</dd> <dt>Roma</dt> <dd>Capital de Italia</dd> <dt>Madrid</dt> <dd>Capital de España</dd> </dl> </pre>	

3.5. ENLACES DE HIPERTEXTO

Uno de los elementos más característicos de una página web es el hipertexto (hiper enlace, hipervínculo, enlace, etc.) que no es otra cosa que la herramienta que hace que podamos establecer un enlace entre nuestra página y otro recurso (otra parte del documento, otro documento, otra página web, etc.).

Los enlaces generalmente se delimitan con la etiqueta ancla: `<a>...`. La sintaxis básica de esta etiqueta es:

```
<a href="url">Texto_del_enlace</a>
```

- `url`. Es la ubicación del recurso al que se quiere dar acceso a través del enlace. Lo comentaremos en el siguiente apartado en detalle.
- `Texto_del_enlace`. Será el texto que el usuario visualizará en el navegador y que al picar sobre el mismo nos desplazará donde indique el atributo href. Este texto del enlace, puede ser

- un texto

```
<a href="http://www.google.com" >Web de google </a>
```

- una imagen.

```
<a href="http://www.google.com" ></a>
```

3.5.1. ESTRUCTURA DE UN ENLACE: LA DIRECCIÓN DE INTERNET O URL

3.5.1.1. Enlaces a otras páginas web

URL (Uniform Resource Locator, Localizador Uniforme de Recursos) nos permite determinar la ubicación de cualquier recurso para que podamos acceder desde el navegador.

El formato estándar de una URL será:

```
servicio://máquina.dominio:puerto/camino/fichero
```

donde,

- `servicio`. Indica el protocolo que se utilizará para acceder a la información contenida en ese enlace. Puede ser http, https, ftp, etc.
- `maquina.dominio`. Se refiere al servidor que contiene alojado el recurso al que queremos acceder. Aunque todos los servidores se identifican por una IP es más normal que nos refiramos a él con el nombre de la máquina y el dominio. Por ejemplo, `www.atcgijon.com`
- `puerto`. El puerto es opcional y no se suele utilizar salvo que se utilice un puerto diferente al estándar del servicio.
- `camino`. Será la ruta de directorios que es necesario recorrer para llegar al fichero al que se quiere acceder. Si para llegar al fichero necesitamos pasar por varios directorios, separaremos cada directorio del siguiente con la barra vertical / Al especificar el nombre de los subdirectorios deberemos tener cuidado entre mayúsculas y minúsculas ya que los servidores suelen diferenciar entre unas y otras.
- `fichero`. Nombre del fichero que contiene la página o recurso que queremos abrir. Deberemos especificar el nombre del fichero y su extensión (`catalogo.html`, `logo.jpg`,

etc). Al igual que con el atributo camino, deberemos distinguir entre mayúsculas y minúsculas.

NOTA. Si sólo especificamos el camino y no especificamos fichero, accederemos al fichero html por defecto (que generalmente se llamará default.html o index.html).

3.5.2. ESTILOS DE ENLACES

3.5.2.1. Estilos por defecto

En general, la información encerrada entre las etiquetas `<a>` `` tiene un formato asociado:

- En el caso de texto, el texto aparecerá subrayado. Además si el enlace no ha sido visitado aún, el color del texto será azul y si ya se ha visitado en alguna ocasión el texto cambiará de color (generalmente a un morado).
- En el caso de que la información que se inserta entre las etiquetas ancla, sea una imagen, ésta suele aparecer con un borde alrededor. Si el enlace asociado a la imagen no ha sido visitado nunca, el borde será azul y si ya se ha visitado el borde será de otro color (generalmente morado).

De todas formas, estos estilos pueden cambiar según el navegador y versión con la que se esté visualizando la página. Lo que sí suele ser común en todos los casos, es que al situar el puntero del ratón sobre el objeto que contiene el enlace, éste tomará la forma de una mano.

3.5.2.2. Definición de estilos

El estilo con el que se muestren los vínculos en el navegador puede ser modificado. Aunque lo aconsejable es utilizar css, se puede hacer utilizando directamente etiquetas HTML especificándolo en las propiedades de la página, en la etiqueta `<body>`, mediante los atributos `link` (vínculo) y `vlink` (vínculo visitado).

```
<body link="red" vlink="grey">
```

3.5.3. DIFERENCIAS ENTRE ENLACES ABSOLUTOS Y RELATIVOS

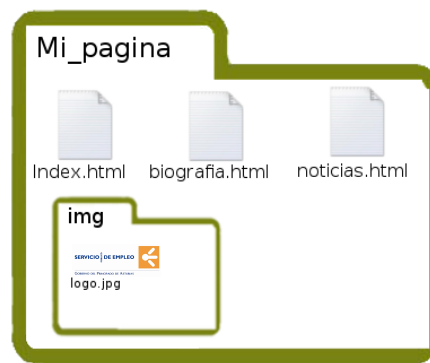
Existen diferentes formas de clasificar los enlaces que se incluyen en una página web. Una de estas clasificaciones es la que distingue entre:

- **Enlaces absolutos.** Son aquellos que incluyen todas las partes de la url (protocolo, servidor y ruta).

```
http://www.cursosepepa.com/archivos/formativa/access_avanzado.pdf
```

- **Enlaces relativos.** Estos tipos de enlaces prescinden de alguna de las partes que componen una url: generalmente el protocolo, maquina.dominio y, en ocasiones, la ruta. Habitualmente, este tipo de enlaces se emplean para redireccionarnos hacia archivos que se encuentran en el mismo servidor que el documento en el que se está incluyendo el enlace.

Ejemplo 1. Referencia a recurso que se encuentra en el mismo directorio que el origen



Supongamos que en la imagen anterior tenemos la estructura de carpetas de un dominio llamado ejemplo.com y que desde el archivo index.html quiero hacer una llamada al archivo noticias.html:

Página origen	<code>http://www.ejemplo.com/mi_pagina/index.html</code>
Página enlazada	noticias.html que se encuentra en la misma carpeta que index.html
url absoluta	<code>http://www.ejemplo.com/mi_pagina/noticias.html</code>
url relativa	<code>noticias.html</code>

Cuando el navegador encuentra la url relativa (noticias.html), debe determinar la url absoluta a partir de la url relativa para poder cargar el recurso y para ello:

1. A la url le falta el protocolo y el servidor por lo que supone que es el mismo que el de la página original (index.html)

`http:// + www.ejemplo.com`

2. Se añade la parte que falta a la url relativa para obtener la url absoluta del recurso al que se está referenciando (noticias.html).

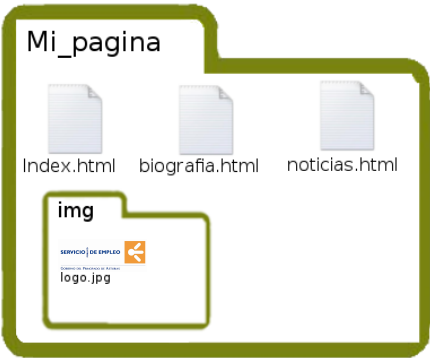
`mi_pagina/noticias.html`

`http:// + www.ejemplo.com + mi_pagina/noticias.html`

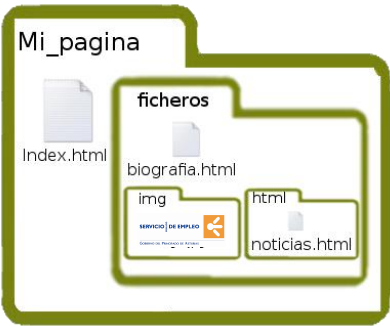
3. Se ha obtenido la url absoluta del recurso referenciado:

`http://www.ejemplo.com/mi_pagina/noticias.html`

Ejemplo 2. El recurso al que se hace referencia se encuentra en un directorio inferior al del origen

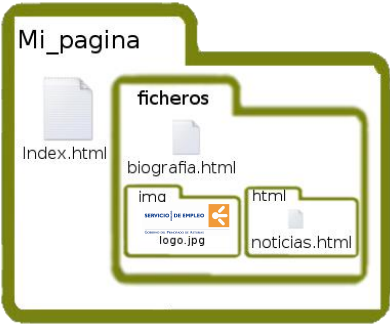


Página origen	http://www.ejemplo.com/mi_pagina/index.html
Imagen enlazada	logo.jpg que se encuentra en la carpeta img que está dentro de mi_pagina
url absoluta	http://www.ejemplo.com/mi_pagina/img/logo.jpg
url relativa	img/logo.jpg



Página origen	http://www.ejemplo.com/mi_pagina/index.html
Página enlazada	noticias.html que se encuentra dentro de la carpeta html que a su vez está dentro de la carpeta ficheros y ésta se encuentra dentro de la carpeta mi_pagina donde está la página de origen
url absoluta	http://www.ejemplo.com/mi_pagina/ficheros/html/noticias.html
url relativa	ficheros/html/noticias.html

Ejemplo 3. El recurso al que se hace referencia se encuentra en un directorio superior al del origen



Página origen	http://www.ejemplo.com/mi_pagina/ficheros/biografia.html
Página enlazada	index.html que se encuentra dentro de la carpeta mi_pagina que es el

	directorio superior a ficheros
url absoluta	http://www.ejemplo.com/mi_pagina/index.html
url relativa	../index.html

Página origen	http://www.ejemplo.com/mi_pagina/ficheros/html/noticias.html
Página enlazada	index.html que se encuentra dentro de la carpeta mi_pagina que es el directorio superior a ficheros y ficheros es el directorio superior a html
url absoluta	http://www.ejemplo.com/mi_pagina/index.html
url relativa	../../index.html

En ocasiones, si tenemos que subir muchos niveles suele ser más interesante recurrir a la url absoluta del recurso.

3.5.4. ENLACES INTERNOS

Los enlaces internos son aquellos que apuntan a un lugar diferente dentro de la misma página en la que nos encontramos.

Este tipo de enlaces nos permiten ofrecer al visitante la posibilidad de desplazarse rápidamente a aquella sección que le interese dentro de nuestra página.

Para la creación de estos enlaces deberemos seguir dos pasos:

- 1 Definir las secciones del documento. Para ello utilizaremos el parámetro name:

```
<A name="identificador_seccion"> texto sección </A>
```

- identificador_seccion. Este será el nombre que asignaremos a cada sección de las que queremos crear en nuestro documento. A cada sección le deberemos asignar un identificador distinto.

- 2 Especificar un enlace a cada una de las secciones que hayamos definido:

```
<A href="#identificador_sección"> Texto_enlace_a_sección </A>
```

Si pinchamos en este enlace, el navegador saltará automáticamente a la sección correspondiente.

3.5.5. ENLACES ESPECIALES: CORREO ELECTRÓNICO. ENLACES DE DESCARGA

La mayoría de las ocasiones utilizaremos los enlaces para cargar una página nueva en el navegador (bien sea en la misma ventana o en una ventana diferente). Sin embargo, existen otras opciones.

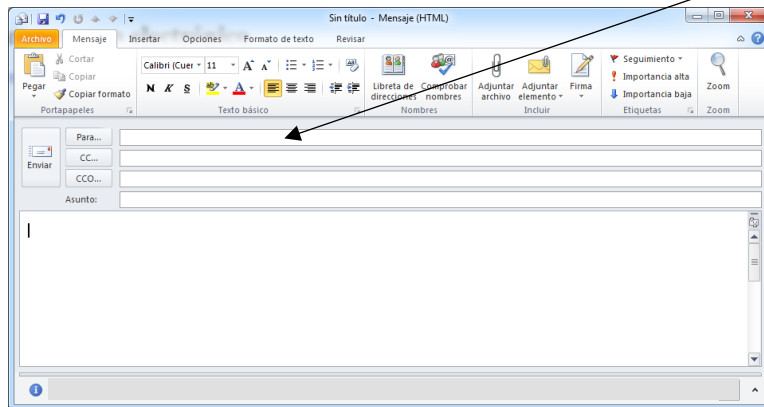
3.5.5.1. Enlaces de correo electrónico

Este tipo de enlaces hace que al pinchar sobre el enlace se abra un nuevo mensaje de correo en la aplicación cliente de correo electrónico que el usuario tenga instalada. En este mensaje, el campo para aparecerá cubierto con la dirección de mail que hayamos indicado a la hora de definir nuestro mensaje.



Al colocar el puntero del ratón, el navegador mostrará la dirección del correo electrónico.

Al picar sobre el texto del enlace, se abre la aplicación de correo electrónico que tenga el usuario instalado y en el campo para ya estará escrito el correo de envío



Para incluir un enlace de correo electrónico utilizaremos la etiqueta `<a>` con el atributo **mailto:**

```
<a href="mailto:miempresa@cursosepepa.com">Enviar correo</a>
```

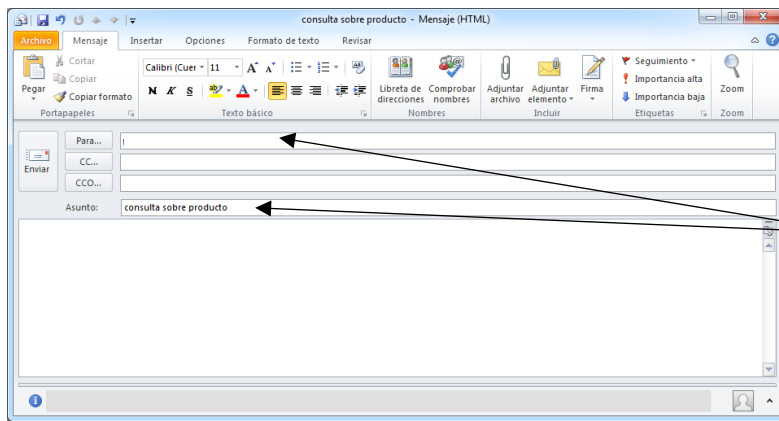
Dirección de correo electrónico que queremos poner como contacto	Texto que queremos visualice el usuario en el navegador

Nota. Es aconsejable que dentro del texto que ponemos para que lo visualice el usuario entre las etiquetas `<a>`, incluyamos la dirección de mail por si el usuario no tuviera configurado un programa de correo electrónico y lo quisiera copiar en un sistema de correo web-mail.

```
<a href="mailto:miempresa@cursosepepa.com"> miempresa@cursosepepa.com </a>
```

Podemos hacer que además del campo Para, al crear el mensaje de correo se cubra el asunto del mensaje. Para esto deberemos añadir el atributo `subject` en la etiqueta `<a>`.

```
<a href="mailto:miempresa@cursosepepa.com?subject=consulta sobre producto">  
miempresa@cursosepepa.com</a>
```



Al picar sobre el enlace, se abrirá la aplicación cliente de correo electrónico del usuario con un nuevo mensaje en el que ya pueden verse cubiertos el campo "Para" con la dirección especificada mediante el atributo `mailto` y el campo "Asunto" con el texto especificado en el atributo `subject`

3.5.5.2. Enlaces de descarga

Otro tipo de enlace muy habitual es el de descarga que hace que al picar sobre él, el usuario pueda descargarse un archivo a su ordenador y visualizarlo en local.

Para crear este tipo de enlaces deberemos utilizar el atributo `download`.

```
<a download href="reglamento.pdf">Descargar reglamento</a>
```

3.5.6. ATRIBUTOS ESPECÍFICOS: TÍTULO, DESTINO Y ATAJS DE TECLADO

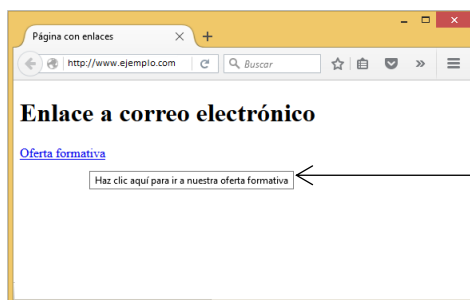
Además de los atributos vistos anteriormente, en la etiqueta de ancla podemos definir algunos otros que nos permiten definir aún más el enlace.

3.5.6.1. Atributo `title`

Este atributo se utilizará para ofrecer una descripción emergente del enlace.

```
<a title="Haz clic aquí para ir a nuestra oferta formativa"
href="ofertaformativa.html">Oferta formativa</a>
```

En el ejemplo anterior, al colocar el puntero del ratón sobre el enlace veríamos un mensaje emergente que pondría "Haz clic aquí para ir a nuestra oferta formativa".



Mensaje emergente que aparece al colocar el puntero del ratón sobre el enlace

3.5.6.2. Atributo `accesskey`

El atributo `accesskey` permite definir accesos directos desde el teclado para vínculos.

Para seguir el enlace utilizando el teclado, pulsaremos la tecla `<Alt>` y la tecla de acceso definida. Al pulsar `<Alt>+<tecla acceso>` el navegador seguirá el vínculo al que se haya asociado la tecla.

El atributo `accesskey` permite mejorar la accesibilidad de nuestra página.

Aunque no existe ningún estándar que nos diga qué teclas debemos asignar a cada característica, hay algunos accesos directos utilizados frecuentemente. Por ejemplo, la tecla 1 para ir a la página de inicio o la tecla 9 para ir al apartado de Contacto.

```
<a href="contacto.html" accesskey="9">Contacto</a>
```

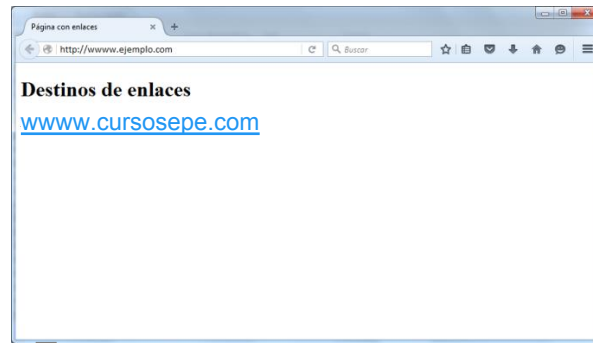
3.5.6.3. Atributo target

Este atributo nos permite cambiar el modo de abrir las páginas web. Por defecto, cuando el usuario hace clic sobre un enlace, la página web a la que hace referencia ese enlace sustituye a la página que el usuario estaba visualizando y que contenía dicho enlace. Sin embargo, a través del atributo target podemos cambiar este comportamiento.

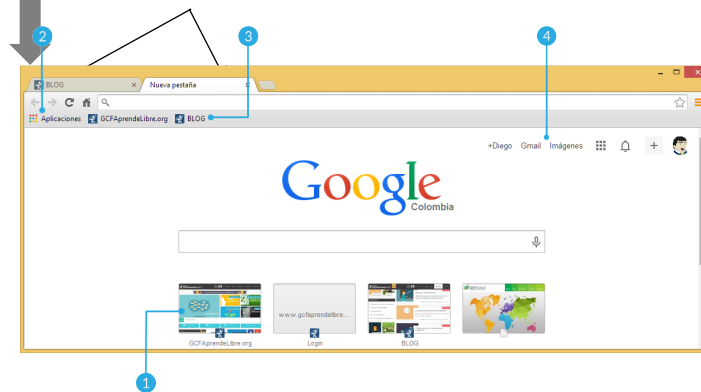
Al atributo target le podemos asignar varios valores, de los que, de momento vamos a ver dos:

- `_blank`. Abre una nueva ventana del navegador y en ella muestra el documento vinculado.

```
<a href="http://www.cursosepe.com" target="_blank">Visita www.cursosepe.com</a>
```



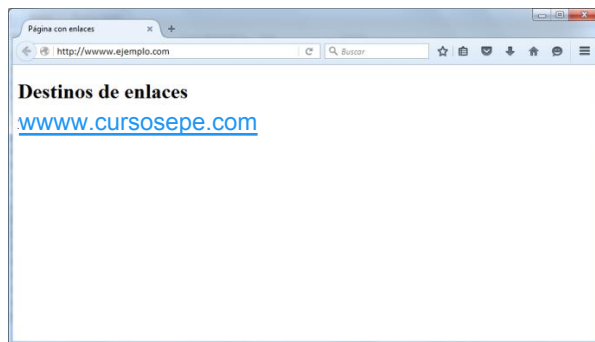
Vemos que al picar sobre el enlace se abre la página web solicitada en el mismo, pero en una pestaña nueva, de manera que el usuario verá dos pestañas: una con la página original y otra con la página que abrió al picar sobre el vínculo.



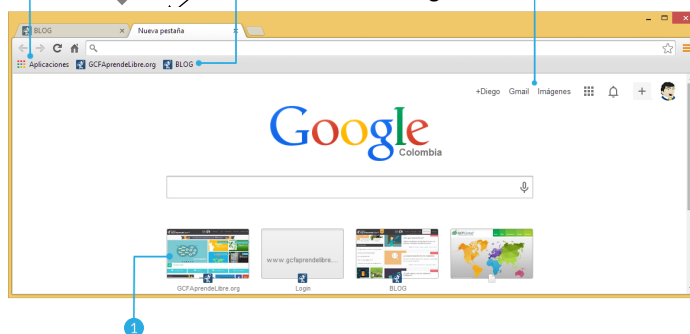
- `_self`. Abre el documento vinculado en la misma ventana donde se está visualizando el documento que contiene dicho vínculo (este es el comportamiento por defecto).

```
<a href="http://www.cursosepe.com" target = "_self">Visita www.cursosepe.com</a>
```

```
<a href="http://www.cursosepe.com">Visita www.cursosepe.com</a>
```



En esta ocasión, al picar sobre el enlace la página enlazada se abre en la ventana donde estaba abierta la página original, de manera que el usuario sólo verá la pestaña de la página enlazada. Si quisiera volver a la página original tendría que usar el botón "Atrás" del navegador



NOTA. El atributo target puede tomar más valores que comentaremos cuando veamos el tema de frames.

3.6. IMÁGENES

Una página web no sólo consta de texto, sino que en la mayoría de los casos, acompañando al texto nos encontramos imágenes.

En ocasiones, una imagen resultará mucho más descriptiva que largos párrafos de texto que pueden llegar a cansar al usuario, ya lo dice el refrán "Vale más una imagen que mil palabras". Aunque debemos evitar abusar de las imágenes, utilizarlas en su justa medida y cuando su uso esté justificado.

Para incluir una imagen en una página web utilizaremos la etiqueta `` con sus atributos correspondientes. Esta etiqueta no tiene etiqueta de cierre aunque se aconseja

La sintaxis básica de la etiqueta `` es la siguiente:

```

```

- url. Es la url de la imagen que se quiere mostrar. Esta url puede ser absoluta o relativa.

Por ejemplo,

```

```

La etiqueta `` es una etiqueta vacía, es decir, no delimita ningún texto por lo que podemos utilizar el atajo ya comentado para otras etiquetas para cerrarla: ``

3.6.1. FORMATOS DE IMÁGENES

HTML admite la inclusión de imágenes en diferentes formatos, aunque los más utilizados son: jpg, gif y png.

- **GIF**
 - Diseñado para comprimir las imágenes digitales y, así, reducir su peso.
 - Admite una paleta de hasta 256 colores. Esto nos permite optimizar el tamaño del archivo, pero no resulta muy adecuado para imágenes con cierta calidad.
 - Permite transparencias e imágenes animadas.
 - Es uno de los formatos más utilizados para publicar imágenes en la web por el bajo peso de las mismas.
- **JPG-JPEG**
 - Admite una paleta de hasta 16 millones de colores.
 - Es el formato más común junto con el GIF para publicar imágenes en la web.
 - Permite la compresión de imágenes, lo que hace que resulten imágenes de poco peso, pero esta compresión genera pérdida de calidad en la imagen (que será más apreciable cuantas más veces se guarde la imagen).
- **PNG**
 - Es un formato relativamente reciente que se está imponiendo rápidamente como alternativa a los dos formatos anteriores, ya que incorpora todas sus ventajas y evita sus inconvenientes.
 - Permite comprimir las imágenes sin pérdida de calidad.
 - Permite manejar transparencia.
 - Permite imágenes animadas.

3.6.2. CARACTERÍSTICAS DE IMÁGENES: TAMAÑO, TÍTULO, TEXTOS ALTERNATIVOS

A continuación veremos algunos atributos de la etiqueta `` que nos permiten controlar la manera en la que un navegador interpreta una imagen

3.6.2.1. *Tamaño de la imagen*

Si no indicamos otra cosa, al incluir una imagen en nuestra página, esta se mostrará con el tamaño original de la imagen. Sin embargo, podemos especificar un ancho y un alto para mostrar la imagen.

Esto lo podemos conseguir de dos formas: mediante los atributos `Width` y `Height` o mediante el atributo `Style`.

Atributos `Width` y `Height`

Estos dos atributos nos permiten indicar al navegador el ancho y el alto, en píxeles, que deberá tener la imagen.

```

```

- `value1`, `value2`. Será un número y representa el tamaño en píxeles que queremos asignar de ancho y alto a la imagen

```

```

Si solo ponemos uno de los dos parámetros, el navegador ajustará la medida del otro de manera proporcional.

```



```

Los tres comandos anteriores, hacen que se visualice la imagen con igual tamaño.

También se puede representar el ancho y alto en % respecto al tamaño de la pantalla

Atributo Style

Otra forma de especificar el ancho y alto de una imagen es mediante el uso del atributo style:

```

```

- value1, value2. Será un número y representa el tamaño en píxeles que queremos asignar de ancho y alto a la imagen.

```

```

Al igual que en el apartado anterior, si sólo se indica uno de los dos parámetros (width o height) y se omite el otro, el navegador ajustará el tamaño que no se indica a la misma proporción que el parámetro especificado.

Consejos

1. Es aconsejable que crees las imágenes con el tamaño adecuado para la web.
2. Tanto los atributos width y height, como el atributo style son válidos en HTML5. Sin embargo, es más aconsejable utilizar el atributo style ya que evita que las hojas de estilos cambien el tamaño original de las imágenes.

3.6.2.2. Título de la imagen

Este atributo se utilizará para ofrecer una descripción emergente de la imagen.

```

```

En el ejemplo anterior, al colocar el puntero del ratón sobre la imagen veríamos un mensaje emergente que pondría "Imagen con copyright. Prohibida su reproducción".



Mensaje emergente que aparece al
— colocar el puntero del ratón sobre el enlace

3.6.2.3. Textos alternativos

El atributo alt permite especificar un texto alternativo para una imagen, para el caso de que ésta no se pueda visualizar (ya sea porque la imagen no se cargue por alguna causa o por limitaciones del usuario, como por ejemplo una persona invidente).

Si una imagen no puede mostrar la imagen, se mostrará el texto definido en el atributo alt.

A continuación mostramos un ejemplo del uso de este atributo:

```

```



En esta página debería verse el logotipo de la academia, pero por algún problema el navegador no es capaz de mostrarlo, por lo que muestra el texto que se ha definido dentro del atributo `alt` y que no es otro que el de *“Logotipo de la academia”*

Las personas ciegas o con dificultades visuales, en muchas ocasiones utilizan screen readers (lectores de pantalla). Estos lectores de pantalla no son más que programas de software que pueden leer lo que el navegador está mostrando en la pantalla. Los lectores de pantalla son capaces de leer el texto incluido dentro del atributo `alt`.

3.6.2.4. Diferencias entre `title` y `alt`

El atributo `alt` muestra información sustitutiva de la imagen para el caso de que esta no esté visible mientras que el atributo `title` se emplea para proporcionar información adicional o complementaria, esta información no sustituye a la imagen.

Siguiendo las recomendaciones del W3C para hacer una página accesible, el `alt` debería ser obligatorio en todas nuestras imágenes sin embargo el `title` es opcional.

Tampoco es correcto incluir la misma información en los dos atributos.

3.6.3. ENLACES EN IMÁGENES

Como comentábamos en el apartado 3.5 de este mismo tema, podemos asociar un enlace a una imagen de manera que al hacer clic sobre ella, nos lleve a otro recurso.

Para asociar un enlace a una imagen utilizaremos la etiqueta `<a>` ya comentada:

```
<a href="url"></a>
```

Por ejemplo:

```
<a href="http://www.atcgijon.com"></a>
```

3.6.3.1. Mapas de imagen

A veces deseamos que una misma imagen tenga diferentes enlaces según la zona de la imagen sobre la que pique el usuario. Esto se consigue mediante los mapas de imágenes.

Para mapear una imagen, necesitaremos varias etiquetas:

- Añadir el atributo `usemap` a la etiqueta `` para indicar que la imagen es un mapa.

```

```

- La etiqueta `<map>` `</map>` que nos permite definir el mapeo de la imagen. Esta etiqueta llevará un nombre que deberá ser el mismo que se ha especificado en el atributo `usemap`.

```
<map name="mapa1">
```

- Entre las etiquetas <map> de apertura y </map> de cierre incluiremos las etiquetas <area> que nos permitirá definir las áreas que van a contener los enlaces dentro de la imagen. Esta etiqueta cuenta con varios atributos:
 - shape="rect/poly/circle". Permite definir el tipo de área. Puede contener los valores rect (área rectangular), poly (área poligonal) o circle (área ovalada).
 - coords="coordenadas". Permite definir las coordenadas del área. En las áreas rectangulares deben especificarse las coordenadas de la esquina superior izquierda y las de la esquina inferior derecha. En las poligonales especificaremos las coordenadas de todos los vértices del área. En las circulares indicaremos las coordenadas del centro del círculo y el valor del radio
 - href="URL". Indica la dirección de la página web a la que se accede si pinchamos en un área predeterminada.

```

<map name="mapa">
  <area shape="rect" coords="18,124,111,219" href="paginal.html" />
  <area shape="circle" coords="128,60,50" href="pagina2.html" />
  <area shape="poly" coords="139,200, 193,110, 236,199" href="pagina3.html" />
</map>
```

NOTA. En el caso de la forma rect, las coordenadas son las de la esquina superior izquierda y la esquina inferior derecha del cuadrado o rectángulo. En el caso de circle, las dos primeras coordenadas definen el centro del círculo y la tercera es el radio. En el caso de poly especificaremos las coordenadas de dos en dos de cada uno de los puntos que forma el polígono.

3.6.4. IMÁGENES DE FONDO

Generalmente, para poner una imagen de fondo en una página utilizaremos CSS pero aquí vamos a ver cómo lo haríamos utilizando exclusivamente html.

Para añadir una imagen de fondo desde html utilizaremos el atributo background de la etiqueta body:

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

donde fondo.jpg es el nombre del archivo de imagen. Dependiendo de dónde esté guardada la imagen, bastará con incluir el nombre y la extensión o deberá especificarse la ruta donde se encuentra la imagen. Más adelante veremos cómo hacer que la imagen se adapte a la pantalla a través de CSS.

