

Lenguajes de marcado para presentación de páginas web

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 inclusión de imágenes ni demás recursos multimedia. Algunos ejemplos son desde el bloc de notas propio de Windows, TextWrangler para Mac, que ofrecen funcionalidades muy limitadas, a otros mucho más completos, como los siguientes, por citar algunos:

- Visual Studio Code: <https://code.visualstudio.com/>
- Sublime text: <http://www.sublimetext.com/>
- Notepad++ : <https://notepad-plus-plus.org/>

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.

3.1 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 (<https://www.w3c.es/>) que se corresponden con las diferentes versiones de este lenguaje que se han ido estableciendo por este organismo.

Nivel 0: 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 <head> y gran parte de la etiqueta <body> (estilos lógicos, listas, enlaces, imágenes, cabeceras y el conjunto de caracteres especiales).

Nivel 1 (versión 1.0): añade algunos estilos físicos y lógicos. Recordemos que la estructura lógica de un documento está formada por las partes y relaciones que lo constituyen y que la estructura física indica el aspecto del documento. Esta definición del lenguaje era muy corta, por lo que se añadieron ciertos comando para mejorar el aspecto de los documentos web.

Nivel 2 (versión 2.0): la principal novedad fue la inclusión de los formularios.

Nivel 3 (versión 3.0): nace con el objetivo de evolucionar el lenguaje, pero la aparición de Netscape en la época llevó a que se definiera un nuevo estándar, y las mejoras se incluyeron en la versión 3.2.

Nivel 4 (versión 4.0): se propone en 1997 y trató de normalizar el lenguaje y añadirle una serie de mejoras propuestas por los fabricantes de navegadores. Durante casi diez años fue el último estándar.

A partir de la versión 4 el W3C se dedicó a desarrollar el estándar XHTML. La primera versión de este data del 2000 y se revisó en 2002. Surge 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.

Nivel 5: (versión 5.0). EL W3C lo reconoce en el 2007. No cambia el lenguaje en lo esencial, aunque sí 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 como video y audio. Otra de las novedades más importantes es la aparición del DOM (Document Object Model), una estructura que genera el navegador en el momento que carga un documento en html. Esta estructura se puede alterar, utilizando JavaScript, por ejemplo, con el propósito de modificar dinámicamente los contenidos y la apariencia de la página en cuestión. Más adelante en este curso estudiaremos con detenimiento el DOM.

3.2 Estructura de un documento. [SEP]

La estructura básica de un documento básico HTML es la siguiente:

`<!DOCTYPE html>` → Versión `<html>`

`<head>` → Cabecera . . .
`</head>`

`<body>` → Cuerpo . . .

`</body>`
`</html>`

Versión:

es lo primero que hay que incluir en cualquier documento escrito en HTML e informa al navegador acerca del tipo y versión del documento que se está creando. 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 de la versión que correspondan son interpretadas por el navegador. En caso de utilizar otras etiquetas no incluidas en la versión el navegador las ignorará.

HTML5

```
<!DOCTYPE html>
```

HTML 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

XHTML 1.0

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

Cabecera:

es uno de los dos grandes bloques de un documento html. Permite, 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.

Estas funcionalidades se consiguen utilizando las siguientes etiquetas:

- **<meta>**

Los metadatos son datos (información) sobre los datos. Los metadatos no se mostrarán en la página, pero podrá ser “parseable” por las máquinas, es decir, podrán analizar la sintaxis del documento para ver si se cumplen ciertas reglas o patrones o también se utilizarán por parte de los motores de búsqueda. Lo que se haga con la información que se pueda obtener de esta manera es otro tema diferente. La etiqueta meta no tiene cierre.

Ejemplos:

- Definir palabras clave en motores de búsqueda:

```
<meta name="keywords" content="HTML, CSS, XML, XHTML, JavaScript">
```

- Definir una descripción de la página:

```
<meta name="description" content="Free Web tutorials on HTML and CSS">
```

- Definir el autor de una página:

```
<meta name="author" content="Pepe García"> - Referescar el documento cada 30 segundos:
```

```
<meta http-equiv="refresh" content="30">
```

- El siguiente ejemplo es válido a partir de HTML5, el meta viewport permite adaptar el diseño a las pantalla de los diferentes dispositivos que pueden mostrar una determinada web:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- Aplicaciones o tecnologías utilizadas para desarrollar el documento:

```
<meta name="generator" content="Visual Studio Code">
```

• <title>

Sirve para escribir el título de la página web. Por lo general, el título de un documento web aparece en la parte superior de la venta o pestaña en la que se está visualizando la página.

• <link>

Define un enlace entre el documento y un recurso externo. Su uso general es enlazar páginas web con un archivo CSS determinado.

```
<link rel="stylesheet" type="text/css" href="theme.css">
```

Aunque es lo recomendable, 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.

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 es <body>.

El contenido puede ser extremadamente variado y lo estudiaremos detenidamente en los puntos siguientes y se incluyen: texto, tablas, enlaces, formularios, imágenes, vídeo y listas, por citar algunos ejemplos importantes.

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 estructura el contenido de las páginas web. La nueva especificación ha traído consigo una serie de etiquetas que sirven para estructura el contenido de los documentos web. Estas etiquetas consisten en capas o áreas dentro de las cuales se incluye el contenido que corresponda.



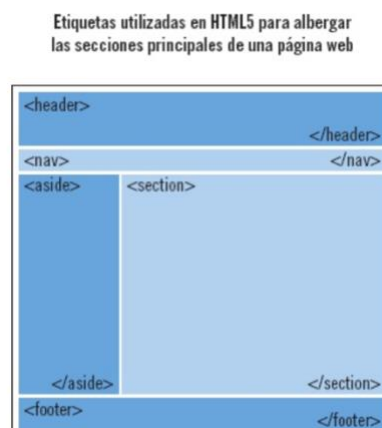
A continuación se muestra el ejemplo de diseño típico anterior pero esta vez con las etiquetas que en HTML5 delimitan cada uno de los bloques.

-`<header>`: Dentro de esta etiqueta se incluye la cabecera del contenido del un documento web.

No puede incluirse dentro de `<footer>` u otra `<header>`.

-`<nav>`: Se usa para albergar los elementos de navegación de un documento web.

-`<aside>`: Directiva para delimitar un área del documento en el que se desee añadir algo que complemente o aclare el texto y/o contenido principal al que hace referencia.



3.3 Color.

Como se ha comentado con anterioridad la estructura visual (el aspecto) de un documento web desde hace tiempo se recomienda encarecidamente separarlo de la estructura lógica del documento referenciando una hoja de estilo o CSS, no obstante en este punto vamos a hablar del color en una página web, y de cómo podríamos incluir cambios en este dentro del documento HTML.

- Codificación de colores:

La codificación de los colores dentro de un documento html se realiza a partir del sistema RGB, el cual usa 3 colores básicos: rojo (Red), verde (Green) y azul (Blue), y es un sistema aditivo, a partir de estos tres colores y sumando más colores obtendríamos todos los diferentes colores disponibles hasta alcanzar el blanco. Para cada uno de los colores básicos existen 256 valores posibles. Combinar estos 256 valores para los 3 colores básicos permite obtener un total de 16.777.216 colores diferentes. Para representar cada uno de los 256 valores se utiliza la codificación hexadecimal, pudiendo variar entre el 00 que correspondería a la ausencia de ese color hasta el FF que sería el valor máximo de ese color. Juntando los valores para cada uno de los 3 colores obtendríamos un código de 6 caracteres en el que los dos primeros corresponden al valor del rojo, los dos siguientes al verde y los últimos al azul. El código va precedido por la almohadilla, la forma general por tanto sería: #RRGGBB.

Para introducir un color dentro de un elemento del documento HTML utilizaremos el atributo style, dentro de este atributo lo que hacemos es declarar una propiedad de CSS para ese elemento en concreto, pero no para todas las etiquetas iguales, solo en la que declaramos el atributo. Por ejemplo, si quisiéramos que el contenido de un párrafo concreto apareciese en rojo utilizaríamos el siguiente código:

```
<p style="color:#FF0000">Párrafo</p>
```

- Colores tipo:

Algunos de los códigos hexadecimales más utilizados tienen nombre propio y puede utilizarse en lugar del código hexadecimal. Por ejemplo, el siguiente código modificaría el color del párrafo de la misma manera que el anterior.

```
<p style="color:Red">Párrafo</p>
```

A continuación se puede observar una tabla con los nombres de los colores tipo en HTML5:

DarkTurquoise 00CED1	MediumTurquoise 48D1CC	Turquoise 40E0D0	Cyan 00FFFF	Snow FFFAFA	GhostWhite F8F8FF	WhiteSmoke F5F5F5	Gainsboro DCDCDC
LightCyan E0FFFF	CadetBlue 5F9EA0	MediumAquamarine 66CDAA	Aquamarine 7FFFD4	FloralWhite FFFAF8	OldLace FDF5E6	Linen FAFAD2	AntiqueWhite FAEBD7
DarkGreen 006400	DarkOliveGreen 556B2F	DarkSeaGreen 8FBC8F	SeaGreen 2E8B57	PapayaWhip FFEDD5	BlanchedAlmond FFEBD3	Bisque FFEAF0	PeachPuff FFDAB9
MediumSeaGreen 3CB371	LightSeaGreen 20B2AA	PaleGreen 98FB98	SpringGreen 00FF7F	NavajoWhite FFDAB9	Moccasin FFE4B5	Cornsilk FFF8DC	Ivory FFFFF0
LawnGreen 7CFC00	Green 00FF00	Chartreuse 7FFF00	MedSpringGreen 00FA9A	LemonChiffon FFFACD	Seashell FFF5EE	Honeydew F0FFF0	MintCream F5FFFA
GreenYellow ADFF2F	LightGreen 32CD32	YellowGreen 9ACD32	ForestGreen 228B22	Azure F0FFFF	AliceBlue F0F8FF	Lavender E6E6FA	LavenderBlush FFF0F5
OliveDrab 6B8E23	DarkKhaki BDB76B	PaleGoldenrod EEE8AA	LightGoldenrodYello FADFAD	MistyRose FFE4E1	White FFFFFF	Black 000000	DarkSlateGray 2F4F4F
LightYellow FFFFE0	Yellow FFFF00	Gold FFD700	LightGoldenrod EEDD82	LightSlateGray 778899	Grey BEBEBE	LightGray D3D3D3	MidnightBlue 191970
Goldenrod DAA520	DarkGoldenrod B8860B	RosyBrown BC8F8F	IndianRed CD5C5C	CornflowerBlue 6495ED	DarkSlateBlue 483D8B	SlateBlue 6A5ACD	MediumSlateBlue 7B68EE
SaddleBrown 8B4513	Sienna A0522D	Peru CD853F	Burlywood DEB887	LightSlateBlue 8470FF	NavyBlue 000080	DimGrey 696969	SlateGrey 708090
Beige F5F5DC	Wheat F5DEB3	SandyBrown F4A460	Tan D2B48C	MediumBlue 0000CD	RoyalBlue 4169E1	Blue 0000FF	DodgerBlue 1E90FF
Chocolate D2691E	Firebrick B22222	Brown A52A2A	DarkSalmon E9967A	DeepSkyBlue 00BFFF	SkyBlue 87CEEB	LightSkyBlue 87CEFA	SteelBlue 4682B4
Salmon FA8072	LightSalmon FFA07A	Orange FFA500	DarkOrange FF8C00	LightSteelBlue B0C4DE	LightBlue ADD8E6	PowderBlue B0E0E6	PaleTurquoise AFEEEE
		Coral FF7F50	LightCoral F08080	Tomato FF6347	OrangeRed FF4500		
		HotPink FF69B4	HotPink FF69B4	DeepPink FF1493	Pink FFC0CB		
		MediumVioletRed CF1585	VioletRed DD6090	Magenta FF00FF	Violet EE82EE		
		Plum DDA0DD	LightPink FFB6C1	PaleVioletRed DB7093	Maroon 800000		
		Orchid DA70D6	MediumOrchid BA55D3	DarkOrchid 9932CC	DarkViolet 9400D3		
		BlueViolet 8A2BE2	Purple 800080	MediumPurple 9370DB	Thistle D8BFD8		

- Colores seguros:

En este grupo se adscriben los colores que se muestran de la misma manera independientemente del navegador o el sistema operativo utilizado. En total son 216, y surgen de combinar solo los siguientes valores hexadecimales para cada uno de los colores: 00, 33, 66, 99, CC o FF.

No existe ni la recomendación ni la obligación de limitarse a utilizar estos colores, ya que en la actualidad cualquier equipo informático puede mostrar millones de colores, y además, las variaciones que surgen de mostrar los colores no seguros en diferentes dispositivos, en su mayoría, apenas serán perceptibles. No obstante, si fuese una cuestión capital mostrar los colores de una manera determinada en cualquier dispositivo, utilizar estas 216 combinaciones sería la manera adecuada.

En cuanto a recursos online para obtener códigos de colores en hexadecimal podemos utilizar la siguiente web:

<http://htmlcolorcodes.com/es/>

3.4 Texto.

En este apartado se estudiarán las etiquetas para organizar, estructurar y dar estilo al texto de un documento HTML.

- Encabezados. jerarquía y estructura del contenido de un documento:

Los encabezados se utilizan para dar mayor jerarquía dentro de un texto a una determinada sección. Las etiquetas que se utilizan para tal efecto, en orden de importancia adjudicada son `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` y `<h6>`.

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.

- Párrafos:

HTML considera que los párrafos son bloques de texto. Para colocar textos en párrafos se utilizan el par de etiquetas `<p>` y `</p>`.

Por defecto, cada vez que se utiliza la etiqueta de párrafo y se cierra, el navegador insertará un salto de línea automático al final de cada uno de ellos. Los saltos de línea que se hacen dentro del propio código HTML no son interpretados como tales por el navegador. Más adelante veremos como insertar saltos de línea sin necesidad de crear nuevos párrafos.

Los navegadores hacen lo posible para adaptar el contenido de los párrafos a la ventana, a menos que los atributos `nowrap` o `noBr` estén explícitamente especificados.

Las siguientes características de estilo de texto se deberán controlar mediante una hoja CSS.

- Alineación, espaciado y sangrado de texto.
- Características de letra: tipos, tamaños y colores.
- 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 utilizar esta funcionalidad podemos utilizar la etiqueta `<hr/>`, que no necesita cierre. Para modificar el estilo, como anteriormente, deberemos hacerlo a través de una hoja de estilo.

- Etiquetas específicas para el marcado de texto. Estilos lógicos.

A continuación se estudian unas etiquetas que cambian el estilo del texto, pero al contrario que lo apuntado con anterioridad en referencia a los estilos, si tienen sentido utilizarlas en un documento definido con HTML5, por varias razones:

Las etiquetas lógicas permiten al navegador mostrar la información de la forma más apropiada para ese navegador. Un texto que necesite enfatizarse, se enfatizará con cursiva en Windows, y en negrita en Unix.

También ayudan al autor a controlar el contenido que quiere mostrar, sin distraerse con la presentación. Por ejemplo, si vamos a citar a alguien, podemos utilizar la etiqueta `<blockquote>` y los navegadores se preocuparán por mostrarnos de la forma que consideran adecuada ese contenido.

A continuación se listan las etiquetas lógicas más importantes y se explica su funcionalidad:

<code><address></code>	Especifica direcciones de email	<code></code>	Texto con énfasis
<code><blockquote></code>	Cita textual	<code><kbd></code>	Texto introducido por teclado
<code><cite></code>	Título de una obra (libro, artículo, etc)	<code><samp></code>	Mensaje de estado de un PC
<code><code></code>	Código fuente de un lenguaje	<code></code>	Texto tachado
<code><dfn></code>	Definición	<code></code>	Texto destacado

Las etiquetas que dividen las secciones del cuerpo `<header>`, `<footer>`, `<nav>`, etc también se consideran lógicas, aunque evidentemente no están relacionadas con el texto en sí, como si lo son las anteriores, y por eso ya las hemos comentado en la sección del cuerpo.

3.5 Hiperenlaces.

La característica que más influyó en el rápido desarrollo de la WWW y su revolución ha sido el hipertexto, que se puede definir como una herramienta que permite establecer enlaces o vínculos en los documentos web que sean capaces de direccionar al usuario hacia otra parte del documento web o hacia otro recurso web independiente.

- Estructura:

Para crear un enlace se utiliza la etiqueta `<a>`. Su sintaxis básica siempre utiliza el atributo `href`, en la que definiremos el destino del enlace. El texto del enlace será el que, al hacer clic sobre él, permita direccionar al usuario al destino definido por `href`. Por ejemplo:

```
<a href="http://www.coopsalus.com/">Coop Salus. Centre de  
formació</a>
```

La dirección que incluimos en el atributo `href` será una URL (Universal Resource Locator, localizador de recurso universal) que siempre tiene la siguiente estructura:

```
servicio://maquina.dominio:puerto/path/fichero
```

Donde:

- Servicio: es un protocolo de Internet, como puede ser: http, ftp, etc.

- Maquina.dominio: servidor que proporciona el recurso.
- Puerto: esta parte es opcional y, por lo general, no se utiliza.
- Path: ruta de directorios que es necesario seguir para acceder al recurso correspondiente. Para separar los distintos subdirectorios, es necesario utilizar el carácter “/”
- Fichero: nombre de la página o recurso que se desea abrir, acompañado de su correspondiente extensión

Por ejemplo, en este enlace:

```
<a href="http://www.xarxatecactiva.com/formacion/
uploads/2021/06/apuntes-html-junio.pdf">Apuntes HTML</a>
```

La URL sería:

```
http://www.xarxatecactiva.com/formacion/uploads/2021/06/
apuntes-html-junio.pdf
```

y los componentes serían:

Servicio: http://

Maquina.dominio: www.xarxatecactiva.com Puerto: no lo especifica.

Path: formacion/uploads/2021/06 Fichero: apuntes-html-junio.pdf

- Atributos específicos:

Además de href la etiqueta <a> admite varios atributos, se presentan los más importantes a continuación:

- title: en este atributo se establece un texto que se desea que aparezca junto al enlace en el momento 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 se abrirá en una nueva venta o pestaña.
 - _self: el documento se 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 Alt junto con la definida en este atributo, el navegador dirigirá al lugar donde se encuentre el enlace.

- Estilos de enlaces:

Por defecto los enlaces se muestran en azul y subrayados. Para modificar esta apariencia deberemos utilizar una hoja de estilos CSS.

- Diferencias entre enlaces absolutos y relativos:

Los enlaces absolutos son aquellos que contienen una dirección completa, incluyendo los nombres del servicio, dominio, ruta, etc. Un ejemplo sería el enlace que hemos visto con anterioridad.

En cambio, 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, la máquina y el dominio y, en muchos casos 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 desde el que se llama al enlace.

Por ejemplo, imaginemos que un sitio web está definido por dos documentos html, `pagina1.html` y `pagina2.html`. Si los dos estuvieran en el mismo directorio almacenados, para enlazar desde `pagina1.html` a `pagina2.html` la siguiente dirección en el atributo `href`:

```
<a href="pagina2.html">Página 2</a>
```

Si por el contrario `pagina2.html` estuviese dentro un subdirectorio llamado `documentos` dentro del mismo directorio que `pagina1.html` lo enlazaríamos así: `Página 2`

- Enlaces internos:

Los enlaces internos son hipervínculos que direccionan al usuario a otra sección del mismo documento donde se encuentra el enlace. Un ejemplo podría ser un enlace al pie del documento que vuelve al principio del documento. Para poder usarlos debemos previamente nombrar a la sección del documento en cuestión con el atributo `name`. Y en el enlace que creamos en `href` escribiremos el nombre que hayamos usado antecedido del carácter `#`.

Por ejemplo, imaginemos que desde una parte de nuestro documento queremos enlazar a un párrafo concreto; lo primero que haríamos sería nombrar el párrafo de la siguiente manera:

```
<p name="p1">...</p>
```

y luego en el sitio del documento donde queramos añadir el enlace utilizaremos el siguiente código:

```
<a href="#p1">Texto del enlace</a>
```

También podríamos querer enlazar a una parte concreta de otro documento web, no dentro del mismo. Para hacerlo la referencia de `href` será del nombre del archivo web seguido de `#` y el nombre que hayamos utilizado para esa sección concreta del documento. Por ejemplo, si el anterior párrafo se encontrase dentro de `pagina2.html` el enlace quedaría así:

```
<a href="pagina2.html#p1">Texto del enlace</a>
```

Enlaces hacia otros archivos:

Exactamente igual que los enlaces a documentos web, solo que los archivos tendrán una extensión diferente a .htm o .html. El navegador procederá a abrirlo, si pudiese, como por ejemplo un archivo mp3 o una imagen. Si no pudiese ejecutarlos por si mismo automáticamente iniciaría su descarga. Por ejemplo:

```
<a href="documentos/casa.jpg">Página 2</a>
```

El enlace anterior abriría en el navegador la imagen casa.jpg. Posteriormente el usuario podría descargarla también desde el navegador.

Enlaces de correo electrónico: 

Para hacer que un enlace apunte a una dirección de correo electrónico es necesario que en el atributo href aparezca la cadena mailto: seguida de la dirección de mail destino. Además podemos configurar información adicional con algunas cadenas reservadas para ello, siempre y cuando aparezcan precedidas del carácter ?. Podemos configurar fijo el asunto del mail, podemos añadir direcciones con copia...

Si utilizamos varias de estas cadenas debemos usar el carácter & para unirlos. Por ejemplo:

```
<a href="mailto:info@coopsalus.com?subject=Solicitud de
informacion&cc=info2@coopsalus.com">Enviar correo</a>
```

Este enlace enviaría un mail a info@coopsalus con copia a info2@coopsalus.com con el asunto Solicitud de información.

En el momento en que se hiciera clic sobre el enlace el navegador abriría el programa asociado por defecto para mandar correo electrónico en nuestro sistema operativo para que pudiéramos escribir el cuerpo del correo y enviarlo.

3.6 Imágenes.

En este apartado estudiaremos la manera de insertar y configurar imágenes dentro de un documento web html, un aspecto muy importante a tener en cuenta en el diseño y desarrollo web profesional.

- Formatos de imágenes.

Los archivos de imágenes se clasifican, normalmente, por el formato es que se codifica la información y su como se comprime esta. Los formatos más habituales en las páginas web son el GIF, el JPEG y PNG.

El formato GIF se utiliza principalmente porque permite hacer animaciones. El JPEG porque su relación calidad/tamaño es muy buena y el PNG se utiliza cuando queremos utilizar transparencias, además tienen la cualidad de que los navegadores previsualizan las fotos antes de estar completamente cargadas, a diferencia de los otros formatos.

- Características de las imágenes.

Para insertar imágenes en nuestros documentos usaremos la etiqueta , cuya sintaxis básica es la siguiente (no necesita etiqueta de fin):

```

```

Los principales atributos de esta etiqueta son:

- `src`: especifica la URL donde se ubica el archivo de la imagen a insertar.
- `title`: se utiliza para añadir un texto que aparecerá cuando el cursor se sitúe encima de la imagen.
- `alt`: en caso de que el navegador no pueda cargar la imagen (porque no pueda acceder al a URL o la imagen se haya borrado, o alguna otra causa) el texto que guardemos en este atributo se mostrará. Lo correcto es añadir una descripción completa de la imagen.
- `width`: especifica el ancho de la imagen en píxels.
- `height`: especifica la altura de la imagen es píxels también.

3.7 Listas.

Consisten en una sucesión de elementos precedidos por una viñeta (listas no ordenadas) o un número (listas ordenadas).

- Características.

Listas no ordenadas: son aquellas cuyos elementos van precedidos por algún símbolo o imagen. Para crear la lista se usa la etiqueta `` y para añadir cada elemento de la lista la etiqueta ``, así un ejemplo de código para una lista de este tipo podría ser el siguiente:

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

Para modificar el aspecto de las viñetas puede usarse el atributo `type` del etiqueta `` que puede tener los valores `disc` (círculo de color negro), `circle` (círculo con fondo blanco) y `square` (cuadrado).

- Ordenación de listas.

EN este caso la etiqueta a utilizar es `` y, al igual que las anteriores, cada elemento que añadamos con ``. En estas listas, los símbolos que precederán a cada uno de los elementos son de carácter numérico y se irán generando automáticamente conforme se añadan puntos.

Se puede establecer el número inicial a partir del cual se van a contabilizar los elementos de la misma. Esto se consigue gracias al atributo `value`. Se incluye dentro de una etiqueta `` y los siguientes puntos que se escriban se generarán automáticamente por orden, partiendo del número indicado en dicho atributo.

El tipo de la numeración también se puede modificar con el atributo `type`, con las siguientes opciones:

- 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, ...
- Anidamiento en listas.

Se pueden establecer listas anidadas, es decir, listas que contengan otras listas dentro de ellas. Para hacerlo, basta con entender que se puede definir una lista como elemento de otra lista. Por ejemplo, de la siguiente manera:

```
<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>
```

3.8 Tablas.

Se utilizan para estructurar datos y presentar información de manera ordenada y relacionada. En este apartado, se estudiará la manera de implementar y configurar este tipo de elementos en los documentos web.

- Estructura básica.

Una tabla en HTML se puede definir 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. Se definen a partir de tres etiquetas:

- - `<table>`: contenedor principal.
- - `<tr>`: necesitamos una por cada fila.
- - `<td>`: se utiliza una por cada celda. Dentro se escribe el contenido que se desea incluir en la

misma.

Por ejemplo, este código: `<table>`

```
<tr>
    <td>Celda1</td>
    <td>Celda2</td>
    <td>Celda3</td>
</tr>
<tr>
    <td>Celda4</td>
    <td>Celda5</td>
    <td>Celda6</td>
</tr>
<tr>
    <td>Celda7</td>
    <td>Celda8</td>
    <td>Celda9</td>
</tr>
```

`</table>`

daría lugar a la siguiente tabla:

Celda 1	Celda 2	Celda 3
Celda 4	Celda 5	Celda 6
Celda 7	Celda 8	Celda 9

- Agrupamiento de filas y columnas.

En HTML existe un conjunto de etiquetas que permite establecer ciertas agrupaciones en las filas y columnas de las tablas para así poder ejercer un mayor control sobre la visualización mediante una hoja CSS. Las etiquetas son:

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

En cuanto a las agrupaciones de columnas tenemos:

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

Con el uso del atributo `span` de la etiqueta `<colgroup>` se puede especificar el número de columnas que va a tener dicho grupo.

- Buenas prácticas en el uso de tablas.

Aunque antes se solían utilizar las tablas para maquetar los contenidos de las páginas web, las tablas deben usarse únicamente en el caso de que se deseen tabular datos. El uso de hojas CSS para posicionar los diferentes elementos que forman parte de un sitio web proporcionar una serie de ventajas en relación con la utilización de tablas para tal fin:

- Se logra separar el contenido de la presentación, ya que si utilizamos tablas, no habría separación estaría todo en el mismo archivo.
- La utilización de CSS minimiza considerablemente la cantidad de etiquetas HTML necesarias para maquetar una web, lo cual reduce el tiempo de carga por parte del navegador.
- Al separar el contenido de la presentación la página será rastreada mejor por los buscadores.
- El tiempo empleado para maquetar una página mediante tablas es mucho mayor que con CSS, es muy engorroso y poco "limpio".
- Tener separado el contenido del estilo ayuda a poder actualizar el aspecto de una web sin tener que alterar la estructura de la página.

3.9 Formularios.

Los formularios son secciones interactivas de una página web que los autores insertan en las 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 y lo veremos más adelante en el curso.

- Descripción general y uso 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.

Todo formulario se crea a partir de la etiqueta `<form>`, dentro de la cual se van insertando los diferentes elementos que formen parte de él y que más adelante describiremos. Esta etiqueta presenta dos atributos que deben incluirse obligatoriamente, son:

- `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 atributos y opcionales:

- `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.

- Elementos de un formulario.

La etiqueta `<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>:

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.
- `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 el archivo que se quiera enviar desde el explorador de archivos del sistema operativo correspondiente. Cuando se utilizan este tipo de elementos, es necesario incluir, en el atributo `enctype` de la directiva `<form>`, el valor `multipart/form-data`.
- `image`: corresponde a un botón de envío personalizado que consiste en una imagen definida por el atributo `src`.
- `password`: consiste en una casilla de textos cuyos caracteres escritos aparecen como asteriscos (el símbolo depende del navegador) para ocultar el texto que se escribe.
- `radio`: elemento similar al `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 de que se seleccione.
- `reset`: ese botón sirve para borrar todos los datos introducidos por el usuario, restaurando el formulario a su estado inicial.
- `submit`: botón de envío del formulario. El texto que aparece escrito sobre el botón puede definirse en su atributo `value`.
- `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`.

- <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: `rows` y `cols`.

- `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`).

- `<select>`:

Estos elementos consisten en listas desplegables de opciones que son mostradas al hacer clic en un pequeño botón adjunto a los mismo. 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 las correspondientes 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.