1.- TEMA 1: HTML

Módulo Lenguajes de Marcas y Sistemas de Gestión de Información

1.1.- Qué es un lenguaje de marcas

Un **lenguaje de marcado** o **lenguaje de marcas** es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del texto o su presentación. (Wikipedia)

No es un lenguaje de programación, por lo que no tiene instrucciones para alterar el flujo, ni puede ejecutar sentencias.

Tienen su origen en el mundo editorial

1.2.- Historia de HTML

- En los años sesenta IBM crea ***GML*** (Generalized Markup Language).
- En los años ochenta aparece ***SGML*** (Standard Generalized Markup Language) que era dificil y requería de herramientas caras
- HTML nace en 1990 en el ***CERN***. Es una simplificación de SGML. ***HTML 2.0*** fue la primera versión oficial de HTML el ***IETF*** (The Internet Engineering Task Force) publicó el estándar en septiembre de 1995.
- ***HTML 3.2*** se publicó el 14 de Enero de 1997 por el **W3C**.
- ***HTML 4.0*** se publicó el 24 de Abril de 1998. Entre las novedades que presenta se encuentran las hojas de estilos CSS y la posibilidad de incluir pequeños programas en las páginas web.
- ***HTML 4.01*** es la última especificación oficial de HTML se publicó el 24 de diciembre de 1999. Es una actualización de la versión anterior. En ese momento el W3C detuvo la actividad de estandarización de HTML
- A partir de este momento se crea lenguaje ***XHTML*** que combina la sintaxis de HTML 4.0 con la de XML.
- ***XHTML 1.0*** fue la primera versión, se publicó el 26 de Enero de 2000. Es una adaptación de HTML 4.01 al lenguaje XML, por lo que mantiene sus características, y añade algunas restricciones y elementos de XML.
- En 2004, Apple, Mozilla y Opera se unen para crear el grupo ***WHATWG*** (Web Hypertext Application Technology Working Group) para continuar desarrollando HTML. En 2006 W3C se une para desarrollar HTML 5.0
- A finales de 2014 aparece la recomendación final de HTML 5.0
- En 2016 aparece la recomendación final de HTML 5.1
- El desarrollo de HTML ya no está sujeto a W3C, sino que las empresas van desarrollando sus propios elementos y luego se va incluyendo en las recomendaciones, aunque W3C sigue publicando recomendaciones y asegura la calidad técnica de HTML. Los navegadores van incluyendo las nuevas etiquetas al criterio de la empresa.

Sitios de interés:

- http://www.evolutionoftheweb.com/static
- https://www.w3.org/TR/2016/REC-html51-20161101/ Recomendación final W3C
- https://whatwg.org/ WHATWG
- https://developer.mozilla.org/en-US/docs/Web/HTML Projecto Mozilla
- https://www.w3schools.com/html/default.asp W3School

1.3.- Elementos del HTML

1.3.1.- Etiquetas

Cada uno de los elementos de un documento HTML será iniciado con una etiqueta. Las ***etiquetas*** son palabras definidas por el lenguaje y que encerraremos siempre entre paréntesis angulares (<>). Algunas etiquetas no incluyen nada más (elementos vacíos). Por ejemplo:

```
<br>
```

es un elemento completo que introduce un salto de línea.

Otros elementos encierran un contenido. En ese caso habrá una ***etiqueta de apertura*** y una ***etiqueta de cierre***. El contenido del elemento será lo que haya entre ambas. La etiqueta de cierre será igual a la de apertura, pero incluyendo el símbolo / justo después de abrir el paréntesis angular. Ejemplo:

```
<title>Título del Documento</title>
```

Etiqueta de apertura: `<title>`

Contenido del elemento: ***Título del Documento***

Etiqueta de cierre: `</title>`

El contenido de un elemento puede incluir, no sólo texto como en el ejemplo, sino nuevos elementos. Cuando esos elementos tengan también apertura y cierre, se estará creando una ***estructura anidada***, en la que siempre se habrá de cerrar en primer lugar la última etiqueta abierta. Ejemplo:

```
<head><title>Título del Documento</title></head>
```

EJEMPLO INCORRECTO:

```
<head><title>Ejemplo de estructura mal anidada.</head></title>
```

El ejemplo anterior está mal anidado porque la etiqueta ***title*** se abre en último lugar y por tanto ha de ser la primera en cerrarse.

NOTA: La recomendación para ***HTML 5.01*** pide que las etiquetas se escriban ***siempre en minúsculas***. Aunque escribirlas de otro modo no crea ningún problema para visualizar los documentos en la mayoría de los navegadores, durante el curso procuraremos seguir las recomendaciones de siempre que sea posible.

1.3.2.- Atributos

Algunas etiquetas permiten concretar su acción mediante ***atributos***. Estos atributos dependen de la etiqueta y son en su mayoría opcionales, pero en algunas ocasiones encontraremos etiquetas con atributos obligatorios. Los atributos son también palabras definidas por el lenguaje que escribiremos en minúsculas según la recomendación, en la etiqueta de apertura, justo antes del paréntesis angular de cierre. Ejemplo:

```
<a href="fichero.html">Contenido del elemento</a>
```

En este ejemplo, la etiqueta ***a*** se acompaña de un atributo ***href*** al que se asigna el valor ***fichero.html***

Una misma etiqueta puede tener varios atributos separados por espacios. No todos los atributos necesitan que se les asigne un valor. Ejemplo:

```
<img src="juego.gif" ismap alt="diana">
```

En este ejemplo ***img*** se acompaña de 3 atributos: ***ismap*** al que no se asigna ningún valor, ***src*** y ***alt***

El valor que se asigna a un atributo puede encerrarse entre comillas dobles, como en los ejemplos, o bien, entre comillas simples, siempre que no se mezclen. También pueden escribirse sin comillas, siempre que no haya espacios y otros atributos. Puede necesitarse usar las comillas simples si la cadena de texto del valor incluye comillas dobles.

Es una buena práctica poner siempre los valores entre comillas.

Hay atributos que se pueden incluir en todas las etiquetas como ***id*** y ***title. id*** identifica a la etiqueta para usarla como referencia y title muestra un cuadro con el texto introducido como valor del atributo.

```
<p id=”nuevo” title=”inicio”>Párrafo de inicio</p>
```

1.3.3.- Comentarios

Como en cualquier otro lenguaje que deba interpretar un ordenador, es posible incluir ***comentarios*** destinados al propio autor o a quienes deseen revisar el documento posteriormente. Estos comentarios no tienen ningún efecto sobre la visualización, de hecho, se ignoran. Es una buena práctica incluir comentarios para aclarar detalles del documento que no se entiendan a simple vista, especialmente cuando se trabaje en grupo o el encargado del mantenimiento sea una persona distinta. Pero incluso cuando se trate de una sola persona, los comentarios ayudan a entender lo que se hizo hace meses o incluso años.

Los comentarios se encierran entre los símbolos <*!*-- y -->. Ejemplo:

```
<!-- Esto es un comentario. El navegador lo ignora, pero
para ti puede ser útil -->
```

1.3.4.- Texto libre

Al margen de las etiquetas, los atributos y los valores, fuera de los paréntesis angulares, nuestro documento tendrá texto, información que queremos visualizar. Este texto es al que nos referimos con ***texto libre***.

Hay que tener en cuenta que el navegador web no lo visualizará exactamente como lo introduzcamos, en particular, ignorará los saltos de línea y cuando encuentre una secuencia de espacios y/o tabuladores los convertirá en un solo espacio.

También es importante saber que los espacios al principio y final de los contenidos de elementos no siempre se respetarán:

EJEMPLO CORRECTO:

```
<p>ofrecemos <b>soporte</b> gratuito</p>
```

EJEMPLO INCORRECTO:

```
<p>ofrecemos<b> soporte </b>gratuito</p>
```

1.3.5.- Elementos de bloque o de línea

HTML distingue entre dos tipos de elementos (de bloque y de línea), quedando otros elementos que no entran dentro de ninguno de los dos grupos:

- ***ELEMENTOS DE BLOQUE*:** Están pensados para contener un conjunto de información que abarca varias líneas. Generan un salto de línea al principio y al final y, EN GENERAL pueden contener otros elementos de bloque y elementos de línea. Ejemplos: <***p***>, <***div***>
- ***ELEMENTOS DE LÍNEA*:** Están pensados para contener información de una sola línea. No genera saltos de línea y, EN GENERAL puede incluir elementos de línea y datos (**pero nunca elementos de bloque**). Ejemplos: <***a***>, <***img***>

Aclaración 1: El contenido que puede incluir un elemento se refiere a lo que hay entre la etiqueta de apertura y etiqueta de cierre (nada tiene que ver con los atributos)

```
<etiqueta atributo1="valor1" atributo2="valor2">Este es el
contenido</etiqueta>
```

Ejemplo de la aclaración 2: La especificación de la ***W3C*** para ***HTML*** dice que el elemento ***BODY*** sólo puede incluir elementos de bloque, pero no elementos de línea. Eso significa que ***BODY*** no puede contener un elemento <***A***> directamente y el siguiente código es erróneo:

Ejemplo incorrecto:

```
<body>
  <a href="fichero.html">pincha aquí</a>
</body>
```

Sin embargo, ***BODY*** puede contener el elemento de bloque <***p***> y <***p***> puede contener elementos de línea. Luego el siguiente código sería correcto:

```
<body>
  <p><a href="fichero.html">Pincha aquí</a></p>
</body>
```

Una vez entendido esto, la realidad es bastante más compleja, pues hay muchas excepciones. Por ejemplo, aunque **<*p*>** sea un elemento de bloque no admite como contenido otros elementos de bloque.

1.4.- Estructura de un documento HTML

Un documento ***HTML*** comenzará siempre por la declaración del tipo de documento (***DOCTYPE***) seguida de una etiqueta **<*html*>**, que naturalmente será la última en cerrarse. Todo lo que se encuentre entre la apertura y cierre de esta etiqueta constituirá nuestro documento.

W3C recomienda que se especifique el idioma de la página incluyendo el atributo lang en la etiqueta html

Tendrá dos partes: una cabecera delimitada por la etiqueta **<*head*>** y su correspondiente cierre y un cuerpo del documento delimitado por la etiqueta **<*body*>**. Ejemplo:

```
<!DOCTYPE html>
<html lang=”es”>
  <head>
    <title>Ejemplo</title>
    <meta charset="UTF-8">
  </head>
  <body>
    <h1>Página de ejemplo</h1>
    <p>Esto es un <a href="demo.html">ejemplo</a> facilito</p>
    <!-- esto es un comentario -->
  </body>
<html>
```

En el ejemplo anterior se observa que el contenido de un elemento está más separado del margen izquierdo que sus etiquetas. A esto se le llama ***indentación*** y resulta muy útil para revisar el documento y, por ejemplo, localizar si hemos olvidado incluir alguna etiqueta de cierre. Cuando usemos una herramienta general, como los editores de texto ***gedit*** o ***Bloc de Notas***, tendremos que indentarlo manualmente. Otras herramientas que están más orientadas a lenguajes de marcas harán ese trabajo por nosotros.

- ***NOTA 1:*** La ***declaración DOCTYPE*** no es un elemento ***HTML***. Fíjate que está antes de la etiqueta ***HTML*** anunciando que el tipo de documento que estamos tratando es un documento ***HTML***. Por este motivo su sintaxis es distinta.
- ***NOTA 2:*** En la cabecera del documento hemos puesto explícitamente el elemento ***title*** porque es obligatorio, mientras que el resto de los elementos que pueden aparecer en la cabecera no lo son.
- ***NOTA 3:*** La ***codificación de caracteres*** no es obligatoria en el sentido de que nuestro documento podrá ser validado sin ella. No obstante, su importancia es tal que recomendamos que se incluya siempre, incluso cuando se esté seguro de que no será necesaria. Más adelante en este mismo tema se cuenta el significado y la sintaxis de la codificación de caracteres.

1.5.- Especificación de W3C para HTML 5.1

Una de las tareas de la ***W3C*** ha sido la de publicar la especificación del ***HTML*** (también otras, como ***CSS***) que detalla lo que oficialmente es el ***HTML***. La implementación de esta recomendación depende de los desarrolladores que no siempre siguen las recomendaciones al pie de la letra.

En la especificación del ***HTML 5.1*** (que podemos encontrar en https://www.w3.org/TR/2016/REC-html51-20161101/) y con la que trabajaremos estrechamente en este módulo) desaparecen los tres modelos de la versión 4.01: la estricta, la transicional y la de marcos.

1.6.- Declaración del tipo de documento

Esta declaración irá incluida al principio del documento, justo antes de la etiqueta <html>:

```
<!DOCTYPE  html>
```

donde se indica al navegador (o agente de usuario) que debe utilizar la recomendación html 5 y posteriores para interpretar correctamente el documento.

La declaración del tipo de documento se ha simplificado mucho con respecto a las versiones anteriores.

Ejemplo de doctype de la versión 4.01:

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

1.7.- URL y URI

Una **URL** (Uniform Resource Locator, Localizador uniforme de recursos) es el nombre completo de un recurso de la web. Naturalmente ese nombre ha de ser único para cada recurso.

La ***URL*** se compone de cuatro partes:

El ***esquema*** especifica el protocolo que hay que usar para acceder al recurso (http, https, ftp, file). Se escribe con minúsculas seguido de :// salvo en el caso de los protocolos news y mailto, en los que se omiten las barras. Ejemplos de esquemas:

```
http://
ftp://
mailto:
```

La segunda parte de una ***URL*** indica el ***servidor*** donde se aloja el recurso (por ejemplo, www.google.com o iesjulioverne.es)

A continuación pondremos la ***ruta*** o ***path***, que es la carpeta donde se encuentra y finalmente el ***nombre*** del recurso.

esquema	servidor	ruta	nombre
http://	iesjulioverne.es	/help/	ayuda.html
La URL será: **http://iesjulioverne.es/help/ayuda.html**			
file://	/	/help/	ayuda.html
La URL será: **file:///help/ayuda.html**			

La descripción que hemos visto corresponde a un **URL absoluto**. Un **URL relativo** sólo necesita la ruta y el nombre. En este caso el recurso deberá encontrarse en el mismo equipo que el documento desde el que se hace referencia.

Una ***URI*** (Uniform Resource Identifier, identificador uniforme de recursos) permite situarse en una parte concreta de un recurso, por ejemplo, al principio de la página 4 de un documento. Para construir una URI sólo hace falta añadir a una URL el símbolo almohadilla (#) y el nombre asignado al fragmento de recurso. Ejemplo:

```
http://iesjulioverne.es/help/ayuda.html#apendice
```

Naturalmente, dentro del documento ***ayuda.html***, en el sitio adecuado ha sido establecido ***apendice*** como nombre de la sección.

1.8.- Validación de documentos HTML

En la página http://validator.w3.org disponemos de una herramienta para validar nuestros documentos ***HTML***, o sea, comprobar si son correctos y conformes a la especificación de la ***W3C***.

La organización WHATWG también tiene su propio validador en la dirección https://checker.html5.org/

Los ***errores*** tienen que ser corregidos siempre, mientras exista un error el documento no será válido.

Los ***warnings*** son errores leves pero aún así debemos corregirlos y nunca permitir un warning que no entendemos. Por ejemplo, la falta de una codificación de caracteres en nuestro documento provocará un warning. También provocará un warning si no reconoce (o falta) la declaración del tipo de documento. Observa que este último warning significa que ni siquiera se ha podido validar.

Nota: Observar que se nos advierte que el sistema es experimental, ya que no hay un estándar oficial definido. La nueva versión de HTML es mucho menos restrictiva respecto a los elementos que están o no permitidos.

1.9.- Codificación de caracteres

Un documento ***HTML*** es una secuencia ordenada de caracteres. Tal y como la muestra el editor (o cualquier otra herramienta que usemos para escribir el código) son caracteres reconocibles por los humanos. Cuando guardamos el documento en un fichero de texto, cada uno de esos

caracteres se guarda como un código numérico. El código que corresponde a cada carácter dependerá del ***conjunto de caracteres*** que estemos usando.

Un conjunto de caracteres estará formado por un ***repertorio*** (conjunto abstracto de caracteres) y unas ***posiciones de código*** (o sea, la asignación de un número entero distinto a cada uno de los caracteres del repertorio). Hay que tener en cuenta que el conjunto de caracteres nada tiene que ver con la representación que se haga de esos caracteres. Por ejemplo, en el repertorio puede aparecer como carácter la letra ***a***, pero es el concepto abstracto de la letra ***a*** que podrá ser representada de muchas formas distintas. A cada una de estas representaciones o dibujos de los caracteres se les llama ***glifos***.

Las letras mayúsculas y minúsculas suelen aparecer en el repertorio de un conjunto de caracteres como caracteres distintos y, por tanto, con códigos asignados distintos.

Un ejemplo de conjunto de caracteres es el ***ASCII*** (American Standard Code for Information Interchange) (Código Estándar Americano para Intercambio de Información). Éste conjunto de caracteres dispone de 256 posiciones de código: la primera mitad (del 0 al 127) es lo que se denomina ***ASCII Standard*** e incluye los caracteres más usuales (para la población americana, claro está). La segunda mitad es conocida como ***ASCII extendido*** y dispone de muchas versiones adaptadas a distintos países. (En el ***ASCII extendido*** para España se incluyen caracteres como la ***ñ*** mayúscula y minúscula y las vocales con tilde). Cada carácter del código ***ASCII*** se representa con 8 bits, esto es, un byte.

En seguida vemos los problemas que nos acarreará usar el código ***ASCII*** para codificar nuestra página web. Por lo pronto, una página hecha en España incluirá vocales con tilde y si alguien la visualiza desde Grecia, donde seguramente la parte extendida del ***ASCII*** será distinta, verá un montón de símbolos incomprensibles aunque sepa castellano.

Otros problemas vendrán si tenemos que incluir en una página en castellano una cita en otro idioma, por ejemplo alemán con el carácter *beta*. O en el caso de que estemos escribiendo un texto de física y necesitemos símbolos matemáticos como el que hace referencia a las integrales o a las sumatorias.

Resulta evidente que necesitamos un juego de caracteres universal, que nos permita representar todos los caracteres posibles. La ***W3C*** establece el ***ISO 10646*** como el conjunto de caracteres para ***HTML***. Este conjunto de caracteres es ampliable, de modo que se van añadiendo progresivamente los que se echen en falta, pero conservando los códigos de los anteriores (o sea, nunca una revisión del ***ISO 10646*** cambiará el código de un carácter que ya estuviera en el conjunto).

Existen otros muchos conjuntos de caracteres, por ejemplo:

ISO-8859-15

UTF-8 (Unicode Transformation Format)

Este último incluye todos los caracteres del ***ISO 10646*** con los que se corresponde uno a uno. Por tanto, es este el que recomendamos usar en nuestro código ***HTML***.

Nota: ***UTF*** es compatible con ***ASCII***. ***UTF-8*** usa de uno a cuatro bytes para cada carácter, mientras que ***UTF-16*** usa un mínimo de dos bytes por carácter.

Dentro del documento tendremos que informar del conjunto que hemos usado. Para ello incluiremos un elemento ***META*** al principio de la cabecera (***HEAD***) de la siguiente forma:

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

Sustituyendo ***UTF-8*** por la codificación de caracteres que hemos usado.

Hay otros dos momentos en los que se puede crear un problema con los juegos de caracteres:

El navegador tiene establecido un conjunto por defecto, no podemos configurar todos los navegadores de las personas a las que va dirigida la página. Normalmente esto no será un problema porque debe primar lo que definimos en el documento sobre la configuración.

Los servidores que sirven la información en internet pueden cambiar la codificación. Aquí poco podemos hacer, salvo cambiar nuestro conjunto si detectamos que esto ocurre con nuestro servidor.

Es posible que en alguna ocasión tengamos que expresar caracteres que no estén incluidos en la codificación elegida. En ese caso echaremos mano de las ***referencias de caracteres***:

- **numéricas**: Mediante las referencias numéricas podemos incluir caracteres que no estén en nuestro teclado. Sólo tendremos que incluir la posición del carácter deseado (según el conjunto ***ISO 10646***) precedido de los símbolos ampersand ***(&)*** y almohadilla ***(#)*** y seguido de punto y coma ***(;)*** Por ejemplo, el código del carácter "***a***" en ***ISO 10646*** es el 229. Podemos hacer referencia a la letra "a" con la siguiente secuencia: ***å*** También podemos usar su equivalente hexadecimal ***E5*** si anteponemos al número la letra ***x***. Por ejemplo ***å å å å*** (Todas estas expresiones son equivalentes)
- **referencias a entidades de caracteres**: Las referencias a entidades de caracteres son palabras definidas en el ***HTML***. Cuando usemos un carácter "raro" con frecuencia nos será más fácil recordar este nombre que su código numérico. En la siguiente tabla hay unas cuantas (son muchas más):

Refer.	**Carácter**		**Refer.**	**Carácter**		**Refer.**	**Carácter**
á	á		Á	Á		ñ	ñ
é	é		É	É		Ñ	Ñ
í	í		Í	Í			espacio
ó	ó		Ó	Ó			
ú	ú		Ú	Ú			

Especial atención merecen ***<*** para el símbolo "menor que" ***(<)***, ***>*** para el símbolo "mayor que" ***(>)***, ***"*** para las comillas ***("")*** y ***&*** para el símbolo "ampersand" ***(&)*** que deberán usarse siempre mediante referencias, ya que su uso directo crearía problemas de interpretación por el significado especial de estos caracteres en la sintaxis de ***HTML***.

NOTA: La referencia a entidad de caracteres *** *** actúa como un espacio insertado con la barra espaciadora pero hay dos diferencias:

**** HTML*** reduce una secuencia de varios espacios a uno solo, pero esto no ocurre si los espacios están indicados con la referencia a entidad de caracteres *** ***

* Si la separación entre dos palabras es *** *** en lugar del espacio habitual, entonces esas dos palabras permanecerán en la misma línea, esto es, si no caben al final de una línea bajarán a la siguiente, pero las dos palabras.

1.10.- Etiquetas básicas

1.10.1.- Título de la página: <title>

Contiene el título de nuestro documento, que aparecerá en la barra de título. Es ***obligatoria*** para todo documento ***HTML*** y debe formar parte de la cabecera (***head***)

1.10.2.- Párrafo: <p>

Contiene un párrafo del documento. Provoca un salto de línea antes y después de la etiqueta al ser de bloque. Se recomienda que no haya texto libre en el body.

```
<p>Texto de ejemplo</p>
```

Si queremos producir un salto de línea dentro del párrafo, se recomienda usar **
** en vez de <p>

```
<p>Texto de ejemplo <br> con dos líneas</p>
```

1.10.3.- Enlaces: la etiqueta A

La etiqueta ***a*** es la que nos permite incluir enlaces en nuestros documentos ***HTML***.

Un enlace tiene dos partes: el origen y el destino.

El ***origen del enlace*** es una zona activa que permite al usuario "saltar" a otro documento o una parte de este u otro documento. La zona activa mostrará el contenido de la etiqueta destacado de alguna forma que indique al usuario que se trata de un enlace (generalmente en color azul y subrayado). La sintaxis para incluir un origen de enlace es la siguiente:

```
<a href="URIdestino">Zona activa</a>
```

donde:

- ***URIdestino*** es la ***URL*** (o la ***URI***) que contiene la dirección del documento (o parte del documento) al que queremos saltar. Obsérvese que usamos el atributo ***href*** para indicar la ***URI*** de destino.
- ***Zona activa*** es el texto que aparece en el documento y sobre el que tendrá que actuar el usuario para provocar el salto.

El **destino del enlace** es el documento o el punto interno de un documento al que queremos que se viaje. Cuando nuestro objetivo es una parte concreta del documento necesitamos señalar esa parte asignándole un nombre para poder referenciarla. Un destino de enlace tiene la siguiente sintaxis:

```
<etiqueta id="nombreSitio">
```

donde:

- ***nombreSitio*** es el nombre que estamos asignando a esa parte del documento. Obsérvese que el atributo que usamos es ***id*** y que el nombre del sitio NO va precedido del símbolo almohadilla ***(#)***
- ***etiqueta*** *es cualquier etiqueta html*

EJEMPLO: SALTO A OTRO DOCUMENTO

```
<a href="miPagina.html">Ir a mi página</a>
```

En este caso queremos ir al documento ***miPagina.html***, concretamente al principio de ese documento y no hace falta poner un destino de enlace.

EJEMPLO: SALTO A UN APARTADO DEL MISMO DOCUMENTO

```
<a href="#parrafo2">Ir al párrafo 2</a>
```

En este caso queremos desplazarnos hasta el segundo párrafo del documento. El nombre del destino comienza por el símbolo almohadilla ***(#)***, porque es un punto interno del documento. Será necesario asignar el nombre en el lugar de destino de la siguiente manera:

```
<p id="parrafo2">Contenido del párrafo 2</p>
```

EJEMPLO: SALTO A UN APARTADO DE OTRO DOCUMENTO

```
<!-- Código en el documento de origen -->
<a href="miPagina.html#parrafo2">Ir al párrafo 2 de mi página</a>
```

En este caso queremos desplazarnos hasta el segundo párrafo del documento ***miPagina.html***. Dentro de ese otro documento, en el punto de destino, será necesario asignar el nombre:

```
<!-- Código en el documento de destino (miPagina.html) -->
<p><a id="parrafo2"></a>Contenido del párrafo 2</p>
```

Atributos permitidos:

- **href**: especifica la dirección de destino
- **target**: especifica dónde se abre el nuevo documento.
  - **_blank**: una nueva ventana
  - **_parent:** en el marco padre
  - **_self**: en la misma ventana (por defecto)
  - **_top**: ocupa toda la ventana
  - **id_elemento**: en un elemento definido en la página

1.10.4.- Encabezados: las etiquetas H

Los **encabezados** de un documentos son los títulos, subtítulos, etc. de cada unos de los capítulos, secciones o apartados en los que se divide. En un documento bien construido estos encabezados siguen una jerarquía: en el primer nivel estará el título del documento, en el segundo nivel estarán los capítulos en que se divide el tema, en el tercer nivel estarán las secciones en las que se divide cada capítulo y así sucesivamente.

HTML nos permite indicar estos encabezados hasta 6 niveles con los elementos ***h1, h2, h3, h4, h5 y h6***, siendo ***h1*** el encabezado de primer nivel. Cada agente de usuario mostrará estos encabezados con mayor resaltado en primer nivel y menor resaltado en el último nivel, si bien el aspecto de un mismo nivel diferirá de un navegador a otro. El objetivo de estos elementos es el de "estructurar" el documento de forma que posteriormente podremos usarlos para diferentes actividades (por ejemplo, construir un índice automáticamente). Por tanto no deben usarse como etiquetas presentacionales o, dicho de otra manera, *NUNCA DEBEN USARSE PARA RESALTAR TEXTO*.

EJEMPLO INCORRECTO:

```
La siguiente palabra aparece en negrita: <h5>Negrita</h5> pero este uso de encabezado es incorrecto
```

El siguiente ejemplo muestra un uso correcto de encabezados:

```
<h1>Tema 1: Seres vivos</h1>
<h2>Capítulo 1: Animales</h2>
<h3>Clasificación de animales</h3>
<p>Aquí escribimos el texto</p>
<h3>Reproducción de animales</h3>
<p>Aquí escribimos el texto</p>
<h2>Capítulo 2: Plantas</h2>
<h3>Clasificación de plantas</h3>
<p>Aquí escribimos el texto</p>
<h3>Reproducción de plantas</h3>
<p>Aquí escribimos el texto</p>
```

1.10.5.- Salto de línea y separador horizontal

Salto de línea: **<*br*>** , se utiliza para insertar saltos de línea dentro de los párrafos. No se recomienda usarlos para separar párrafos.

Seperador horizontal: **<*hr*>** representa un salto temático y aparece como una línea horizontal

1.10.6.- Imágenes

Aunque un navegador puede mostrar imágenes en casi cualquier formato debemos preferir los formatos ***GIF***, ***PNG, JPG*** (o ***JPEG***) y ***SVG***. Los dos primeros son más adecuado para dibujos, mientras que el ***JPG*** permite muchos más colores y será mejor para fotografías. Entre ***GIF*** y ***PNG*** nos decantamos por ***PNG*** que usa el mismo algoritmo pero mejorado consiguiendo archivos más ligeros cuando el número de colores supera los 256. ***SVG*** se utiliza para dibujos vectoriales que adaptan su tamaño sin perder resolución (en realidad son archivos XML).

<img>

La etiqueta que se utiliza es ***<img>*** que es un elemento vacío pero que admite los siguientes atributos:

- ***src*** (obligatorio): para indicar la ***URI*** de la imagen
- ***alt*** (obligatorio): para añadir un comentario alternativo. Este comentario debe hacer una descripción breve de la imagen que será necesario cuando la página se represente en un agente de usuario no visual o cuando en un AU visual, por algún motivo, no se pueda mostrar la imagen.
- ***width***: anchura (en pixels) de la imagen. (html 5 no admite porcentajes)
- ***height***: altura de la imagen.

La altura y/o anchura que proporcionamos indica el tamaño que ocupará en pantalla (no es el tamaño de la imagen original). Téngase en cuenta que si los valores que ponemos no guardan las proporciones originales, la imagen aparecerá deformada. Si sólo damos valor a la altura o sólo damos valor a la anchura, el otro parámetro se adaptará a las proporciones originales.

```
<img src="familia.png" alt="mi familia en el lago" width="200">
```

<picture>

HTML 5 incluye una nueva etiqueta para mostrar imágenes (orientada al diseño *responsive*). Con <picture> podemos especificar varios orígenes de archivos de imágenes que serán cargados en función del tamaño de la pantalla. ***Puede que no esté implementado en todos los navegadores o no funcione correctamente***

La etiqueta está compuesta por un elemento <img> y uno o varios <source>. Si el navegador no cumple con los criterios de la etiqueta source, se salta y pasa a la siguiente. Si ninguno lo cumple, utiliza la etiqueta <img> que es obligatoria.

```
<picture>
 <source srcset="logo-ancho.png" media="(min-width: 600px)">
 <source srcset="logo-peque.png" media="(min-width: 300px)">
 <img src="logo-defecto.png" alt="Logo">
</picture>
```

1.11.- Etiquetas permitidas cambio de aspecto

Aunque se recomienda que el aspecto y el contenido estén claramente separados y que html se ocupe solo del contenido y CSS del aspecto, HTML 5 admite ciertas etiquetas para modificar el aspecto del texto:

ETIQUETA	**USO**
******	Cuando queremos enfatizar un texto
******	Cuando queremos resaltar el texto
<s>	Cuando queremos tachar el texto
<small>	Cuando queremos el texto más pequeño

<***cite***>	Se utiliza para indicar títulos de libros, obras, etc
<***code***>	Cuando queremos indicar que el texto es un código informático
<***sub***> <***sup***>	Cuando queremos que el texto aparezca como subíndice o superíndice

Hay que tener en cuenta que no debemos abusar de estas etiquetas, que algunas están pensadas para su uso con lectores de voz. **El contenido de estas etiquetas es texto y elementos de línea nunca de bloque**

1.12.- Listas

Las **listas** son conjuntos de elementos jerarquizados que se muestran uno tras otro en líneas consecutivas (aunque un elemento puede ocupar varias líneas). Un ejemplo de lista puede ser el índice que hay al principio de este tema. Ese ejemplo constituye una **lista ordenada** porque cada uno de los elementos de la lista está precedido por un número. Cuando delante de cada elemento hay un guión o una flechita o cualquier otro símbolo igual para todos los elementos de la lista, hablamos de **listas no ordenadas**.

Las etiquetas HTML para crear listas son: ***ol, ul y li***.

<ol>: Su contenido (entre etiquetas de apertura y cierre) será una lista ordenada.

<ul>: Su contenido será una lista no ordenada.

<li>: Delimita cada uno de los elementos de la lista, ya sea ordenada o no ordenada

Elemento	Es	Puede contener	Etiqueta final
ol	Bloque	Sólo elementos ***li***	Obligatoria
ul	Bloque	Sólo elementos ***li***	Obligatoria
li	--------	Elementos de línea y bloque	Opcional, pero recomendada

Un ejemplo de lista ordenada sería el siguiente:

```
<p>La clasificación final de Moto GP ha quedado así:</p>
<ol>
    <li>Jorge Lorenzo</li>
    <li>Dani Pedrosa</li>
    <li>Valentino Rossi</li>
</ol>
```

Un agente de usuario mostrará los tres elementos anteponiéndoles, por ejemplo, los números 1, 2 y 3:

Si cambiamos ***OL*** por ***UL***, obtendremos una lista no ordenada, donde cada elemento será precedido por un mismo símbolo.

```
<p>Los ingredientes para esta receta son:</p>
<ul>
    <li>Macarrones</li>
    <li>Nata Líquida</li>
    <li>Yema de huevo</li>
</ul>
```

NOTA: Podremos modificar la numeración en las listas ordenadas (números, números romanos, letras mayúsculas o minúsculas, ...) o el símbolo que aparece ante los elementos de las lista no ordenadas (guión, círculo, flecha,...). Esto tendremos que hacerlo desde una hoja de estilo.

Podemos anidar listas, pero siempre teniendo en cuenta que los elementos ***OL***, ***UL*** no pueden incluir listas, sólo ***LI***. Supongamos que queremos conseguir la siguiente lista:

Obsérvese en el dibujo que los elementos de la lista principal *CONTIENEN TODA LA SUBLISTA*, por tanto el siguiente ejemplo sería *INCORRECTO* :

```
<p>Miembros del equipo:</p>
<ul>
  <li>Delanteros</li>
  <ul>
    <li>Iniesta</li>
    <li>Navas</li>
  </ul>
  <li>Defensas</li>
  <ul>
    <li>Pujol</li>
    <li>Arbeloa</li>
  </ul>
</ul>
```

Las dos sublistas que están tachadas son incorrectas porque hemos incluido un elemento ***UL*** dentro de otro. La *FORMA CORRECTA* sería la siguiente (***listaAnidada.html***):

```
<p>Miembros del equipo:</p>
<ul>
  <li>Delanteros
    <ul>
      <li>Iniesta</li>
      <li>Navas</li>
    </ul>
  </li>
  <li>Defensas
    <ul>
      <li>Pujol</li>
      <li>Arbeloa</li>
```

```
    </ul>
  </li>
</ul>
```

Ahora, la lista principal sólo contienes elementos ***LI***. El contenido de esos elementos puede ser una lista completa, como vemos en el ejemplo.

1.12.1.- Listas de definiciones

Las **listas de definiciones** permiten estructuras como la de un diccionario donde aparecen términos y sus definiciones.

Los elementos que usamos para las listas de definiciones son:

dl: Contiene la lista de definiciones.

dt: Contiene un término de la lista.

dd: Contiene una definición de la lista.

Elemento	Es	Puede contener	Etiqueta final
dl	Bloque	Sólo elementos ***dt*** o ***dd***	Obligatoria
dt	--------	Sólo elementos de línea	Opcional, pero recomendada
dd	--------	Elementos de línea y bloque	Opcional, pero recomendada

Los términos y definiciones no tienen que alternarse necesariamente, esto es, pueden aparecer varios términos o definiciones consecutivas.

```
<dl>
  <dt>Centro</dt>
  <dd>Punto equidistante a todos los puntos de la superficie de
la esfera.</dd>
  <dd>En algunos deportes, el jugador que se sitúa en la posición
central del campo.</dd>
</dl>
```

1.13.- Tablas

1.13.1.- Estructura de una tabla sencilla

Para construir tablas en HTML utilizamos el elemento ***table***. Toda nuestra tabla se describirá entre las etiquetas **<*table*>** y **</*table*>**

- Una tabla está compuesta por filas (definidas por el elemento ***TR*** (del inglés Table Row - Fila de Tabla)
- Una fila está compuesta por celdas (definidas por el elemento ***TD***)
- Al principio de la tabla podemos incluir opcionalmente el elemento ***caption*** para mostrar un título de la tabla.

Veamos un ejemplo de una tabla con tres filas y dos celdas por fila :

```
<table>
  <caption>Ejemplo de tabla</caption>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
  </tr>
  <tr>
    <td>Celda 3</td>
    <td>Celda 4</td>
  </tr>
  <tr>
    <td>Celda 5</td>
    <td>Celda 6</td>
  </tr>
</table>
```

1.13.2.- Atributos y valores del elemento table

<***table***> ha perdido todos los atributos en html 5. Para modificar el aspecto que presenta una tabla hay que utilizar **CSS**. El atributo ***border*** que indica el grosor de las líneas de separación en algunos validadores aparece como **aviso** y en otros como **error**.

1.13.3.- Atributos de los elementos TR, TD y TH

Hemos visto como una celda se representa mediante el elemento ***TD***. El elemento ***TH*** representa igualmente una celda pero con un matiz: se trata de una celda cuyo contenido afecta a otras celdas, algo así como un encabezamiento. Los agentes de usuario suelen representar el contenido de las celdas ***TH*** con un resaltado superior al de las celdas ***TD***.

Ejemplo: si tenemos una tabla que representa el horario de clases de un grupo del instituto, probablemente las celdas de la primera fila contengan el nombre de los días de la semana. Todas esas celdas son cabecera de su columna y, por tanto, deberían marcarse con ***TH***.

Los elementos celda ***TD*** y ***TH*** pueden usar los atributos ***rowspan*** y ***colspan*** para construir celdas que ocupan más de una fila o más de una columna respectivamente.

```
<td colspan="2">Esta celda tiene la anchura de dos celdas</td>
<td rowspan="3">Esta celda tiene la altura de tres celdas</td>
```

A continuación tenemos una tabla y el código necesario para conseguirla:

```
<table>
  <caption>Mi primera tabla:</caption>
  <tr>
```

```
    <th>Lunes</th>          <!-- ATENCION: celdas TH -->
    <th>Martes</th>
    <th>Miércoles</th>
    <th>Jueves</th>
  </tr>
  <tr>
    <td  rowspan="2">Taller</td>
    <td></td>
    <td  colspan="2">Llevar niños al cole</td>
  </tr>
  <tr>
    <td>Estudio</td>
  </tr>
  <tr>
    <td></td>
    <td  rowspan="2" >Compra</td>
    <td>Compra</td>
    <td  rowspan="2" >Compra</td>
  </tr>
  <tr>
    <td>Cocinar</td>
    <td>Cocinar</td>
  </tr>
</table>
```

1.13.4.- Grupos de filas

Las filas se agrupan en bloques de tres tipos: ***cabecera***, ***pie*** y ***cuerpo***. Sólo puede haber una cabecera y un pie, pero puede haber ***varios cuerpos*** en una misma tabla. Es especialmente útil incluir cabecera y pie porque esto permite que un agente de usuario pueda mantenerlos fijos mientras se desplazan el resto de filas de la tabla.

Lo primero que se define es la cabecera con las etiquetas <***thead***> y <***/thead***> que englobarán las filas de la cabecera. A continuación se define el el cuerpo (o cuerpos) de la tabla con <***tbody***> y <***/tbody***> al final el pie con las etiquetas <***tfoot***> y <***/tfoot***> .

En el siguiente ejemplo hemos recuperado la tabla anterior, le hemos especificado como cabecera la primera fila y además hemos añadido una columna al principio con el horario. Esta primera columna está compuesta de celdas de encabezamiento porque el horario afecta a todas las celdas de la misma fila. También hemos añadido un pie que es idéntico a la cabecera.

```
<table >
  <caption>Mi segunda tabla:</caption>
  <thead>
    <tr>
      <th>Hora</th>
      <th>Lunes</th>
      <th>Martes</th>
      <th>Miércoles</th>
      <th>Jueves</th>
```

```
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>8:30h</th>
      <td  rowspan="2">Taller</td>
      <td>Libre</td>
      <td  colspan="2">Llevar niños al cole</td>
    </tr>
    <tr>
      <th>10:00h</th>
      <td  colspan="3">Estudio</td>
    </tr>
    <tr>
      <th>11:00h</th>
      <td>Libre</td>
      <td  rowspan="2" >Compra</td>
      <td>Compra</td>
      <td  rowspan="2">Compra</td>
    </tr>
    <tr>
      <th>12:30h</th>
      <td>Cocinar</td>
      <td>Cocinar</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Hora</th>
      <th>Lunes</th>
      <th>Martes</th>
      <th>Miércoles</th>
      <th>Jueves</th>
    </tr>
  </tfoot>
</table>
```

1.13.5.- Columnas y grupos de columnas

Hasta ahora sólo hemos hablado de filas y de celdas. Obsérvese que las columnas aparecen por si solas y, por tanto, no hace falta crearlas.

Sin embargo, es posible que queramos hacer referencia a todas las celdas de una misma columna para, por ejemplo, indicar la anchura que deben tener **utilizando CSS**.

Una columna se define con el elemento ***COL***, al principio de la tabla antes de crear las filas. La definición de una columna ***NO CREA CELDAS***, las celdas se crean exclusivamente con los elementos ***TD*** y ***TH*** dentro de las filas. La definición de una columna es solamente una forma de poder referenciarla.

El primer elemento ***COL*** hace referencia a la primera columna de la tabla, el segundo elemento ***COL***, a la segunda columna de la tabla, y así sucesivamente.

```
<table>
```

```
  <COL id="primera">

  <COL id="segunda">


  <tr>
      <td> <!-- Primera celda --> </td>

      <!-- Resto de filas y celdas .....  -->
</table>
```

Si las características que deseamos aplicar son las mismas para varias columnas consecutivas, podemos hacerlo con un solo ***COL***, usando el atributo ***span***:

```
<COL id="uno" span="3">
```

- significa que la columna identificada como "uno" ocupa 3 columnas consecutivas.

NOTA: Obsérvese que a pesar de la similitud con ***colspan*** y ***rowspan***, el significado de ***span*** es totalmente distinto: el valor de ***span*** indica el número de columnas a las que afectan las características que estamos detallando.

De la misma forma que hacíamos grupos de filas, también podemos hacer grupos de columnas. El elemento que nos permite definir los grupos de columnas es ***COLGROUP***. Un elemento ***COLGROUP*** agrupa todos los elementos ***COL*** que formarán parte del grupo.

```
<COLGROUP>
  <COL id="uno1">
  <COL id="uno2">
  <COL id="uno3">
</COLGROUP>
<COLGROUP id="grupoUno">
```

1.14.- Etiquetas multimedia, audio, video, iframe y object

HTML 5 ha mejorado mucho la gestión de los elemento multimedia respecto a versiones anteriores.

1.14.1.- iframe

La etiqueta <***iframe***> permite insertar elementos exteriores en nuestra página, como otras páginas html, vídeo, etc.

```
<iframe src="http://www.iesjulioverne.es"></iframe>
```

Atributos permitidos:

En HTML 5 la mayoría de los atributos han desaparecido, controlándose mediante CSS:

- ***src***: indica la ruta del elemento
- ***height***: especifica la altura del marco en píxeles

- ***width***: especifica la anchura del marco en píxeles
- ***name***: nombre para identificar el marco. Se puede usar como destino en target de <a>
- ***srcdoc***: Especifica el contenido HTML de la página que se muestra en el <iframe>. Al usarlo deja sin efecto el atributo ***src.***
- ***sandbox***: se utiliza para aplicar restricciones al documento que se muestra en el marco, como evitar que se ejecuten scripts, formularios, etc. Si sandbox está presente sin valor, se aplican todas las restricciones. Los valores permitidos para sandbox son: ***allow-forms, allow-pointer-lock, allow-popups, allow-same-origin, allow-scripts,*** *y* ***allow-top-navigation***

```
<iframe src=”mipagina.html” sandbox=”allow-forms allow-popups”
name=”marco” height=”200” width=”100”></iframe>
```

1.14.2.- Vídeo

Para reproducir vídeos se necesitaban pluggins compatibles con el formato a reproducir. Con la nueva etiqueta <***video***> de html 5 ya no es necesario.

```
<video src=”mivideo.mp4” >Tu navegador no admite videos</video>
```

El navegador reproducirá el archivo especificado en el atributo src.

La etiqueta admite que se especifiquen varios formatos de vídeos para los distintos navegadores con la etiqueta <source>

```
<video width="640" height="360" controls preload>
     <source src="video.mp4" type='video/mp4; codecs="avc1,mp4a"' />
     <source src="video.ogv" type='video/ogg; codecs="theora,vorbis"' />
     <source src="video.webm" type='video/webm; codecs="vp8,vorbis"'/>
</video>
```

Atributos permitidos

poster: permite insertar una imagen de sustitución cuando el vídeo no está disponible.

controls: se incluyen controles de play, pause...

width y ***height***: ancho y alto, se ajusta al valor y el otro se calcula automáticamente manteniendo la proporción.

loop: reproduce el vídeo continuamente .

autoplay: reproduce el vídeo automáticamente.

preload: carga el vídeo

HTML 5 solo admite los formatos de vídeo MP4, WebM, y Ogg

Para más información sobre los formatos y los navegadores https://developer.mozilla.org/es/docs/Web/HTML/Formatos_admitidos_de_audio_y_video_en_html5

Otra forma de insertar un vídeo y evitar los problemas de formato es subirlo a Youtube e insertar el vídeo usando iframe

```
<iframe width="640" height="360" src="https://www.youtube.com/embed/CFp_8Sah6YU"
frameborder="0" allowfullscreen>
</iframe>
```

Algunos de los parámetros de YouTube:

autohide , valor 0 para tener los controles visibles y 1 si prefieres que se oculten cuando el video se está reproduciendo.

autoplay, 1 si quieres que el vídeo se reproduzca automáticamente, 0 si espera al play.

controls, con 0 los controles no se muestran con 1 si y en ambos el vídeo se descarga el vídeo, con 2 los controles se muestran pero el video no se descarga hasta hacer play.

Si queremos incluir más de un parámetro se enlazan con &

```
<iframe width="640" height="360"
src="https://www.youtube.com/embed/Cfp_8Sah6YU?autoplay=0&controls=0" >
</iframe>
```

Para más información sobre controles del reproductor Youtube en los navegadores: https://developers.google.com/youtube/player_parameters#controls

1.14.3.- Audio

El audio EN html4 se inserta de manera similar a los vídeos pero usando la etiqueta <***audio***>:

```
<audio src="audio.mp3" controls autoplay loop >
</audio>
```

El formato mp3, no es un formato abierto, por lo que para aumentar la compatibilidad podemos usar varios formatos con <***source***>

```
<audio controls>
      <source src="audio.ogg" type="audio/ogg">
      <source src="audio.mp3" type="audio/mpeg">
      <source src="audio.wav" type="audio/wav">
</audio>
```

1.14.4.- El elemento OBJECT

El elemento ***OBJECT*** nos permite incluir un objeto en nuestra página. Disponemos de varios tipos de objetos, entre ellos:

- ***una imagen***: en este caso el elemento ***OBJECT*** actúa como el elemento ***IMG***

- ***un documento HTML:*** en este caso ***OBJECT*** actúa como el elemento ***IFRAME***, pero no podrá ser marco de destino.
- ***un vídeo***
- ***una aplicación:*** las aplicaciones sólo pueden incluirse a través de ***OBJECT***

El elemento ***OBJECT*** tiene contenido pero ***SÓLO SE MUESTRA COMO ALTERNATIVA AL OBJETO***, esto es, si el objeto se encuentra y se puede mostrar sin problemas, el contenido será ignorado.

Los atributos que admite este elemento son:

ATRIBUTO	*VALORES*	*FUNCIÓN*
data	URI	Recurso que contiene el objeto
type	***image/png*** ***image/gif*** ***video/mpg*** ***text/css*** ***audio/basic*** ***text/html***	Tipo de contenido

EJEMPLOS:

```
<object data="familia.png" type="image/png">mi familia en el lago</object>
```

Representa la imagen en las mismas condiciones que el ejemplo anterior. Obsérvese que ***EL CONTENIDO NO APARECERÁ*** si la imagen se encuentra disponible, igual que el atributo ***alt*** en ***IMG***.

```
<object data="prueba.html" type="text/html">
```

Crea un marco que contiene el documento ***prueba.html***.

```
<object data="pelicula.mpg" type="video/mpg">
```

Crea un marco que muestra el vídeo ***pelicula.mpg***.

```
<object classid="java:coche.class" codetype="application/java"
width="500" height="500">
  Un vehículo en movimiento
</object>
```

Reproduce una ***applet java*** que se encuentra en el fichero ***coche.class***.

NOTA: En el último objeto (***applet java***) usamos el atributo ***classid*** en lugar del atributo ***data*** y usamos el atributo ***codetype*** en lugar del atributo ***type***. Los tipos de objetos que admite una página ***HTML*** son muy variados y los atributos que los definen también. No es materia de este módulo entrar en esos detalles.

NOTA: El atributo ***type*** no es obligatorio, sin embargo, ayuda a un navegador a saber si soporta el tipo de objeto antes de cargarlo (y en consecuencia, no cargarlo si no está soportado). No obstante, el objeto incluirá su tipo en la cabecera ***HTTP*** y en caso de contradicción con el especificado en el atributo ***type***, prevalecerá lo indicado en la cabecera.

1.15.- Formularios

Hasta este momento hemos visto como construir páginas en las que la información fluye siempre en un único sentido, desde la página hasta el usuario. Los formularios son el mecanismo que ***HTML*** proporciona para permitir que la información viaje en sentido contrario, es decir, que se pueda recoger información que aporta el usuario.

Un formulario estará formado por un conjunto de ***controles*** que pueden ser muy variados (cajas de texto, casillas de verificación, combos, botones, menús etc) y con los que cualquier usuario de sistemas informáticos gráficos estará familiarizado. Cada uno de esos controles tendrá asignado un nombre y tomará un valor según las acciones que el usuario efectúe sobre ellos.

En cada formulario debe de existir un control que permita darlo por finalizado y envíe los datos al lugar donde se van a procesar. Los datos que se envían son las parejas nombre/valor de todos los controles (en realidad sólo las de aquellos controles que tengan un valor).

El formulario estará contenido entre las etiquetas ***<FORM>*** y ***</FORM>*** y además de los controles puede incluir texto, líneas, saltos de línea y todos los elementos ***HTML*** que necesitemos para darle un mejor aspecto. Hasta la versión 5, los controles debían estar en el contenido de <form>, pero ahora con el atributo ***form*** de los controles se puede vincular un control con el ***id*** de un form.

Atributos permitidos:

La etiqueta form tiene algunos atributos obligatorios:

action: el formulario es enviado a la dirección especificada en action

method: especifica cómo debe ser enviado los datos. Admite los valores ***get*** y ***post*** (por defecto).

name: especifica el nombre del formulario

target: especifica dónde debe ser mostrado el resultado del envío del formulario

autocomplete: admite los valores:

- **on** (los campos muestran datos introducidos anteriormente por el usuario)
- **off** (no se muestran los valores).

Este atributo también puede ser usado en controles dentro del formulario.

novalidate: si aparece, los campos no se comprueban automáticamente antes del envío.

1.15.1.- Tipos de controles

En la tabla siguiente mostramos los tipos de controles que se pueden incluir en un formulario junto al elemento ***HTML*** que tenemos que usar.

HTML 5 ha incluido numerosos nuevos controles, aunque su aspecto y funcionalidad depende mucho del navegador utilizado:

CONTROL	*DESCRIPCIÓN*	*CÓDIGO HTML*	*OTROS ATRIBUTOS*
Cuadro de texto	Caja en la que el usuario puede introducir un texto corto	`<input type="text">`	***size maxlength***
Cuadro contraseña	Como el cuadro de texto, pero al escribir se visualizan todos los caracteres con un único símbolo, por ejemplo, asteriscos	`<input type="password">`	***size maxlength***
Casilla de verificación	Cajita que puede marcarse o desmarcarse	`<input type="checkbox">`	***checked***
Radiobotón	Círculo que puede marcarse o desmarcarse. La diferencia con las casillas de verificación es que sólo se puede marcar un radiobotón del grupo	`<input type="radio">`	***checked***
Color	Control para especificar un color. Una interfaz de selección de color no requiere más funcionalidad que la de aceptar colores simples como texto	`<input type="color">`	
Fecha	Control para introducir una fecha (año, mes y día, sin tiempo).	`<input type="date">`	
Fecha y hora local	Control para introducir fecha y hora, sin zona horaria específica.	`<input type="datetime-local">`	
Email	Campo para introducir una dirección de correo electrónico. El valor introducido se valida para que contenga una cadena vacía o una dirección de correo válida antes de enviarse.	`<input type="email">`	
Fichero	Control que permite al usuario seleccionar un archivo	`<input type="file">`	***accept***
Mes	Control para introducir un mes y año, sin zona horaria específica	`<input type="month">`	
Número	Control para introducir un número de punto flotante	`<input type="number">`	
Rango	Control para introducir un número cuyo valor está en un rango.	`<input type="range">`	***min max***

			step
Área de texto	Como el cuadro de texto, pero para introducir textos largos	`<TEXTAREA>`	***rows*** ***cols***
Botón de reset	Cuando se pulse, todos los controles volverán a los valores iniciales	`<input type="reset">`	
Botón de envío	Cuando el usuario lo pulse, se enviarán las parejas nombre/valor de todos los controles, en las condiciones indicadas en el elemento ***FORM***	`<input type="submit">`	
Botón de envío con imagen	Como el botón de envío, pero en lugar de mostrarse como un botón, aparecerá una imagen (señalada con el atributo ***src***). Además, al conjunto de parejas nombre/valor enviadas se añaden las parejas ***x/valor***, ***y/valor*** con las coordenadas x,y donde se ha hecho el click de ratón	`<input type="image" src="URL_imagen">`	***alt***
Botón personalizado	Será un botón en el que podemos definir el rótulo (atributo ***value***) y la acción a realizar cuando el usuario lo pulse (atributo ***onclick***)	`<input type="button">`	***onclick***

Los atributos tienen el siguiente significado:

Atributo	***Se aplica a***	***Significado***
accept	file	Filtra el tipo de archivos. Accept=".odt, .jpg"
autocomplete	Todos	Especifica si se permite el autocompletado
autofocus	Todos	Especifica que el control recibe el foco al cargar la página
checked	casilla de verificación y radiobotón	(booleano) si está presente, el control estará marcado inicialmente al abrirse la página
cols	textarea	anchura (en caracteres) visible
form	Todos	Especifica a qué formulario pertenece el control
maxlength	cuadros de texto y área de texto	número máximo de caracteres que admite
minlength	cuadros de texto y área de texto	número minimo de caracteres que admite
onclick	botón personalizado	script que define la acción a realizar cuando se pulse el botón
placeholder	Todos	Muestra un mensaje en el control
required	Todos	El control es obligatorio que se rellene
rows	textarea	altura (en filas) visible

size	cuadros de texto	anchura inicial (en caracteres) del cuadro
value*	cuadro de texto	valor inicial del control al abrir la página
value*	checkbox, radiobotón, option	valor que se envía si lo selecciona el usuario
value*	botón personalizado	rótulo que se imprime en el botón
name	todos, salvo botones de envío y reset	asigna un nombre al control que será enviado junto con el valor el usuario le dé al control

NOTA: El atributo ***name*** debe establecer un nombre distinto para cada control, excepto para todos los checkbox de un mismo conjunto y para todos los radiobotones de un mismo conjunto. Esto es especialmente importante en los radiobotones porque esta es la forma que el agente de usuario tiene de saber que todos los radiobotones pertenecen al mismo conjunto y que sólo puede estar marcado uno de ellos.

EJEMPLO :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Formularios</title>
    <meta charset="UTF-8">
  </head>
  <body>

    <form action="www.servidor.com" method="get">

      <p>Usuario:
      <input type="text" name="usuario"><br></p>

      <p>Contraseña:
      <input type="password" name="contra"><br></p>

      <p>Hobbies:<br>
      <input type="checkbox" name="afic" value="cine" checked>Cine<br>
      <input type="checkbox" name="afic" value="lect">Lectura<br>
      <input type="checkbox" name="afic" value="viaje">Viajar<br></p>

      <p>Sexo:<br>
      <input type="radio" name="sexo" value="h" checked>Hombre<br>
      <input type="radio" name="sexo" value="m">Mujer<br></p>

      <p>Edad:<br>
      <input type="radio" name="edad" value="1" checked>Menor de
18 años<br>
```

****OJO!*** el atributo ***value*** tiene igual nombre pero 3 funciones distintas según el control en que esté.

```
      <input type="radio" name="edad" value="2">Entre 18 y 65 años<br>
      <input type="radio" name="edad" value="3">Mayor de 65
años<br></p>

      <!-- Botones -->
      <p>
        <input type="button" name="prueba" value="Botón de
prueba" onclick='alert("Ahora se realiza la tarea asignada al
botón de prueba");'>
        <input type="reset">
        <input type="submit">
      </p>

    </form>

  </body>
</html>
```

NOTA: La ***W3C*** recomienda inicializar los conjuntos de checkbox y radiobotones mediante el atributo ***checked***, en aquellos controles que el diseñador de la página estime oportunos, pero siempre en alguno.

NOTA: Obsérvese que todos los checkbox del grupo aficiones tienen el mismo nombre (***name***="afic"), también todos los radiobotones del grupo sexo (***name***="sexo") y todos los radiobotones del grupo edad (***name***="edad").

NOTA: El contenido del atributo ***onclick*** del botón personalizado contiene un script escrito en lenguaje javascript que no será objeto de estudio en este modulo. Nos basta saber que ese script mostrará el mensaje en pantalla y esperará que el usuario pulse el botón aceptar.

1.15.2.- Otros botones

Ya hemos visto una forma de incluir botones en un formulario, mediante el elemento ***INPUT***. Hay una alternativa: el elemento ***BUTTON***. La diferencia fundamental entre unos y otros es que cuando usamos ***BUTTON*** podemos personalizar los botones mucho más, incluyendo texto y/o imágenes en el contenido del elemento (entre <***BUTTON***> y </***BUTTON***>). Obsérvese que eso no era posible con ***INPUT*** que al ser un elemento vacío no lleva nunca etiqueta de cierre.

Por lo demás, el funcionamiento de los botones generados por uno u otro camino es el mismo. En la siguiente tabla se puede ver la equivalencia entre unos y otros.

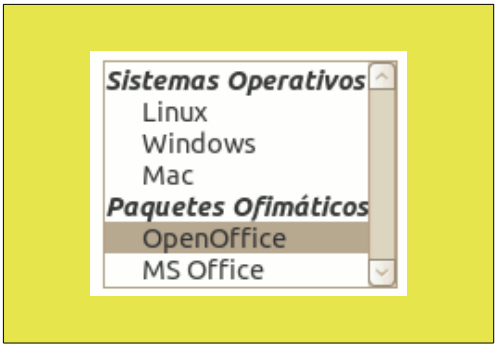
FUNCIÓN	*MEDIANTE INPUT*	*MEDIANTE BUTTON*
Reiniciar	`<input type="reset">`	`<BUTTON type="reset">Limpiar</BUTTON>`

valores		
Enviar valores	`<input type="submit">`	`<BUTTON type="submit">Enviar consulta</BUTTON>`
Enviar valores con imagen ***(*)***	`<input type="image" src="URL_imagen">`	`<BUTTON type="submit"><IMG src="URL_imagen"></BUTTON>`
Botón personalizado	`<input type="button" value="pulsa aquí" onclick="script">`	`<BUTTON type="button" onclick="script">pulsa aquí</BUTTON>`

(*)NOTA: En el caso del envío de valores con imagen hay una diferencia entre ambos botones: con ***INPUT*** se añaden dos parejas nombre/valor, las que indican las coordenadas x,y del punto donde se ha hecho click; con ***BUTTON*** no se envían esos datos.

1.15.3.- Menús

Podemos insertar un menú de opciones en un formulario con el elemento ***SELECT***. Este elemento contiene cada una de las opciones del menú, que se consiguen con el elemento ***OPTION***, y que pueden estar agrupadas en conjuntos de opciones con el elemento ***OPTGROUP***.




```
<select name="menu" multiple size="7">
  <optgroup label="Sistemas Operativos">
    <option value="1">Linux</option>
    <option value="2">Windows</option>
    <option value="3">Mac</option>
  </optgroup>
  <optgroup label="Paquetes Ofimáticos">
    <option value="4" selected>OpenOffice</option>
    <option value="5">MS Office</option>
  </optgroup>
</select>
```

Los atributos que admiten estos elementos son los siguientes:

ELEMENTO	*ATRIBUTOS*	*FUNCIÓN*
select	***name***	Establece un nombre para el control
	multiple	(booleano) si está presente se permite seleccionar varias opciones. Si no lo colocamos, sólo aparecerá la primera opción y mediante un desplegable aparecerá el resto.
	size	(sólo si está el atributo ***multiple***) Número de líneas visibles
optgroup	***label***	Rótulo o título del grupo de opciones
option	***value***	Valor que se envía si el elemento resulta seleccionado
	selected	(booleano) si está presente, la opción estará inicialmente seleccionada

1.15.4.- Agrupación de controles

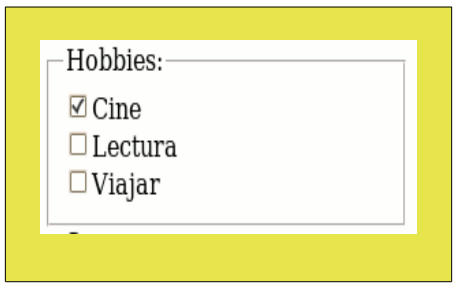
HTML permite agrupar varios controles. Los agentes de usuario normalmente muestran la agrupación dibujando un rectángulo alrededor del grupo.

Formarán parte de un grupo todos los controles contenidos entre ***<FIELDSET>*** y ***</FIELDSET>***. Al principio de un ***FIELDSET*** podemos incluir el elemento ***LEGEND*** para poner un rótulo al grupo que se mostrará encima de él.

EJEMPLO:

```
<fieldset>
  <legend>Hobbies:</legend>
  <input type="checkbox" name="afic" value="cine" checked>Cine<br>
  <input type="checkbox" name="afic" value="lect">Lectura<br>
  <input type="checkbox" name="afic" value="viaje">Viajar<br>
</fieldset>
```

Se visualiza en Mozilla Firefox de la siguiente manera:



1.15.5.- Envío de formularios

La finalidad de un formulario siempre es enviar los datos recabados del usuario a algún sitio. Por eso todo formulario debe incluir un botón de envío (ya sea de tipo ***submit*** o de tipo ***image***). Cuando el usuario pulse ese botón se construirá una cadena de texto con todos las parejas ***nombre/valor*** (nombre de los controles y valor que tienen asignados).

Por ejemplo, si nuestro formulario incluye un cuadro de texto con nombre “usuario" (***name***="usuario") y en esa caja de texto el usuario ha escrito “Fernando", una de las parejas nombre/valor que se envía es: usuario=Fernando.

Cuando un control permite varias selecciones (como un checkbox o un menú con el atributo ***multiple***) aparecen tantas parejas nombre/valor como selecciones haya hecho el usuario. Por ejemplo, si el usuario ha elegido (en el último ejemplo) cine, lectura y viajar, se enviarán las parejas: a***fic=cine&afic=lect&afic=viaje***

NOTA: Observa que las palabras “afic", “cine", “lect" y “viaje" son las que aparecen en el código en los atributos ***name*** y ***value***.

1.15.6.- Foco

Cuando un control acapara el cursor se dice que tiene el foco. Podemos dirigir el foco a un control haciendo click sobre él o pulsando el tabulador sucesivamente hasta llegar a él.

Si no establecemos otra cosa, las pulsaciones sucesivas del tabulador nos llevarán de control en control en el orden de arriba-abajo, izquierda-derecha. Si pulsamos el tabulador con la tecla ***SHIFT*** pulsada el foco irá hacia atrás.

HTML nos permite modificar el orden de asignación del foco con el atributo ***tabindex***, que señala el número de orden para cada uno de los controles. El número de orden puede estar entre 0 y 32.767 y no es necesario que sean consecutivos. Si dos controles tienen el mismo número se seguirá entre ellos el orden por defecto y si un control no tiene el atributo ***tabindex*** cogerá el foco después de todos los demás.

En relación con la asignación de foco, tenemos los siguientes atributos, que pueden usarse en todos los elementos que definen controles:

ATRIBUTO	*VALORES POSIBLES*	*FUNCIÓN*
tabindex	Número entre 0 y 32.767	Número de orden de asignación del foco
autofocus	Booleano	Si aparece, el control recibe el foco al cargar la página
disabled	booleano	Si está este atributo, el control no recibirá el foco
readonly	booleano	Si está este atributo, el control recibe el foco, pero no podrá modificarse el valor

1.15.7.- Etiquetar los controles de formulario

En general, cada control necesita una etiqueta, un rótulo que indique al usuario de que se trata. Por ejemplo si hay un cuadro de texto en el que esperamos que el usuario escriba su nombre, la etiqueta podría ser *"Nombre"* o *"Introduzca su nombre"*. Algunos ya lo incluyen, como los botones (atributo ***value***), para los controles que no lo incluyen podemos usar el elemento ***LABEL***, con el atributo:

- ***for***: nombre del control al que etiqueta. Se puede omitir el atributo ***for***, pero en ese caso, el control deberá estar dentro del contenido de ***LABEL***

EJEMPLOS EQUIVALENTES:

```
Usuario: <input type="text" name="login">
```

```
<label for="login">Usuario:</label>
<input type="text" name="login" id="login">[1]
```

```
<label>Usuario: <input type="text" name="login"></label>
```

Si el usuario hace clic en la etiqueta, el control asociado obtiene el foco de manera inmediata.

1 Tanto ***name*** como ***id*** se usan para dar un nombre a un elemento ***HTML***. Esta duplicidad es un problema heredado de versiones anteriores, que impide que se pueda prescindir de uno de ellos en favor del otro. Cuando en un elemento aparezcan ambos, el valor tiene que ser el mismo.

1.16.- Otros elementos del HTML

Uso de elementos ABBR : Para aclarar el significado de abreviaturas y acrónimos. También ayudará a sintetizadores de voz, correctores y traductores.

```
<ABBR lang="es" title="Do&ntilde;a">Dª</ABBR>
```

1.17.- Vínculos

1.17.1.- Enlaces con el elemento LINK

Pero los vínculos no siempre se usan para "saltar" al documento de destino, pueden tener otros muchos usos:

- Ayuda de navegación
- Ayuda para construir documentos HTML más amplios con coherencia
- Ayuda para los robots de búsqueda

Cuando usamos los vínculos con este objetivo ni siquiera es necesario representarlos en pantalla, es únicamente información estructural que sitúa a nuestro documento en un contexto más amplio. El elemento que nos permite definir un vínculo para esta tarea es ***LINK*** y siempre se sitúa en la cabecera (***HEAD***), a diferencia de ***A*** que siempre se sitúa en el cuerpo.

Nota: Este tipo de vínculo (***LINK***) ofrece información relacional generalizada sobre el documento, que ***usualmente no es mostrada en los navegadores*** (aunque podría ser mostrada, por ejemplo, como información en una barra de herramientas), pero que puede proveer valiosa información para otros intérpretes como los motores de búsqueda.

1.17.2.- Atributos de A y LINK

Puesto que tanto ***A*** como ***LINK*** crean enlaces, ambos disponen de prácticamente los mismos atributos, algunos de los cuales, ya conocemos. Otros de sus atributos son:

- ***type*** indica el tipo de contenido que tenemos en el destino del enlace, de modo que un agente de usuario pueda decidir previamente si lo descarga o no porque no sea un tipo soportado. Los tipos de contenido son los mismos que aparecen en la sección sobre objetos en este mismo capítulo (***text/html***, ***image/png***, ...).
- ***charset*** indica la codificación de caracteres del documento de destino.
- ***hreflang*** indica el idioma en el que está escrito el documento de destino (***es***: español, ***en***: inglés, ***en-us***: inglés americano, ***it***: italiano, ***fr***: francés, ***pt***: portugués, ***nl***: holandés, ***ar***: árabe,...)
- ***title*** contiene una explicación muy breve del documento de destino. Cuando la incluimos en un enlace ***A***, los navegadores mostrarán el valor de este atributo al pasar

el ratón sobre la zona activa. Cuando la incluimos en un enlace ***LINK*** no existe una zona activa, pero puede ayudar a los robots de búsqueda.

- ***rel*** establece la relación que existe entre el documento actual y el documento de destino (***vínculo directo***: dice lo que es el destino respecto del documento actual)

VALORES VÁLIDOS PARA LOS ATRIBUTOS ***rel***	
Alternate	Versión alternativa del documento (en otro idioma o para otro medio)
Stylesheet	Hoja de estilo para aplicar al documento
Start	Primer documento de un conjunto de documentos
Next	Documento siguiente en una serie ordenada de documentos
Prev	Documento anterior en una serie ordenada de documentos
Index	Documento que proporciona un índice para el documento actual
Glossary	Documento que proporciona un glosario de términos del documento actual
Copyright	Documento que contiene el aviso de copyright del documento actual
Apendix	Documento que actúa como apéndice en una colección de documentos
Help	Documento que ofrece ayuda

EJEMPLO:

```
<A href="documento.html" type="text/html" charset="ISO-8859-15"
hreflang="en">Zona activa</A>
```

Es un enlace visible (***A***) al documento ***HTML documento.html***, que está codificado con ***ISO-8859-15*** y contiene texto en ***inglés***.

EJEMPLO:

```
<LINK title="El manual en portugués" type="text/html"
rel="alternate" hreflang="pt" href="http://algunsitio.com/manual/
portugues.html">
```

Crea un enlace con un documento que contiene una ***alternativa*** al documento actual, pero en ***portugués***.

EJEMPLO:

```
<LINK title="El manual en Árabe" type="text/html" rel="alternate"
charset="ISO-8859-6" hreflang="ar"
href="http://algunsitio.com/manual/arabe.html">
```

Crea un enlace con un documento que contiene una ***alternativa*** al documento actual, pero en ***árabe***. Obsérvese que avisa que ese otro documento usa la codificación ***ISO-8859-6***, que debe contener los caracteres del árabe.

EJEMPLO:

```
<LINK lang="fr" title="La documentation en Fran&ccedil;ais"
type="text/html" rel="alternate" hreflang="fr"
href="http://algunsitio.com/manual/frances.html">
```

Crea un enlace con un documento que contiene una ***alternativa*** al documento actual, pero en ***francés***. El atributo ***lang*** informa de que el elemento ***LINK*** (ojo! sólo el elemento ***LINK***) también está en ***francés***.

EJEMPLO:

```
<LINK media="print" title="El manual en postscript"
type="application/postscript" rel="alternate"
href="http://algunsitio.com/manual/postscript.ps">
```

Crea un enlace con un documento que contiene una ***alternativa*** al documento actual, pero en formato ***para imprimir***.

EJEMPLO:

```
<LINK rel="Start" title="La primera página del manual"
type="text/html"
href="http://algunsitio.com/manual/portada.html">
```

Crea un enlace con un documento que es el ***primero*** de una serie de documentos de los que forma parte el documento actual.

EJEMPLO:

```
<head>
  <LINK rel="index" href="index.html">
  <LINK rel="prev" href="doc1.html">
  <LINK rel="next" href="doc3.html">
  <LINK rel="copyright" href="copyright.html">
  <LINK rel="alternate" media="print" href="doc2-printer.html">
  <LINK rel="alternate" lang="en" href="doc-ingles.html">
  <LINK type="text/css" href="basic.css" media="screen">
</head>
```

Suponemos que esta es la cabecera de un documento que ocupa el número 2 en una serie de documentos. Como vemos contiene enlaces a:

- el índice del documento 2
- al documento anterior (1)
- al documento siguiente (3)
- al aviso de copyright sobre el documento 2
- a una versión para medios paginados del documento 2
- a una versión en inglés del documento 2
- a la hoja de estilo con que se visualizará el presente documento 2

EJEMPLOS EQUIVALENTES:

```
<LINK rel="prev" href="doc1.html">
```

El ***doc1.html*** es el anterior al documento actual.

```
<LINK rev="next" href="doc1.html">
```

El documento actual es el siguiente al ***doc1.html***

1.18.- Metadatos

Los metadatos, como su nombre indica, son datos que añaden información sobre el contenido de la página. Están compuestos de una pareja de ***propiedad/valor*** y se crean con elemento ***META***

```
<meta name="propiedad" content="valor">
<meta name="Autor" content="Miguel">
<meta name="descripcion" content="vacaciones idílicas en Grecia">
```

En general no se muestran, pero los agentes de usuario deberían buscar formas de ofrecerlos (en ***Mozilla***: ***Herramientas/Información de la página***).

Algunos metadatos son objeto de atención especial por parte de los motores de búsqueda, por ejemplo los que establecen las palabras clave:

```
<meta name="keywords" lang="es" content="vacaciones, Grecia,
sol">
<meta name="keywords" lang="en" content="holiday, sun">
```

Informar de cómo verse nuestra página.

Con el meta viewport informamos de cómo debe verse en los dispositivos.

```
<meta name=”viewport” content=”width=device-width; initial-scale=1.0” >
```

Cuando pretendemos que los metadatos viajen como parte de la cabecera HTTP en la transmisión del documento cambiamos el atributo ***name*** por ***http-equiv.***

Pero tiene otros usos: establecer el lenguaje de scripts por defecto, el lenguaje de hojas de estilo por defecto o la hoja de estilo por defecto

```
<meta http-equiv="Content-Script-Type" content="text/javascript">
<meta http-equiv="Content-Style-Type" content="text/css">
<meta http-equiv="Default-Style" content="compacto">
```

Establecer refresco:

```
<meta http-equiv="Refresh" content="900">
```

1.19.- Elementos y atributos para hojas de estilo y scripts

1.19.1.- Elementos para uso de hojas

Existen algunos elementos ***HTML*** que tienen una especial utilidad de cara a las hojas de estilo, como ***DIV*** y ***SPAN***. Son elementos de bloque y línea respectivamente que no imponen un significado presentacional. Son ideales para crear estructura artificial que será referenciada desde CSS.

EJEMPLO:

```
<h1>Título</h1>
<div class="resumen">
 <p>---------</p>
 <p>---------</p>
</div>
<p>------
```

Con este elemento ***DIV*** conseguimos que el conjunto que forman los dos primeros párrafos estén recogidos bajo un único elemento.

```
<p>Se ha estrenado <span class="pelicula">Millenium</span> en
TV</p>
```

Con este elemento ***SPAN*** hacemos que una o varias palabras sean recogidas en un elemento ***HTML*** que podrá ser referenciado posteriormente.

Otro elemento importante de cara a las hojas de estilo es LINK que nos permite enlazar el documento a una hoja de estilo y establecer el lenguaje de hojas de estilo por defecto.

EJEMPLO:

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

indica que el documento actual debería visualizarse según las reglas que se encuentran en la hoja de estilo (***stylesheet***) ***hoja.css***, que está descrita con el lenguaje ***CSS***.

EJEMPLO:

```
<meta http-equiv="Content-Style-Type" content="text/css">
```

indica que lenguaje usado para indicar reglas de estilo (incrustadas en el propio documento ***HTML***) será el ***CSS***.

1.19.2.- Atributos para uso de hojas y scripts

También existen atributos HTML que tienen especial interés para las hojas de estilo:

- ***id:*** establece un nombre para el elemento que será reconocido desde la hoja de estilo

```
<p id="miparrafo">----</p>
```

- ***class***: asigna uno o más nombres de clase, para selectores de hojas de estilo. Si hay varios nombres de clase se separan con espacios.

```
<p class="codigo">----</p>
<p class="intro ejemplo">----</p>
```

- ***style:*** aplica reglas de estilo al elemento, directamente como valor del atributo, sin un fichero externo.

```
<p style="text-align:justify">-------</p>
```

1.19.3.- Establecer icono:

```
<link rel="icon" href="favicon.png" type="image/png" >
```

1.20.- Preparado para la web semántica 3.0

Existen una serie de **etiquetas semánticas** en html 5.0 que sirven para dar sentido al contenido que se escribe en la página. No suelen tener efecto visual (se utiliza CSS) pero cada vez son más usadas porque los buscadores las utilizan para entender el contenido de nuestra página y así poder catalogar mejor su contenido.

<***header***> Sirve para indicar que su contenido pertenece a una zona de cabecera. No confundir con <**head**>. Puede haber varios header a nivel de body o estar dentro de otras etiquetas semánticas (pero no dentro de footer, address o header).

<***footer***> Indica que su contenido se sitúa al final de una zona.

<***section***> Permite dividir el contenido en distintas partes o secciones generalmente con un encabezado

<***article***> Indica contenido independiente, contenido que se podrá leer independientemente del resto del sitio web

<***aside***> Indica que su contenido no es parte del artículo sino contenido extra, aclaratorio, etc.

<***nav***> Indica que su contenido son enlaces

<***figure***> y <***figcaption***> se utilizan para las imágenes. ***Figcaption*** añade una explicación a la imagen.

```
<figure>
     <img src=”foto.jpg” alt=”foto”>
     <figcaption>Imagen 2. Foto en el lago</figcaption>
</figure>
```

<***time***> indican que el contenido es una fecha/hora en formato legible para humanos. Puede incluir el atributo ***datatime*** .

```
<time datatime=”19-11-2018 8:15”>Examen Marcas</time>
```

<***main***> Indica que el contenido principal de un documento. Solo debería haber uno y no contener elementos que se repitan en otros documentos como barras de navegación, enlaces de navegación, información de copyright, etc. No afecta a la estructura DOM, es solo informativo.

<***mark***> Indica texto que será resaltado por el navegador debido a su importancia.

<***details***> Indica que el texto lo puede ocultar o mostrar si el usuario quiere. El texto aparece oculto por defecto. Incluye la etiqueta <***summary***> que muestra siempre su contenido. Además con el atributo **open**, podemos forzar a que el texto de details sea visible por defecto

```
<details open>
      <summary>Lista de elementos</summary>
      <p>Elemento 1</p>
      <p>Elemento 2</p>
      <p>Elemento 3</p>
</details>
```