Unidad 2: Lenguajes para la visualización de información

1.1. El modelo de objetos del documento
1.2. Lenguaje HTML
1.3. Lenguaje XHTML
1.4. Hojas de Estilo

1.1. El modelo de objetos del documento

El Modelo de Objetos del Documento (DOM) es un API (Application Programming Interface) estándar del W3C para documentos HTML y XML. Nos va a mostrar una representación estructural del documento que nos permitirá modificar su contenido. Esencialmente comunica a las páginas web con los scripts ó los lenguajes de programación.

Con el DOM de un documento los programadores pueden construir documentos, navegar por su estructura y añadir, quitar ó modificar elementos y contenidos

Una página web es un documento HTML que es interpretado por los navegadores en forma gráfica permitiendo también el acceso al código. El DOM nos permite ver el mismo documento pero de otra forma es decir describiéndolo como un conjunto de objetos para que un programa JavaScript pueda acceder a ellos

El DOM permite acceder a la estructura de una pagina HTML mediante el mapeo de los elementos de esa página en un árbol de nodos cada elemento se convierte en un nodo , que se suele asociar a una etiqueta, puede tener nodos hijos y atributos y cada porción de texto en un nodo de texto que no va a poseer ni nodos hijos ni atributos

Como modelo de objetos el DOM identificará:

➔ las interfaces y objetos usados para representar y manipular un documento

➔ la semántica de las interfaces y objetos incluyendo el comportamiento y atributos

➔ las relaciones y colaboraciones entre estas interfaces y objetos

Actualmente el DOM consiste en dos partes: el núcleo del DOM que representa la funcionalidad usada para los documentos XML y además sirve de base para el DOM HTML

1.2. Lenguaje HTML

El lenguaje HTML (Hiper Text Markup Languaje) es el lenguaje que se utiliza para escribir la mayoría de las páginas web en internet.

El consorcio **W3C** define HTML como ***un lenguaje reconocido universalmente y que permite publicar información de forma global.***

HTML se creó en un principio con objetivos divulgativos. En aquel momento no se pensó que la web llegaría a ser un área de ocio con carácter multimedia, por lo que entonces se creó para lo que hacia falta y sin dar respuesta a todos los posibles usos que más adelante se le iba a dar (como por ejmplo tiendas online, banca, buscadores etc) y sin pensar en todos los colectivos de gente que lo utilizarían en un futuro.

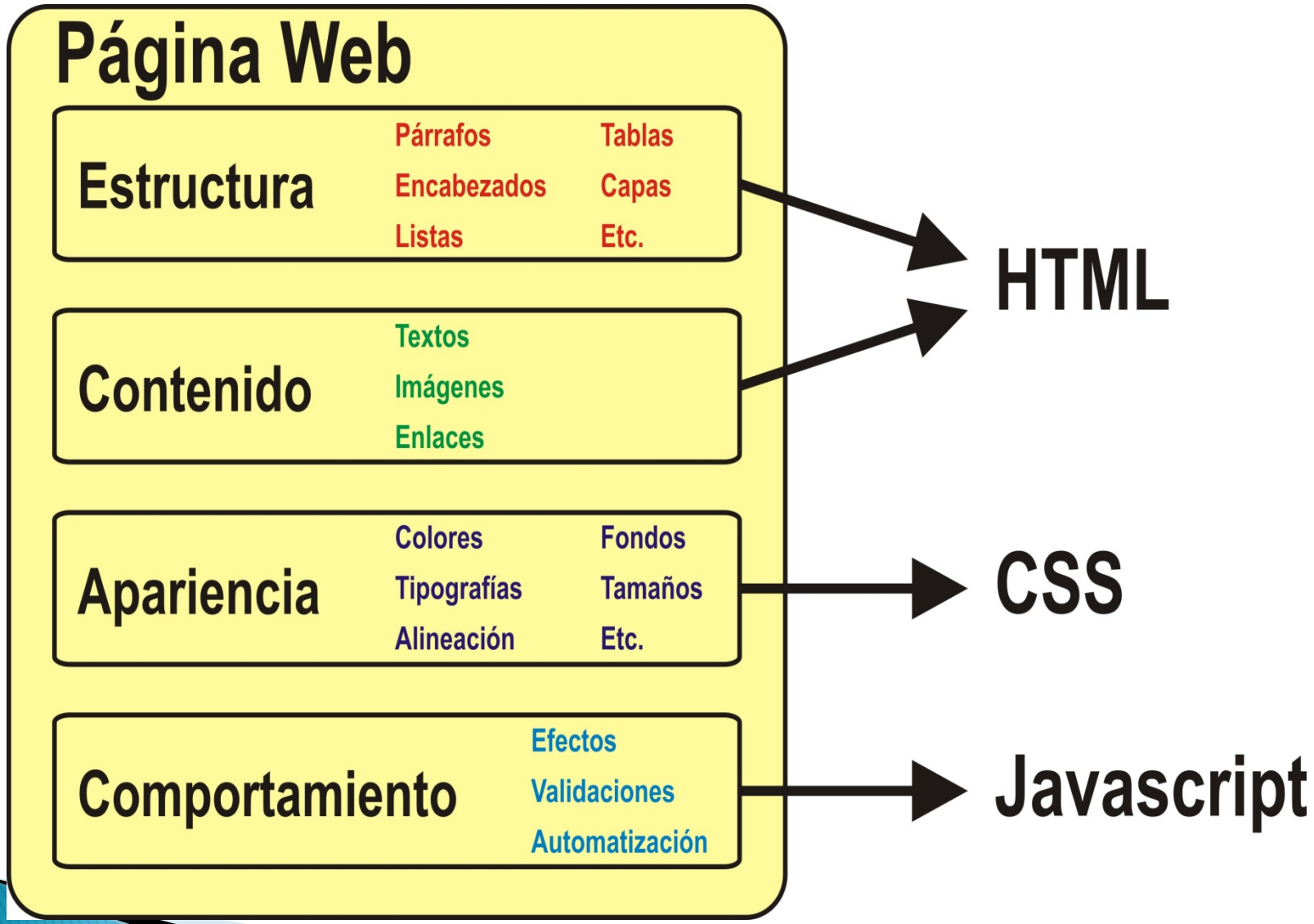
Como ya hemos mencionado en el tema anterior **HTML nace al comienzo de los 80** pero no se convirtió realmente en **estándar hasta la versión HTML 2.0 en 1995** (es decir quince años más tarde)

Después fueron apareciendo versiones en las que se añadieron nuevas funcionalidades para intentar dar servicio a las necesidades que iban surgiendo. Así en 1997 aparece la versión HTML 3.2 , en 1998 la versión HTML 4.0 y en 1999 la versión 4.1.

En 2008 se publicó un borrador de la versión HTML 5 en la que todavía se esta trabajando

Cuando, como usuarios, estamos metidos en una página web podemos comprobar que código tiene escrito sin más que pulsar botón derecho y seleccionar la opción *"Ver código fuente"*

1.2.1. Esquema real en las paginas web actuales



1.2.2 Características del Lenguaje HTML

- No es un lenguaje de programación sino un lenguaje de marcado que sirve para estructurar y publicar documentos (paginas web) y también para especificar hipervínculos.
- Esta basado en etiquetas ó marcas que sirven para establecer una serie de normas de formato en un trozo determinado de texto
- Esas etiquetas **"tags"** también van a permitir la introducción de diferentes objetos en la página web como por ejemplo: texto, imágenes, sonido, videos, enlaces a otros documentos, enlaces a servicios de correo electrónico, enlaces a servicios FTP etc.
- Los documentos escritos en lenguaje HTML se almacenan en formato de texto ASCII y deben tener la extensión **.html** ó .htm
- Para **editar** los documentos (es decir crearlos, modificarlos..) solo necesitamos un editor de textos ASCII sencillo como por ejemplo el block de Notas ó **Notepad**++. Siempre podremos usar cualquier otro editor si nos permite salvar los documentos como ficheros de texto planos.
- También se suele usar lo que se llama un **editor WYSIWYG (***"What you see is what you get"* que significa "lo que ves es lo que tienes" para elaborar páginas Con los editores WYSIWYG podremos diseñar el documento visualmente y sin tener que ir poniendo etiquetas (aparentemente más fácil) pero si no se aprende HTML con sus distintas etiquetas y especificaciones no se podrá saber retocar un documento ó adaptarlo cuando sea necesario a nuestras necesidades. Un ejemplo muy conocido de editor de este tipo es el Microsoft Front Page aunque casi esta obsoleto ó el Kompozer
- Para **ver** (ejecutar) los documentos **html** se necesita un navegador como por ejemplo **Internet Explorer (ó Mozilla)** que van a ser capaces de interpretar las etiquetas del documento (con algunas diferencias entre ellos) y nos muestran la página tal y como la hemos especificado con esas etiquetas

Normas del Lenguaje HTML

HTML tiene una serie de normas ó recomendaciones que nos sugieren como se deben escribir los documentos para que el navegador pueda interpretarlos.

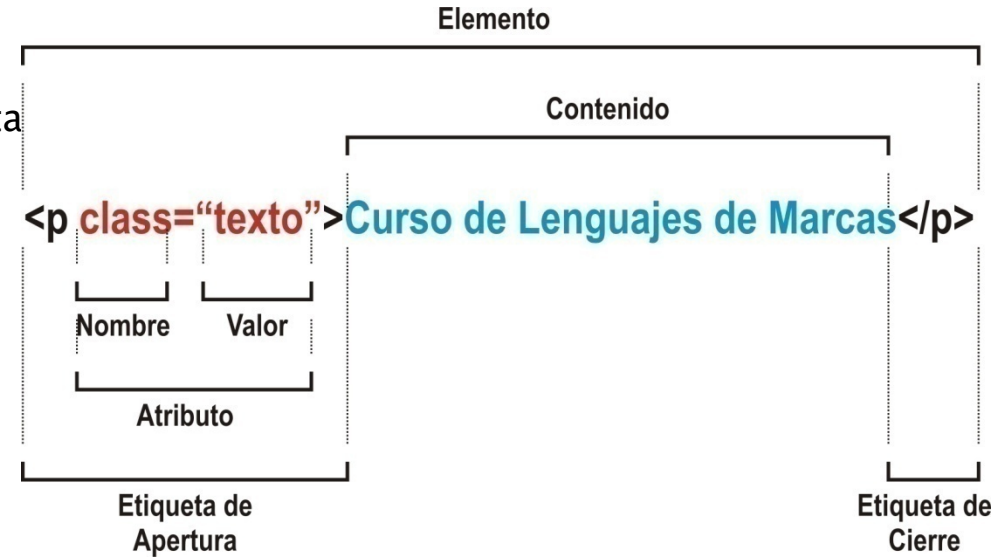
Aunque los navegadores pueden procesar casi cualquier documento va a ser importante mantener en lo posible las reglas de sintaxis que sugiere HTML y que son:

- Los documentos html son documentos **estructurados y organizados**
- Los **nombres de** los elementos ó **etiquetas no son sensibles a mayúsculas ó minúsculas**
- Los **nombres de** los **atributos no son sensibles a mayúsculas ó minúsculas**
- Los **valores de los atributos pueden ser sensibles a mayúsculas ó minúsculas** especialmente si son nombres de archivos hay que ponerlos tal y como están guardados
- Los **nombres de las etiquetas no deben tener espacios en blanco**
- Los **valores de los atributos pueden contener espacios si van entre comillas y si contienen valores que no sean alfanuméricos (números y letras) irán entre comillas**
- HTML toma **cualquier numero de caracteres en blanco como un único carácter** a no ser que estén dentro de la etiqueta <pre>
- Las **etiquetas que llevan contenido se deben cerrar**
- Las **etiquetas se pueden anidar**
- Los **exploradores omiten elementos desconocidos** ya sean etiquetas ó atributos

Actualmente se aceptan tanto mayúsculas como minúsculas pero es mejor elegir un estilo fijo para que sea más legible. **Es conveniente para acostumbrarse poner tanto las etiquetas como los atributos en minúsculas ya que XML y XHTML solo aceptan minúsculas.** Además las hojas de estilo en cascada CSS se hacen también en minúsculas. En cuanto a los atributos aunque no sea obligatorio ponerlos entre comillas en HTML nos acostumbraremos a ponerlos

1.2.3. Elementos, etiquetas y atributos de HTML

- HTML esta formado por **elementos** aunque a veces se confunde el término elemento con el de etiqueta. Realmente un elemento va a estar formado por:
  - una etiqueta de apertura,
  - cero ó mas atributos
  - texto encerrado en la etiqueta
  - una etiqueta de cierre



Existen algunos elementos que pueden no tener la marca ó etiqueta de cierre y se van a llamar elementos **vacios**

- Todas las **etiquetas, marcas ó tags**, de HTML comienzan por el símbolo < y acaban con el símbolo > . Puede haber etiquetas de inicio <title> ó de fin </html>
- Las etiquetas de inicio pueden llevar cero, uno ó más **atributos** que estarán formados por pares de **nombredelatributo-elvalorquetoma**, separados por el símbolo igual (=) y el valor va a ir entre comillas.

Por ejemplo:

```
<title>
<h1   align= "center">
<hr    color= "red"  size ="6" >
```

1.2. 4. Estructura básica de una página web

- Generalmente cualquier página HTML está contenida entre las etiquetas **<html> y </html>** y se dividen en dos partes cabecera y cuerpo.
- **La cabecera** se especifica con la etiqueta **<head> …. </head>** y va a incluir información sobre la propia página, como por ejemplo su título y su idioma. Todo lo que se pone ahí a excepción del título no va a ser visto por el usuario.
- **El cuerpo** se especifica con la etiqueta **<body> ….. </body>** e incluye todos sus contenidos, como párrafos de texto, imágenes, enlaces etc. Todo lo que será visto por el usuario
- Al comienzo del documento y fuera de la etiqueta <html> se deberá poner una etiqueta especial llamada **DOCTYPE** que sirve para indicar el estándar al que va a pertenecer el documento que estamos escribiendo. Es decir en esta etiqueta especificaremos el DTD que hemos usado al escribir la página por si es preciso que nuestra página sea validada en algún lugar.
- Como ya veremos más adelante, para que una página **(X)HTML sea correcta y válida es imprescindible que incluya el correspondiente DOCTYPE indicando el DTD utilizado.**
- Un DTD, del inglés ***document type definition***, va a contener cuales son los tipos de elementos y atributos que vamos a poder utilizar en nuestra página web y también define las reglas de cómo se pueden utilizar estos elementos juntos así como declara los juegos de caracteres permitidos.
- Los validadores van a probar que si la página web esta correctamente escrita de acuerdo al DTD que hayamos puesto en la declaración DOCTYPE, utilizando el identificador del sistema o implicitamente usando el identificador público
- A continuación se muestran los distintos DTD que se pueden utilizar al crear páginas HTML:

<!doctype html public "-//IETF//DTD HTML 2.0//EN“>

Cumple el estándar HTML 2.0 y lo soportan extensamente los navegadores pero este estándar por ejemplo no soporta tablas, marcos etc

<!doctype html public "-//W3C//DTD **HTML 3.2 Final**//EN">

Cumple el estándar HTML 3.2, lo soportan la mayoría de los navegadores pero esta limitado en cuanto a las hojas de estilo y no soporta marcos ni la internacionalización

<!doctype html public "-//W3C//DTD **HTML 4.01 Transitional** //EN"
"http://www.w3.org/TR/html4/loose.dtd">

Cumple el estándar HTML 4.0 y además soporta elementos desaconsejados y obsoletos.

Se aconseja a la hora de convertir desde una version vieja a una nueva

<!doctype html public "-//W3C//DTD **HTML 4.01**//EN"
"http://w3.org/TR/REC-html401/**strict.dtd**">

Cumple el estándar HTML 4.0 y además no contiene elementos desaconsejables,

Este es el modo recomendado por el consorcio W3C ya que se interpretará de forma correcta por todos los navegadores y por otro lado haría más fácil el paso a XHTML que posiblemente sustituirá al HTML en los próximos años

<!doctype html public "-//W3C//DTD **HTML 4.01 Frameset**//EN“
http://w3.org/TR/REC-html40/frameset.dtd>

se usa cuando realizamos una declaración de marcos ó frames

1.2.5. Ejemplo de pagina web simple

- Veamos una página web sencilla

```
!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN“
                        "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
        <title> Primera página  de ---- </title>
</head>
<body>
        Esto es una prueba muy básica
        de cómo se hace una página web
</body>
</html>
```

- Ahora copiarla en vuestro editor. Recordar que el fichero se debe guardar con extensión **.html** (por ejemplo **prueba.html**) y después ejecutarla para ver que sale
- ¿Buscar en la pantalla donde aparece lo que habéis puesto dentro de la etiqueta **title**?
- ¿Cómo aparece el texto que hay dentro del cuerpo (body) de la página?
- Ahora busquemos una pagina web en internet un poco menos sencilla por ejemplo: Una vez en ella darle al botón derecho **“Ver código fuente “** y buscar en ella alguna de esas etiquetas que hemos utilizado.

1.2.6. Cabecera <head>

- La cabecera de una página es el lugar donde se define el título de la página y otra información sobre la página para que los motores de búsqueda como Google establezcan la ubicación de la página, también añade estilos y script.
- Tiene pocas líneas y etiquetas y a excepción del título la demás información, llamada metainformación, no es visible para el visitante
- Las etiquetas mas generales que pueden aparecer son:
  - **<title>** ... **</title>** define el título del documento HTML y los navegadores lo muestran como el título de la propia ventana. Cada página debe tener un título corto, único y adecuado al contenido que muestra. Cuidado no confundir con el atributo title de algunas etiquetas
  - **<meta>** ... **</meta>** permite especificar información de interés como: autor, fecha de publicación, descripción, palabras claves, etc.

    La etiqueta <meta> se usa para indicar información sobre la página que será utilizada por los buscadores para intentar encontrar coincidencias con lo que el usuario pretende encontrar.

    Con esta etiqueta pueden especificarse los atributos **name** indica el tipo de información y **content** indica el valor de dicha información. No existe ninguna especificación de la W3C que defina los posibles valores para el atributo name pero si son algunos que son estándares como (description, keywords, author etc)

    Veamos algunos ejemplos muy comunes

```
<meta name="description" content="El objetivo de este tutorial es conocer HTML" />
<meta name="keywords" content="HTML, CSS, XML, JavaScript" />
<meta name="author" content="Juan Pablo" />
<meta name="generator" content=" NotePad++ " />
```

    También se usa mucho el atributo **http_equiv** para definir la codificación de los carácteres del documento

```
<meta http-equiv="Content-Type" content="texto/html;charset=ISO-8859-1" />
```

- En la cabecera de una página también pueden aparecer otras etiquetas como son:

  - **<script>**
  - **<link>**

  que van a servir para enlazar recursos que se deben cargar de forma automática y que ya veremos
  Cuando el navegador encuentra alguna de estas dos etiquetas, descarga los recursos a los que enlazan y los aplica a la página web.
  La etiqueta <script> tiene dos modos de funcionamiento, ya que se emplea tanto para insertar un bloque de código Java Script en la página como para enlazar un archivo tipo JavaScript externo.

- Existen también otras etiquetas que pueden aparecer en la cabecera que son por ejemplo
  - **<style>** … **</style>** especifica un estilo CSS para ser utilizado en el documento
  - **<base** href="http://www.marca.com/textos"> para URL relativas

  COMENTARIOS: en HTML

  En un documento html se pueden escribir comentarios para lo cual usaremos este formato

  <!-- escribimos todo lo que queramos -->

Elementos de bloque y elementos de línea

- La mayoría de los elementos (etiquetas) que pueden aparecer en el cuerpo de un documento se pueden clasificar en elementos a nivel de bloque y elementos a nivel de texto ó inline según las siguientes especificaciones del W3C
  - Generalmente los elementos de bloque pueden contener otros elementos de bloque y elementos de línea mientras que los de línea solo pueden contener texto y otros elementos de línea. Eso lleva consigo que los elementos de bloque van a organizar el texto en estructuras mas grandes que los de línea
  - Por defecto los elementos de bloque se formatean de distinta forma a los de línea es decir generalmente comienzan en una nueva línea
  - Son elementos de bloque:

    address, blockquote, center, dir, div, dl, fieldset, form, h1—h6, hr, is, menu, noframes, noscript, ol, p, pre, table, ul y tambien **dd,dt, li , tbody, td, tfoot, th, thead,tr**
  - Son elementos de línea:
  - **a, abbr,acronym, applet, b, basefont, bdo,gig, br, cite, code, dfn, em, font, i, img, input, kbg, label, q, s, samp, select, small, span, strike, strong, sub, sup, tt, u, var**
  - Según las circunstancias los siguientes pueden ser en linea y de bloque

    button, del, iframe, ins, map, object, script.

- Existen **dos atributos** que pueden ir asociados a casi todos los elementos pero que veremos mas adelante: **id** que sirve para asignar un nombre a un elemento en el documento (nombre único que no puede repetirse en el documento) y **class** que sirve para asignar uno ó más nombres de clase a un elemento
- Existen **dos etiquetas** que son ** y ** que constituyen un recurso muy potente a la hora de estructurar documentos pero aunque son de HTML se usan más en combinación con las hojas de estilo CSS y las veremos más adelante.

1.2.7. Cuerpo de la página body

- De momento en la etiqueta <**head**> (cabecera) solo usamos la etiqueta <**title**> que es obligatoria ya volveremos a las demás más adelante
- La etiqueta <**body**> (cuerpo) del documento html va a contener a toda la página.
- Puede llevar los siguientes atributos:
  - **bgcolor** que sirve para dar un color de fondo a la página. El valor que toma este atributo puede ser un literal en ingles (red, blue etc) ó un valor hexadecimal con # delante
  - **background** sirve para insertar una imagen como fondo de la página (puede ser una URL)
  - **text, link, vlink y alink** dando un valor a estos atributos podemos modificar el color de texto, vínculo (por defecto azul), vínculo visitado (por defecto morado) y vínculo activo (por defecto rojo) respectivamente para toda la página
  - **leftmargin, topmargin, marginwidth, marginheigth** con estos atributos se establecen los márgenes para la página (izquierdo, superior, anchura y altura respectivamente, los dos últimos no van con IE). A veces se ponen los valores de leftmargin y topmargin a cero y realmente lo que se hace es eliminar los márgenes de la página (nota según navegadores)
  - **scroll (yes/no)** no es estándar permite quitar ó poner las barras de desplazamiento en pantalla si el contenido sobrepasa las dimensiones de la pantalla

Ejemplos de codificación de valores de atributos de body

```
<body bgcolor = “red” > ..... </body>
<body bgcolor = “yellow” text =“blue”> .....</body>
<body background = “imagen.jpg” > ..... </body>
<body leftmargin = “100” topmargin=“70”> ....</body>
<body link="blue" alink ="red" vlink="navy">.....</body>
```

Codigos de colores

- El código del color en HTML va a estar formado por 6 símbolos alfa-numéricos precedidos del símbolo #. Estos símbolos pueden ser cualquiera de las diez cifras numéricas y de las seis primeras letras del alfabeto es decir: 0,1,2,3,4,5,6,7,8,9,a,b,c,d,e,f siendo 0 el valor mínimo (nulo) y f el valor máximo (15).

  A su vez esos seis símbolos alfanuméricos se dividen en tres grupos de dos símbolos:

  - El primer grupo indica la cantidad de **rojo** (00 es el mínimo y ff el máximo)
  - El segundo grupo indica la cantidad de **verde** (00 es el mínimo y ff el máximo)
  - El tercer grupo indica la cantidad de **azul** (00 es el mínimo y ff el máximo)

  Las mezclas y combinaciones de cantidad van dando origen a los diferentes colores.

ffffff	000000	ff0000	00ff00	0000ff	ffff00	cccccc	ff00ff
990000	99ffff	ff9900	ffffcc	66cc99	deb887	ffb6c1	2e8b57

- Este código también se puede sustituir por la palabra inglesa que define el color. Así por ejemplo el color rojo será red ó #ff0000 y el verde será green ó #00ff00.
- El problema viene cuando no es un color primario ni secundario. Por ejemplo, el color blanchedalmond se corresponde con el código hexadecimal #ffebcd

- Podemos ver los nombres y códigos en **tabla_colores.gif**

Etiquetas para MANEJO DE TEXTOS I

- Un porcentaje muy alto del contenido que aparece en las páginas web es texto por lo que es importante conocer las etiquetas de que dispone HTML tanto para estructurar ese contenido en secciones, párrafos, títulos como para poder señalar las palabras importantes
  - ….. se usa para estructurar el texto en párrafos e inserta una línea en blanco antes del siguiente texto. Puede llevar el atributo **align (left/center/rigth)** para alinear el texto y se puede omitir el cierre pero lo pondremos

    <p **align = "left"**> …….. alinea el texto a la derecha y el texto va a ocupar la anchura de la ventana de cada navegador
  -
 se usa como salto de línea (como si diéramos ENTER) y no tiene etiqueta de cierre, se permite usar también solo </br>. Puede llevar el atributo **clear (left /all/rigth/none)** que sirve para alinear de forma correcta el texto cuando existen imágenes flotantes
  - .. sirve para marcar partes del documento. Se usa sobre todo con CSS para señalar capas y tiene el atributo **align** que le permite alinear todos los párrafos que contiene a la vez
  - .. similar a div pero se usa para menor cantidad de texto
  - <hr> esta etiqueta no tiene cierre ni lleva texto asociado, nos permite añadir una línea horizontal 3D en la pantalla con sombra para separar un texto. Puede tener como atributos

    size: grosor de la línea en pixeles

    width: longitud de la línea en pixeles ó en porcentaje del ancho de la pantalla

    align : alineación como antes

    color : para dar otro color a la línea

    noshade para eliminar la sombra de la línea

    <hr **size =**"6" **align = "center" width = 70% color = "#aa00ff" noshade = "noshade"**>

Etiquetas para MANEJO DE TEXTOS II

- <hx>….. </hx> se llaman encabezados y se utilizan para poder poner títulos en el texto Lo que vaya entre esa etiqueta tendrá un cierto tamaño (la x varia desde el valor 1 que es el mayor tamaño hasta el 6 que es el menor) y además el texto sale resaltado como si usáramos negrita. Según el navegador varía su forma de presentación. Los encabezados pueden llevar el atributo **align** con sus cuatro posibles valores: **left, center, rigth, justify**

  <h1 align = "left" > Esto es una prueba </h1>

- …… se usa para especificar el posible formato del texto. Puede llevar como atributos:

  color: para dar otro color al texto (nombre en ingles o formato hexadecimal como antes)

  size: tamaño del texto de 1 hasta 7. También se puede poner +2 ó -2 y con ello indicamos dos veces mayor ó menor que el texto que tenemos

  face: sirve para definir la familia de la fuente ejemplo "Verdana"

   texto de prueba

- <basefont> presenta los mismos atributos que font pero no tiene cierre esta en desuso
- ….. para centrar algo
- <nobr>……..</nobr> para que el texto que hay entre ella no se corte al salir por pantalla

Etiquetas para MANEJO DE TEXTOS III

- Existen un grupo de etiquetas además de las de encabezados para utilizar con los textos (ó unos pocos caracteres) que se denominan etiquetas de **ESTILO FISICO** dado que sirven para cambiar el aspecto visual del texto

  - **** .....**** texto en negrita
  - **** .....**** texto en itálica
  - **<u>** .....**</u>** texto subrayado
  - **<s>** .....**</s>** texto tachado
  - **^{** ..... **}** para los superíndices
  - **_{** .....**}** para subíndices M Subíndice
  - **<strike>** .....**</strike>** testo también tachado
  - **<small>** .....**</small>** texto mas pequeño es acumulativo
  - **<big>** .....**</big>** texto mas grande es acumulativo
  - **<tt>** .....**</tt>** texto en formato maquina de escribir
  - **<pre>** ..... **</pre>** Cuando queremos que aparezca un texto tal y como lo hemos puesto
  - **<blink>..** **</blink>** si quiero que parpadee el texto

- Si queremos que aparezca un espacio en blanco adicional en alguna parte usaremos el código ** ** en el lugar que queramos

Etiquetas para MANEJO DE TEXTOS IV

- Existen otro grupo de etiquetas para utilizar con los textos que se denominan etiquetas de **ESTILO LÓGICO** ya que sirven para describir el uso del texto afectado definiendo un significado, contexto ó uso especifico

  - <**em**> .....</**em**> texto resaltado (por defecto en cursiva)
  - <**strong**> .....</**strong**> texto mas resaltado (por defecto en negrita)
  - <**del**> ....</**del**> texto afectado se ha eliminado en esta versión del documento (por defecto tachado)
  - <**ins**> .....</**ins**> texto afectado se ha añadido en esta versión del documento (por defecto subrayado)

    Tanto <**ins**> como <**del**> pueden llevar los atributos **cite = "URL"** que indica la dirección donde se puede encontrar información sobre la modificación **datetime = "fecha"** que indica la fecha en la que se hizo
  - <**abbr**> .....</**abbr**> texto es una abreviatura
  - <**acronym**> ..... </**acronym**> texto es un acrónimo
  - <**dfn**> .....</**dfn**> texto afectado es una definición de un término

    Tanto <**abbr**>,<**acronym**>,<**dfn**> pueden llevar el atributo **title= "nombre"** que nos va a indicar el significado completo de la abreviatura, el acrónimo ó la definición: Los navegadores suelen mostrar un borde punteado debajo del texto y al poner el ratón sobre el sale lo que es

Etiquetas para MANEJO DE TEXTOS V

- <**code**> ...texto </**code**> texto es un trozo de código de programación
- <**cite**> ... texto </**cite**> texto es una referencia a otras fuentes generalmente el autor
- <**address**> . texto .</**address**> texto es la información de contacto de una persona
- <**blockquote**> texto </**blockquote**> texto es una cita textual de un párrafo. El texto entre esa etiqueta aparece con márgenes Puede tener el atributo **cite= "url"** para indicar de donde se saco el texto
- <**q**> ... texto...</**q**> texto es una cita textual de una frase
- <**var**>... texto.. </**var**> el texto es una variable de u programa
- <**kbd**>... texto .</**kbd**> dice el texto que el usuario debe introducir en una aplicación informatica
- <**samp**> ... texto..</**samp**> texto con formato ejemplo
- <**marquee**>... texto .>/**marquee**> permite incluir un texto móvil, horizontal ó vertical . Algunos de sus atributos

  align alineacion del texto **(top |middle |bottom)**

  behavior como se mueve el texto **(scroll | slide | alternate)**

  bgcolor color fondo letrero

  direction direccion movimiento **(right | left |top)**

  loop numero de veces se produce el movimiento **(n | infinite)**

  width ancho del letrero

Símbolos especiales

- Algunos de los caracteres que se utilizan habitualmente en los textos no se pueden incluir directamente en las páginas web.
- Por ejemplo los caracteres que utiliza HTML para definir sus etiquetas (<, > y ") no se pueden utilizar libremente. Esto se resuelve usando ciertas entidades HTML que representan a cada uno de esos carácteres como son:

  < (<) **≶** (>) **&** (&) **"** (“) **'** (‘) ** ** (espacio en blanco)

- Los caracteres propios de los idiomas que no son el inglés (ñ, á, ç, ¿, ¡, etc.) pueden ser problemáticos dependiendo de la codificación de caracteres utilizada. Esto ocurre por ejemplo cuando el editor HTML del diseñador, el entorno de desarrollo del programador, el servidor web que sirve las páginas y el navegador del usuario no usan todos el formato de caracteres UTF-8 y entonces si se pasa a codificación ISO-8859 en cualquier momento sin realizar una conversión correcta, el navegador del usuario mostrará caracteres extraños en todos los acentos y en todas las letras como la ñ.

  Para solventar ese problema se sustituye el carácter problemático por su entidad HTML como por ejemplo:

  ñ (ñ) **Ñ** (Ñ) **á** (á) **é** (é) **Á** (Á) **€** (€)

Etiquetas para MANEJO DE SONIDOS

- Si queremos introducir un sonido de fondo en una página web disponemos de una etiqueta que hace que los sonidos se ejecuten de forma automática. Es una etiqueta propietaria de Microsoft por lo que solo la interpreta IE. Admite sonidos MID y WAW aunque suele aceptar AU y MP3
  - <**bgsound**> .....</**bgsound**>no precisa cierre y tiene dos posibles atributos:

    src que nos sirve para indicar la URL donde esta el fichero de sonido

    loop sirve para indicar el numero de veces que queremos se repita el sonido si queremos que se repita siempre pondremos **infinite**

    balance indica el balance de sonido entre los dos altavoces (0 es equilibrado)

    volume determina el volumen **disabled, delay, id, class etc**

    En un fichero insertado así prefijamos valores pero no tenemos control sobre el

- Si queremos introducir un sonido de fondo en una página web y poder controlarlo usaremos otra etiqueta distinta
  - <**embed**> .....</**embed**> que hace que aparezcan en pantalla una especie de consola de mando con tres botones (play, pause y stop) para manejar el sonido. Algunos atributos son:

    src para indicar donde esta el fichero

    loop para el número de veces **(nº | true** infinitas **|false** solo una**)**

    type tipo de fichero para poder escoger el navegador un plugin adecuado

    autostart para que comience de forma automática **(true)** ó que lo haga el usuario **(false)**

    heigth y width para indicar el ancho y alto del panel de los controles

    hiddem para ocultar **(true)** ó mostrar **(false)** la consola del control

- También podemos usar la etiqueta <**a**> y la etiqueta <**object**>

LISTAS I

- Existe una forma de organizar la información que a veces puede tener mas sentido que escribirla de forma separada para ello en html se usan las listas que pueden ser de tres tipos: **desordenadas, ordenadas y de definición**

A) LISTAS DESORDENADAS

Son las más sencillas y las que más se usan. Están formadas por un conjunto de elementos relacionados entre si pero en los que no se indica un orden determinado. Cada elemento va precedido de un símbolo llamado "viñeta" ó "bullet".

- La etiqueta <ul> ..... sirve para encerrar al conjunto de elementos que forman la lista. Por defecto el navegador muestra los elementos de la lista tabulados y con una pequeña viñeta que es un circulo negro.

  Esta etiqueta puede tener los atributos:

  type (disc/circle/square) que sirve para cambiar el tipo de viñeta que aparece antes de todos los elementos de la lista.

  compact que hace que se muestre la lista lo más compacta posible.

- Para declarar cada elemento de la lista se usa la etiqueta <li>.... y en ella se puede omitir el cierre. Puede tener también el atributo type para un elemento concreto de la lista

  Ejemplo de lista:

```
<ul type="square">
        <li>Inicio </li>
        <li>Ultimas noticias </li>
        <li>Articulos </li>
</ul>
```

LISTAS II

B) LISTAS ORDENADAS

Están formadas por un conjunto de elementos relacionados en las que cada elemento va precedido de un símbolo de numeración (numero ó letra) que indica el orden del mismo.

- La etiqueta <ol> ..... sirve para encerrar al conjunto de elementos que forman la lista. Por defecto el navegador muestra los elementos de la lista ordenados con numeración común y comienzan con 1.

  La etiqueta <ol> puede llevar varios atributos

  type (1/I/i/A/a) que sirve para cambiar el tipo de viñeta que aparece antes de todos los elementos de la lista (numérica, romanos mayúscula, romanos minúscula, alfabético mayúscula, alfabético minúscula).

  compact que hace que se muestre la lista lo más compacta posible.

  start sirve para definir el valor de comienzo de la numeración de la lista

- Para declarar cada elemento de la lista se usa la etiqueta <li>..... En esta se puede omitir el cierre . Puede tener como atributos:

  type para un elemento en concreto de la lista

  value para definir el número por el que se continua la numeración de la lista

  Ejemplo:

```
<ol type="I" value= "X">
        <li>Inicio </li>
        <li>Ultimas noticias </li>
        <li>Articulos </li>
</ol>
```

LISTAS III

C) LISTAS DESCRIPTIVAS

Su comportamiento es similar a un diccionario ya que están formadas por una lista de términos emparejados con su definición. Se usan poco.

- La etiqueta <dl> .....</dl> sirve para encerrar al conjunto de elementos que forman la lista.
- La etiqueta <dt> …. </dt> encierra el término
- La etiqueta <dd> ….</dd> encierra la definición

Ejemplo:

```
<dl >
        <dt>SQL </dt>
            <dd>  Es un lenguaje ….        </dd>
        <dt> Google </dt>
            <dd>  Es un navegador…        </dd>
</dl>
```

D) LISTAS ANIDADAS

Se pueden anidar todos los tipos de lista unos dentro de otros solo tenemos que tener cuidado con lo que queremos hacer y los índices que usamos

IMÁGENES I

- Las imágenes son importantes en una página web ya que en general las páginas contienen multitud de imágenes. Existen imágenes de dos tipos: **de contenido** (complementan la información del texto) y **de adorno** (sirven para mostrar fondos de página, bordes redondeados, pequeños dibujos etc). Las de contenido se insertan con la etiqueta <img> y las de adorno se deberían insertar a través de las hojas de estilo CSS
  - **<img >** es la etiqueta utilizada para insertar imágenes. No tiene cierre y tiene un atributo obligatorio **src** (source) . Además de este puede llevar los siguientes atributos:

    src: especifica el nombre del fichero gráfico que contiene la imagen ó la ruta donde está es decir su URL. Los estándares son los formatos **gif y jpg**, aunque las últimas versiones ya admiten también el formato **png**

    title: recoge un texto para ponerle título

    border: sirve para indicar el borde de la imagen por defecto vale cero

    alt: es una descripción corta de la imagen y se verá cuando un navegador no puede mostrar una imagen ó al ponernos con el ratón sobre ella

    longdesc: no es habitual pero contiene la dirección donde se puede ver información sobre la imagen

    width: indica el ancho en píxeles con el que se muestra la imagen. Puede indicarse con un número (pixeles) ó con un porcentaje (20%) bien con respecto al elemento en el que esta ó a bien a la página si la imagen no está contenida en un elemento.

    height: indica el alto en píxeles con el que se muestra la imagen. Se indica como el ancho

- Ejemplo:

  <img src="/imagenes/foto1.jpg" alt="Fotografía de un paisaje" width="30%“ height="350">

IMÁGENES II

- Existe un atributo que sirve para crear lo que se conoce como imágenes flotantes

  align: sirve para indicar como se alinea una imagen respecto al texto que la acompaña. Puede ser **top** (coloca el punto más alto de la imagen coincidiendo con el alto de la línea de texto actual), **middle** (alinea el punto medio en altura de la imagen con la base del texto), **bottom** (valor por defecto, alinea el punto más bajo de la imagen con la base del texto), **left** (pone la imagen a la izquierda del texto) y **right** (coloca la imagen a la derecha del texto). Normalmente se emplean los dos últimos
- Una imagen flotante afecta a todos los elementos que siguen a continuación a no ser que pongamos un salto de línea especial. Recordemos que la etiqueta <**br**> tenía un atributo **clear** que sirve para indicar que el texto no debe empezar hasta que el margen especificado este libre (es decir hasta el final de la imagen)
- Ejemplo:

  <**img** **src="/imagenes/foto1.jpg“ align = “left”**>

  <**p**>.....</**p**>

  <**br clear= “all”**/>

  Cuando se usa **align** no se deja ningún espacio en blanco entre las imágenes y los elementos que siguen. En estos casos se puede usar el atributo **vspace= “x”** y **hspace = “x”** siendo x el numero de pixeles de espacio a añadir a la derecha e izquierda de la imagen ó bien arriba y abajo

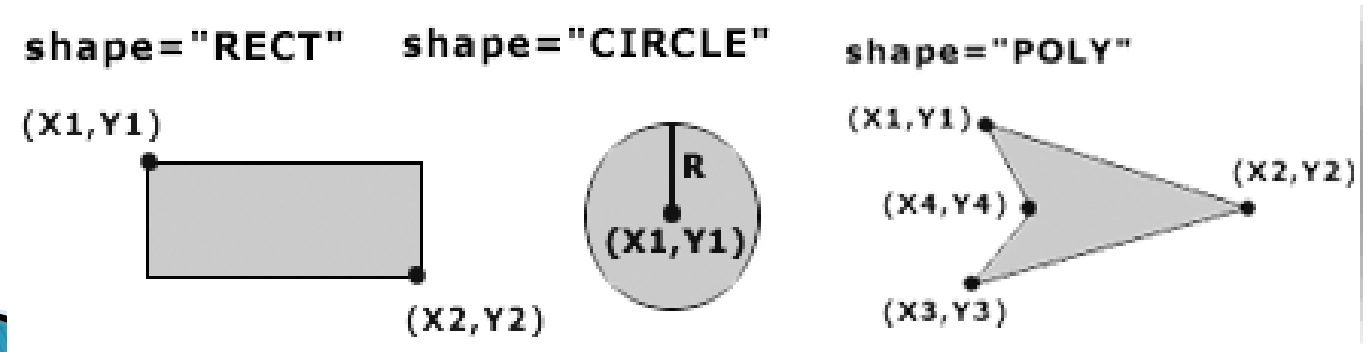
IMÁGENES III

- Hoy en día el uso de los **mapas de imágenes** se ha reducido debido al avance de otras tecnologías (Flash etc) pero en HTML existe la posibilidad de definir diferentes zonas dentro de una imagen de tal forma que al pinchar sobre ellas nos llevan a sitios diferentes. Por ejemplo un mapa de un país con distintas provincias
- Las zonas que se pueden definir en una imagen se crean mediante rectángulos, círculos y polígonos
- Para poder crear un mapa de imagen en html se usan tres etiquetas
  - <**img**> sirve para insertar la imagen y usaremos un atributo de esa etiqueta que es **usemap** donde pondremos con un # el **"#nombre"** del mapa que estará declarado en la etiqueta map
  - <**map**> para poder definir las zonas. Tiene un atributo **name** para poner el nombre que identifica a ese mapa
  - <**area**> para cada una de las zonas. Puede tener los atributos:

    href para indicar la URL a la que se accede al pinchar sobre el área

    shape para decir el tipo de área que se define (**default | rect | circle | poly**)

    coords para indicar las coordenadas del área separadas por comas ej: (X1, Y1, X2,Y2). En el caso de polígonos si las dos últimas coordenadas no son iguales a las primeras se cierra el polígono de forma automática



OBJETOS

- Además de imágenes HTML permite incluir en las páginas otros elementos más complejos como applets de Java y videos en formato QuickTime ó Flash. Estos contenidos no los suele interpretar de forma directa el navegador sino que utiliza pequeños programas (plugins) para hacerlo
- Para poder incluir esos objetos en una página se usa la etiqueta
  - <object> Puede tener los siguientes atributos:

    data para indicar la URL de los datos que usa el objeto

    classid, codebase, codetype para decir información dependiendo del tipo de objeto

    type indica el tipo de contenido de los datos. Los valores están estandarizados

    height y width indican la altura y anchura con la que se muestra el objeto

```
<object data = "PlanetaTierra.mpeg" type= "application/mpeg"/>

<object title = "La Tierra desde el espacio" classid=
                                    http://www.observer.mars/TheEarth.py33>
        <!--Otros formatos alternativos: -- >
        <object data = "PlanetaTierra.mpeg" type= "application/mpeg">
           <object data = "PlanetaTierra.gif" type= "image/gif">
           </object>
       </object>
</object>
```

ENLACES I

- Una de las claves del éxito de HTML fue la incorporación del hipertexto, ya que permitió crear documentos interactivos que nos dan información adicional cuando se solicita. El elemento principal del hipertexto es el "hiperenlace", llamado tambien "enlaceweb“ o simplemente "enlace".
- Los enlaces se usan para establecer relaciones entre dos recursos. Aunque la mayoría de los enlaces relacionan páginas web, también es posible enlazar otros recursos como imágenes, documentos y archivos.
- Algo importante de los enlaces es que están formados por dos extremos y un sentido. Es decir, el enlace comienza en un recurso y apunta hacia otro recurso. Cada uno de los dos extremos se llaman "anchors“ en inglés, que se puede traducir literalmente como "anclas"
- Veamos que es una URL (Uniform Resource Locator) se refiere al identificador único de cada recurso disponible en internet y esas URL son esenciales para crear enlaces pero también se usan en imágenes y en formularios. Veamos un ejemplo y que lo forma

  Supongamos una URL completa ➔ http://www.todo sobreweb.es/internet/capitulo4.web

  protocolo (**http://**) es el mecanismo que debe usar el navegador para acceder a ese recurso. Todas las páginas web usan ese recurso. Las páginas seguras usan **https//** (bancos,email)

  servidor (**www.todossobreweb.es**) de forma sencilla es como indicar el ordenador en el que esta guardada la página a la que se quiere acceder. Los navegadores pueden obtener la dirección de un servidor a partir de su nombre

  ruta (**/internet/capítulo4.web**) es el camino que se debe seguir una vez hemos llegado al servidor para localizar el recurso especifico al que se quiere acceder
- **La URL de una página** es imprescindible para crear los enlaces ya que nos permite distinguir una página de otra y proporciona a los navegadores la información necesaria para poder ellegar a ese recurso

ENLACES II

- Un link ó hiperenlace consiste en hacer que un **texto ó imágen** de los que aparecen en una página web nos permitan, al pincharlos, llevar al usuario a otra parte de la página ó a otra página distinta etc. Para poder ponerlos se usa la siguiente etiqueta:

- <a> ..... Pero no es suficiente solo con poner un texto en esta etiqueta ya que además necesitamos decirle donde queremos que vaya para lo cual necesitamos el atributo **href** en el cual indicaremos la URL del recurso al que queremos enlazar

   enlazo a la página de google

  Esto indica que la frase dentro de la etiqueta <a> va a ser un enlace a google

  y esa frase aparecerá en la página subrayada y con otro color

  Dado que ponemos la URL completa se llama **referencia absoluta**

- Pero normalmente no queremos ir a google sino a otra página nuestra entonces vamos a probar a decirle como referencia el lugar donde esta esa página nuestra que ya hemos hecho. Esto se llama **referencia relativa** ya que solo hace falta indicar en la URL la ruta de acceso al recurso y el nombre ya que el protocolo del sitio donde queremos ir es el mismo que el del lugar en que estamos)
- Cuando las dos páginas están en el mismo sitio ó carpeta basta con poner en el atributo href el nombre de la página a la que quiero ir entre comillas

   Link a otra pagina

  Podemos hacer que el link no sea toda la frase sino solo una palabra

   Link a otra pagina

  Y podemos hacer que a su vez en la pagina a la que vamos aparezca un enlace que nos devuelva de nuevo a la página de partida para lo cual en la segunda página añadiremos una línea como la siguiente

   Vuelve a primera pagina
- Puede ocurrir que nuestra página este en otra carpeta ya sea subcarpeta ó hacia atrás y entonces

   Va a una subcarpeta

   Vuelve hacia atrás un nivel

EJEMPLOS de lo que hace el navegador

Si tenemos el origen del enlace por ejemplo en

http:// www.ejemplo.com/ruta1/ruta2/ruta3/pagina1.html

Si tenemos como posibles Destinos

1) **una pagina en el mismo directorio** **pagina2.html**
2) **una página en nivel superior** **../pagina2.html**
3) **una página en dos niveles superior** **../../pagina2.html**
4) **una pagina en otra tuta del mismo sitio (absoluta)** **http://www.ejemplo.com/ruta4/pagina2.html**

   ../../../ruta4/pagina2.html

   (relativa)
5) **una página en un nivel inferior** **ruta5/pagina2.html**

Si la URL relativa…	El navegador la transforma en URL absoluta
.. Solo es el nombre de un recurso	… añadiendo el protocolo, el servidor y la ruta completa del origen del enlace
.. comienza por ../	..añadiendo el protocolo y el servidor del origen del enlace, subiendo un nivle en la jerarquia de directorios y añadiendo el resto de la ruta incluida en la URL relativa
..comienza por /	..añadiendo el protocolo y servidor del origen del enlace
En cualquier otro caso	.. Añadiendo el protocolo, servidor y ruta completa del origen del enlace a la que se añade la ruta incluida en la URL relativa

ENLACES III

- Una imagen también puede ser un hiperenlace si la metemos dentro de la etiqueta <a>

   <img src = "imagen.gif">

- **TIPOS DE ENLACES en función del destino del enlace**

  en función del destino los enlaces se agrupan generalmente del siguiente modo:

  1.**Enlaces locales**: los que se dirigen a otras páginas del mismo sitio web (subcarpetas tendremos que declarar la ruta y a su vez pueden tener anclas) (ya vistos)

  2.**Enlaces remotos ó externos:** los dirigidos hacia páginas de otros sitios web (por ejemplo a google) (ya vistos)

  NOTA: Todas las direcciones web (URLs) empiezan por **http**://. Esto indica que el protocolo por el que se accede es HTTP, el utilizado en la web. Pero para enlazar con una página no estamos obligados a usar ese protocolo ya que también podemos acceder a recursos a través de otros protocolos como por ejemplo el FTP. En ese caso, las direcciones de los recursos empezarán con **ftp**://.

  3.**Enlaces internos:** los que se dirigen a otras partes dentro de la misma página (se llaman ANCLAS).

  4.**Enlaces con direcciones de correo**: para crear un mensaje de correo dirigido a una dirección (mailto).

  5.**Enlaces con archivos**: para que los usuarios puedan hacer download de ficheros

ENLACES IV

- **3.ENLACES INTERNOS** (PUNTO DE FIJACION ó ANCLA)

  Cuando una página es grande y esta dividida en varios apartados, podemos ir directamente al apartado que queramos para lo cual necesitamos usar los puntos de fijación "anclas" que son muy útiles a la hora de crear índices. Para poner un ancla se usa la etiqueta <a> pero en este caso utilizando el atributo:

  name que sirve para indicar el lugar de destino del enlace. Ese nombre lo vamos a inventar, va a ir entre comillas y no puede tener espacios en blanco ni caracteres especiales (investigar **id**)

  Por ejemplo: si tengo una página **ejemplo.html** y desde el texto de esa página que dice **ver clasificación** quiero ir a otro sitio de la página donde esta esa clasificación tendremos que poner en el lugar de destino un ancla con un nombre pero justo en el punto al que queremos ir. A ese nombre (ancla) nos referiremos en la URL del sitio de partida es decir:

  <p> Los animales son seres vivos, ver clasificaciön , que nacen crecen se reproducen y mueren .....

  (en el lugar de la página donde este la clasificación pondremos esta línea para el ancla)

   Clasificación de los animales.....

  Puede ocurrir que el lugar donde queremos ir este en otra página. En ese caso pondremos la línea del ancla en la otra página, justo en el lugar al que queremos ir, y en nuestra página en el lugar de partida escribiremos la siguiente línea

  <p> Los animales son seres vivos, ver clasificacion , que nacen crecen se reproducen y mueren ...

  target es otro atributo de la etiqueta <a> que nos permite abrir un vínculo en una nueva ventana con lo que el usuario va a tener acceso a la página que llama y a la pagina llamada.

ENLACES V:

El atributo target puede tener distintos valores que hacen que el documento referenciado se muestre en:

_blank en una nueva ventana
_self en la misma ventana (por defecto)
_parent en la ventana padre de la actual
_top usando todo el espacio de la ventana

- **4. ENLACES A DIRECCIONES DE CORREO**

AL pincharlos se abre un nuevo mensaje de correo electrónico dirigido a una dirección de email determinada. Son habituales en las páginas web y sirven para poder contactar con el propietario de la página. Para construirlos ponemos en href **mailto:dirección de correo** a la que se debe dirigir el enlace.

 contenido

(en contendido se suele poner el correo que figura delante por si la persona que consulta la página no tiene un correo configurado en su equipo

Además también podemos poner el asunto del mensaje escribiendo tras la dirección **un interrogante, la palabra subject, un signo igual (=) y el asunto** en concreto es decir

 direccion

- **5. ENLACES A FICHEROS**

Si quiero enlazar con un archivo fichero.zip que esta en el mismo directorio que mi página pondríamos

 Descarga fichero.zip

Al pinchar en este enlace nuestro navegador descargará el fichero, preguntándonos si queremos "abrir o guardar ese fichero en disco".

Si enlazo a otro tipo de archivo (PDF o VRML (Realidad virtual para Internet) es igual. Si el navegador reconoce el archivo como. PDF pondrá el programa Acrobat Reader a funcionar para mostrarlo ó si enlazo con un mundo VRML pondrá en marcha el plugin que el usuario tenga para ver los mundos virtuales (como por ejemplo Cosmo Player)

 Descarga el PDF

TABLAS I

- Las tablas permiten presentar la información en filas y columnas. Las tablas pueden contener elementos simples, agrupaciones de filas y columnas, cabeceras y pies de tabla, subdivisiones cabeceras múltiples y otros elementos más complejos
- Se utilizan sobre todo para dar información tabular y hasta hace unos años se usaban también para definir la estructura de las página web y eso no es correcto ya que para eso existen las hojas de estilo CSS
- Las tablas más sencillas se definen utilizando tres etiquetas:
  - **<table>**...**</table>** para crear la tabla
  - **<tr>**.....**</tr>** para definir cada una de las filas (table row)
  - **<td>**.....**</td>** para declarar una celda con contenido (table date) Texto a la izquierda
  - **<th>**..... **</th>** para declarar una celda cabecera (table head) Texto centrado y negrita
- Las etiquetas de cierre de **tr, th y td** pueden omitirse pero se suelen poner al principio
- En las celdas de una tabla se puede poner cualquier cosa texto, imágenes, videos, enlaces, otras tablas, formularios etc. Por defecto las tablas salen sin borde ni líneas

```
<table>
    <tr>  <td>uno-1</td>
          <td>uno-2</td> </tr>
    <tr>  <td>dos-1</td>
          <td>dos-2</td> </tr>
</table>
```

TABLAS II

- La etiqueta <**table**> puede tener muchos atributos
  border tamaño del borde exterior de la tabla. Si no queremos borde se pone 0
  align alineación de la tabla respecto a la página (center, right, left)
  width ancho de la tabla en pixeles ó en porcentaje %
  height alto de la tabla en pixeles ó en porcentaje%
  background imagen de fondo de la tabla
  bgcolor color de fondo de la tabla
  bordercolor color de borde de la tabla
  ESPACIOS ENTRE CELDAS ó CELDAS y TEXTO
  cellspacing espacio entre celdas
  cellpadding espacio entre la celda y su contenido

- <**caption**> … </**caption**> es una etiqueta que se puede poner justo detrás de la etiqueta <**table**> y va a permitir poner el título de la tabla que además puede estar formateado con otras etiquetas de HTML y puede tener el atributo **aling (top/ bottom. (IE: left, rigth, center)**

TABLAS III

- La etiqueta <**tr**> de fila puede tener distintos atributos
  - **align** alineación horizontal de las celdas (center, right, left)
  - **valign** alineación vertical (top, middle, bottom)
  - **bgcolor** color de fondo
  - **bordercolor** color de borde

TABLAS IV

- Las etiquetas <td> y <th> que se refieren a celda pueden tener muchos atributos y su cierre es opcional aunque se suele poner al principio para que sea legible.

  - **align** alineación horizontal (center, right, left)
  - **valign** alineación vertical del contenido (top, middle, bottom)
  - **bgcolor** color de fondo
  - **bordercolor** color de borde
  - **background** imagen de fondo
  - **wigth** ancho en pixeles ó en %
  - **heigth** altura en pixeles ó en %
  - **nowrap** no toma ningun valor sale es texto en una línea

  OTROS PARA UNIR FILAS O COLUMNAS

  - **colspan** número de columnas unidas
  - **rowspan** número de filas unidas
  - En los dos casos extendemos una fila ó una columna para que ocupen el tamaño de varias

TABLAS V

- Sabemos que existe el atributo border para asignar un borde a una tabla pero desde la versión de html 4 se aceptan otros atributos para bordes.

  Se va a hacer distinción entre

  A) Bordes externos de tablas (pero no los de celdas)

  Para ellos se puede usar el atributo **frame** dentro de la etiqueta <**table** **frame** = " " >

  Este atributo puede tomar los siguientes valores :

  void: Ningún borde externo

  above: Sólo aparece el borde superior

  below: Sólo el borde inferior de la tabla

  hsides:Se mostrará sólo el borde de arriba y abajo.

  vsides: Sólo los bordes de los lados izquierdo y derecho

  lhs: Se mostrará tan solo el borde izquierdo

  rhs: Se mostrará solamente el borde derecho

  box: Los 4 bordes

  border: También se mostrarán los 4 bordes

TABLAS VI

B) Bordes internos de tablas (solo los de las celdas)

Para ellos se usa el atributo **rules** dentro de la etiqueta <**table** **rules** = " " >

Este atributo puede tomar los siguientes valores:

none: No se coloca ningún borde interno

groups: Sólo se mostrarán bordes en los grupos de columnas o filas. (veremos más adelante como agrupar columnas o filas

rows: Sólo aparecerán bordes entre filas.

cols: Los bordes internos sólo aparecerán para separar columnas

all: Se mostrarán todos los bordes internos

TABLAS VII

TABLAS ANIDADAS

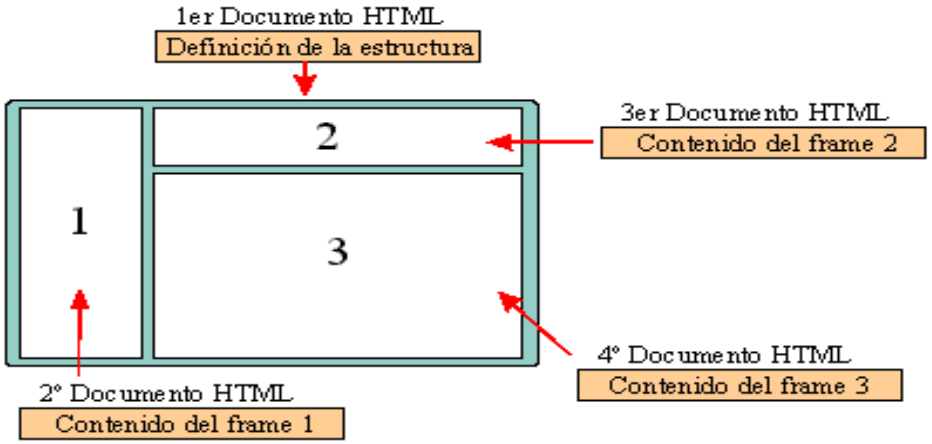
- Las tablas se pueden anidar unas con otras, es decir, se puede colocar una tabla o tablas dentro de otra, para conseguir un efecto complejo de maquetación. Para poder hacerlo deberemos introducir una tabla con toda su estructura dentro de una celda de la tabla principal. Deberemos tener cuidado y por ejemplo tener especial precaución en fijar los valores de los atributos **border, cellspacing y cellpading** de la tabla interior a cero, para no producir efectos indeseados.
- También debemos saber que si usamos una única tabla tendremos mejor control sobre el contenido de la misma, mayor velocidad de carga de la pagina, etc. Este factor de la velocidad de carga se debe a que el navegador antes de mostrar una tabla debe leer e interpretar todo su contenido por lo que en el caso de tablas anidadas se produce un efecto acumulativo que retarda de forma progresiva la carga y posterior presentación de la tabla completa.

TABLAS COMPLEJAS

- Existen tablas complejas en las cuales las filas pueden agruparse en: una sección de cabecera, otra de pie de tabla y una ó mas secciones de cuerpo de tabla para lo cual html dispone de las etiquetas **<thead> <tfood> y <tbody>** respectivamente
- Cada uno de esas etiquetas pueden tener atributos que servirán para asignar propiedades a todas las celdas que forman el grupo
- Cada tabla puede tener una sola cabecera y un pie y si se definen deben ponerse siempre antes que cualquier etiqueta <tbody>
- Buscar ejemplos e información en internet **<col> y <colgroups>**

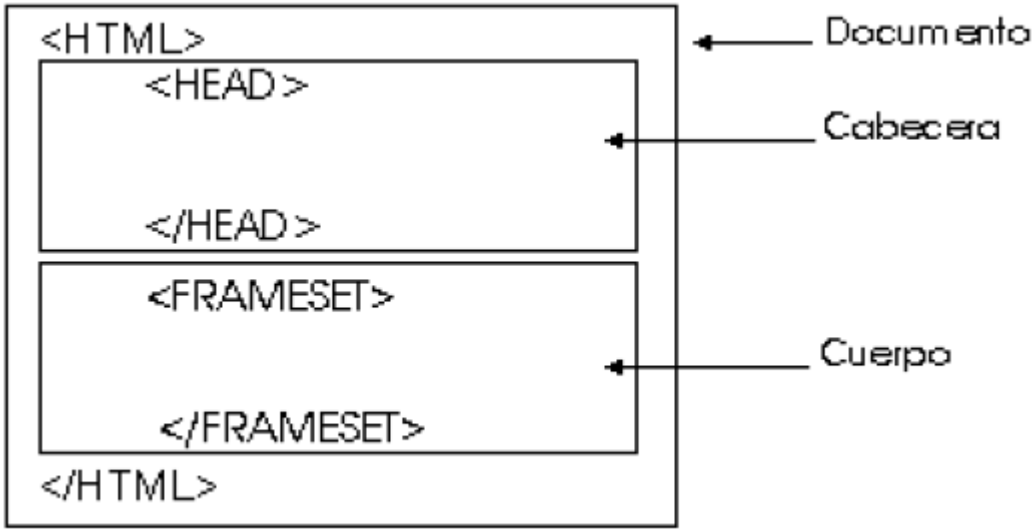
MARCOS:FRAMES I

- Los frames consisten en una técnica para insertar varios documentos HTML en una sola ventana del navegador. Realmente nos van a permitir dividir la ventana del navegador en zonas cada una de las cuales puede tener sus propios bordes, barras de desplazamiento etc y en las que podemos cargar documentos (páginas) independientes.
- Esas zonas o secciones que reciben el nombre de *frames* o *marcos*. En cada uno de estos marcos debemos insertar un contenido, que en general constará de un documento HTML, aunque también puede ser una imagen o algún otro elemento delWWWvisualizable por los navegadores
- Algo importante es que por ejemplo pulsando un enlace que este en un frame se puede cargar en otro de los frames una página determinada. Esto esta bien por ejemplo para tener en una página un frame estrecho con un índice de contenido a la izquierda y al ir pulsando en las diferentes opciones que hay sale en la ventana principal la página que he escogido



MARCOS:FRAMES II

- En el dibujo anterior serán necesarios 4 documentos HTML. El primero contendrá el código que indica al navegador el número de frames de la página, su tamaño, su posición y el nombre de los archivos donde estará el contenido de los 3 frames, a este documento HTML le llamaremos *documento de definición de frames* .
- Cada uno de los otros 3 documentos HTML tendrá el contenido de cada uno de los tres frames y se vana a crear de la misma forma que los documentos HTML que sabemos hacer pero hay que tener en cuenta que el contenido de esos documentos (páginas) será mostrado en un espacio reducido (el del frame correspondiente) y no en toda la ventana del navegador
- Las paginas que contienen las estructuras de frames **no tienen la etiqueta <body>** y las etiquetas que sirven para definir los marcos pueden dividir a la ventana del navegador de forma horizontal ó vertical e incluso estas divisiones pueden anidarse.
- Estructura básica del documento d definición de frames



MARCOS:FRAMES III

- La etiqueta <**frameset**> es la que va a indicar al navegador que existen ventanas "marcos" y además cuanto ocupa cada uno de ellos entre comas.
- Sus atributos más habituales:

cols si la división de la ventana principal es vertical (columnas)

rows si la división es horizontal (filas)

El tamaño es decir el valor de esos dos atributos puede estar expresado como:

- número entero el tamaño se indica de forma absoluta en pixeles
- número entero seguido de % indica un porcentaje del espacio disponible
- número entero seguido de * el tamaño se indica de forma relativa al espacio sobrante

Ejemplos:

<**frameset rows= "30% ,*"**> .....</**frameset**> dos frames horizontales

<**frameset cols= "100, 200 , * "**> .....</**frameset**> 3 frames verticales

Esa etiqueta puede también tener otros atributos; frameborder, framecolor ó framespacing

Se pueden hacer muchas combinaciones para especificar los tamaños de los frames e incluso usar en la misma etiqueta combinación de filas y columnas

<**frameset cols="100,50,*,* " rows="15%,*,3*"**>

- Además la etiqueta <frameset> va a contener dentro de ella una declaración por cada una de las divisiones que se han hecho de la pantalla y para ello usa la etiqueta <**frame**>

<**frame**> va a indicar al navegador las características de cada ventana y no tiene etiqueta de cierre.

Sus posibles atributos son:

src indica donde esta el documento (página) que va a cargarse en esa ventana

name asigna un nombre a la ventana ó al frame para poder referirnos a el con el atributo target

MARCOS:FRAMES IV

noresize es un atributo boleando si se pone el marco no será redimensionable por el usuario scrolling controla el uso de barras de desplazamiento en el marco (auto | yes |no) **por defecto es auto y sale cuando hay un trozo de la página que no cabe**

frameborder si quiero tener borde(1) ó no (0), en el caso de que queramos **quitarlo a todos se pondrá en frameset**

framecolor indica el color del marco

marginheigth y marginwidth indica el espacio entre el contenido y los límites horizontales y verticales respectivamente del marco

Un ejemplo sencillo que divide a la pagina en dos frames verticales uno que ocupa 30 pixeles y el otro el resto de la página

```
<frameset      cols= "30,*">
        <frame  name ="primero"  src = "uno.html">
        <frame  name ="segundo" src = "dos.html">
</frameset>
```

- Existe otra etiqueta que se (puede/**debe**) introducir dentro de la etiqueta <frameset> para aquellos casos en los que el documento se use en navegadores que no soporten frames ó bien que tengan los frames desactivados. Esta etiqueta se ignorara en los demás casos

```
<noframes>
        <body>
        <p> Esta pagina usa marcos pero su navegador no los admite </p>
        </body>
</noframes>
```

MARCOS:FRAMES V

- Cuando pulsamos en un enlace situado en un marco el documento al que llamamos se carga en el mismo marco pero podemos hacer que se cargue en otro marco distinto poniendo el nombre del marco de destino en el atributo target

 enlace

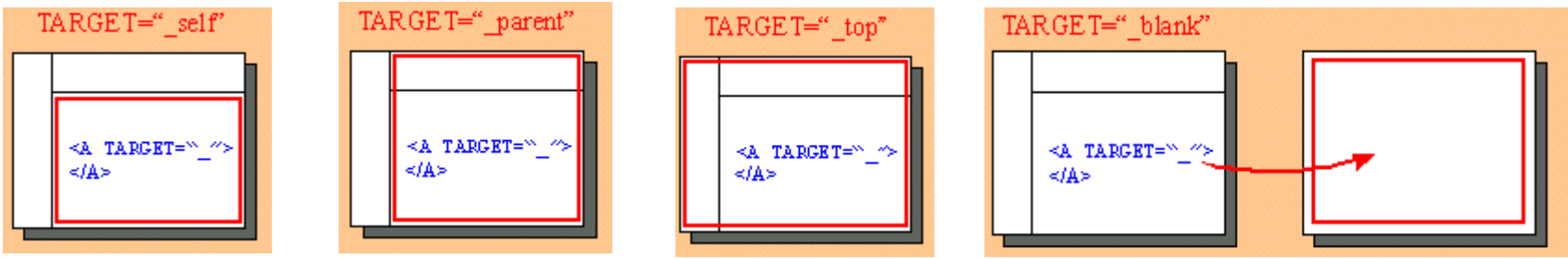
El atributo target puede tener los valores:

_blank el navegador carga el recurso del destino en una nueva ventana

_self el navegador carga el recurso del destino en la misma ventana

_parent el navegador carga el recurso del destino en el marco padre del actual

_top el navegador carga el recurso del destino en la ventana completa



Los cuadrados rojos indican la zona o ventana en la que se mostrara el documento enlzado

MARCOS:FRAMES VI

- Los marcos también se pueden anidar:

```
<frameset rows="20%,*">
        <frame src="marco1.html" name="navegacion">
    <frameset cols="50%,*">
        <frame src="marco2.html" name="info">
        <frame src="marco3.html" name="detalles">
    </frameset>
</frameset>
```

A través del código anterior se crea una página con tres marcos. El primero ocupa la franja superior de la ventana. Los dos siguientes dividen la franja restante en dos columnas del mismo tamaño.

MARCOS FLOTANTES:FRAMES VII

- Los frames flotantes son muy parecidos a los que hemos visto, siendo su principal diferencia que pueden ser insertados en cualquier lugar de una página web, y su función ya no es dividir la ventana del navegador en varias partes, sino insertar en medio de un documento HTML el contenido de otro en un área del tamaño que indiquemos. Es como un agujero que aparece dentro de una página a través del cual se ve otra página. Se suele usar para mostrar bien contenidos externos a la página ó mostrar una aplicación común a varios sitios web de la misma empresa.

Para crearlos se usa la etiqueta <**iframe**>……..</**iframe**> que puede aparecer en cualquier sitio de una pagina y en una pagina pueden aparecer tantos como queramos

Esta etiqueta puede tener los atributos:

src es obligatorio y sirve para indicar donde esta la página que se quiere insertar

name nombre del iframe

align alineacion de la ventana (**left |center |right | top | bottom |middle**)

heigth y **width** dicen el ancho y alto de la ventana

longdesc direccion donde puede haber una descripcion larga del contenido del iframe

scrolling indica si el iframe muestra barras de scroll (horizontal y vertical) si el contendido no cabe en la ventana (**yes |no |auto**)

```
<iframe  src="contenido.html"  width="40%" height="60%"
 align ="center" scrolling="auto“ >
```