

U.T. IV.- Lenguaje de Marcas de Hipertexto. EL LENGUAJE XHTML

Índice.

[1.- El lenguaje HTML.](#)

[2.- Documentos HTML Bien Formados: XHTML.](#)

[3.- Estructura de un documento XHTML.](#)

[4.- Espacios de nombres \(xmlns\).](#)

[5.- Comentarios.](#)

[6.- Caracteres especiales.](#)

[7.- Los colores en XHTML.](#)

[8.- Etiquetas de edición de texto.](#)

8.1.- Títulos `<h?>`

8.2 - Párrafos `<p>`

8.3.- Saltos de línea `
`.

8.4.- Etiqueta de formato `<pre>`

8.5.- Etiqueta `<div>`.

[9.- Imágenes ``.](#)

[10.- Enlaces \(links\).](#)

10.1.- Links a otra página `<a href>`

10.2.- Anclas a otro lugar de la misma página `<a name>`.

10.3.- Anclas desde otra página.

[**11.- Listas Numeradas o con Viñetas ``, ``**](#)

[12. - Tablas `<table>` `<caption>` `<tr>` `<td>`](#)

12.1.- Encabezados de tabla `<th>`.

1.- El lenguaje HTML.

- El origen de HTML (Hyper Text Markup Language) se remonta a 1980, cuando el físico **Tim Berners-Lee**, trabajador del CERN (*Organización Europea para la Investigación Nuclear*) propone un lenguaje para compartir documentos a través de Internet.
- **Tim Berners-Lee**, junto con el ingeniero de sistemas **Robert Cailliau**, presentaron la propuesta, que denominaron "WorldWideWeb", a una convocatoria organizada para desarrollar un sistema de "hipertexto" para Internet, resultando ganadora.
- El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre de HTML Tags (Etiquetas HTML).
- Posteriormente se van publicando diferentes versiones hasta la última especificación oficial, publicada por el **W3C** (*World Wide Web Consortium*), organismo encargado de la estandarización de los lenguajes creados en la red, el 24 de diciembre de 1999, con el nombre de **HTML 4.01**.
 - o Su especificación oficial se encuentra en la página

[Especificación HTML 4.01](#)

- Desde la publicación de *HTML4.01*, la actividad de estandarización de *HTML* se detuvo y el W3C se centró en el desarrollo del estándar *XHTML*, una versión avanzada de *HTML*, basada en *XML*.
- Debemos tener en cuenta que *HTML5* no es una nueva versión de *HTML*, sino una combinación de *HTML*, *CSS* y *Javascript*. Estas tecnologías actúan como una sola unidad organizada bajo la especificación *HTML5*.

2.- Documentos HTML Bien Formados: XHTML.

- La primera versión de XHTML (eXtensible Hypertext Markup Language), denominada **XHTML 1.0**, se publicó el 26 de Enero de 2000, con posterior revisión el 1 de Agosto de 2002.
- Posteriormente se ha publicado un borrador de una nueva versión, *XHTML 2.0*, que aún no ha sido recomendada por el W3C
- Podemos entender un documento *XHTML* como un documento escrito en *HTML* que cumple las reglas de un documento *XML* bien formado.
 - o En concreto, *XHTML 1.0* es una reformulación de *HTML 4.0* con las especificaciones de *XML 1.0*.
 - o Su especificación oficial se encuentra en la página

Especificación XHTML 1.0

- El estándar *XHTML 1.0* incluye el 95% del estándar *HTML 4.01*, y sólo añade pequeñas mejoras y modificaciones.
- La mejora fundamental de *XHTML* con respecto a *HTML* es que obliga a que los documentos sigan las reglas del *XML* bien formado.

Notas:

- Podemos encontrar una guía de referencia de XHTML 1.0 en español en la página

<http://www.w3c.es/Divulgacion/GuiasReferencia/XHTML1/>

- o No obstante, por motivos de simplicidad y operatividad, utilizaremos como manual de referencia el relativo a *HTML 4.01*, aunque algunas de sus etiquetas y atributos han quedado fuera del estándar *XHTML 1.0*.
- Podemos validar los documentos *XHTML* en la página

3.- Estructura de un documento XHTML.

- Un documento XHTML debe estar compuesto, al menos, por los siguientes elementos, en el orden descrito.

```
<!DOCTYPE ...>
< html xmlns="...">
  < head>
    < title> ... </title>
  < /head>
  < body>
  < /body>
</html>
```

- **Instrucción de procesamiento <!DOCTYPE ...>**
 - o Esta instrucción contiene la DTD (Definición del Tipo de Documento), que describe el conjunto de normas, obligaciones y restricciones que se deben seguir al crear un documento de un determinado tipo.
 - o XHTML se encuentra definido por 2 DTDs:

- **XHTML 1.0 Strict**

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

- Se trata de la variante con las normas más estrictas y las restricciones más severas.
 - Las páginas web que incluyan esta DTD no pueden utilizar atributos relacionados con el aspecto de los contenidos, por lo que requiere una separación total de código HTML y estilo CSS.

- o **XHTML 1.0 Transitional**

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

- Se trata de una variante menos estricta que la anterior, ya que permite el uso de atributos HTML relacionados con el aspecto de los elementos.
- **Etiqueta `<html>`.**
 - o Es la etiqueta raíz. En su interior se definen la cabecera y el cuerpo del documento HTML.
- **Etiqueta `<head>`.**
 - o Contiene "metadatos", es decir, información sobre el propio documento.
 - o Las etiquetas contenidas en `<head>` no son visibles para el usuario, con la excepción de la etiqueta `<title>`, que se utiliza para indicar el título del documento.
 - Los navegadores visualizan el contenido de `<title>` en su parte superior izquierda.
 - o Para indicar al navegador que el juego de caracteres del documento corresponde al idioma español, incluiremos en `<head>` la siguiente etiqueta:

```
<meta http-equiv="content-Type" content="text/html; charset=UTF-8" />
```
 - o Atributo **xmlns**. Debe ser el primer atributo de `<head>`, define el espacio de nombres que va a utilizar el documento.
- **Etiqueta `<body>`.**
 - o En ella deben incluirse todos los contenidos que queramos mostrar al usuario (párrafos de texto, imágenes, tablas, etc).

4.- Espacios de nombres (xmlns).

- Según definición del W3C, Un **espacio de nombres XML** es un conjunto de nombres, como tipos de elemento y nombres/valores de atributo, identificado por una referencia URI, que se utilizan en documentos XML.
- Los espacios de nombres permiten eliminar las ambigüedades y solucionar los problemas de homonimia que se producen en los documentos, ya que en un mismo documento existen palabras con el mismo nombre, pero con diferentes significados.
- Los espacios de nombres se identifican mediante URIs, que no deben verse como enlaces a ningún sitio sino como nombres únicos dentro de un documento XML.

Ejemplo 1.

```
<pais nombre="España">
<capital>Madrid</capital>
<inversión>
<capital>200000</capital>
</inversion>
</pais>
```

Nos encontramos con 2 elementos con la misma etiqueta (<capital>) pero con distinto significado.

La solución sería asignar cada etiqueta a un espacio de nombres, así, si creamos los espacios de nombres <http://www.geograf.es> (conteniendo <pais>, <capital>, nombre="España") y <http://bolsa.com> (conteniendo <inversion>, <capital>) y les asignamos los alias "geograf" y "bolsa":

```
xmlns:geograf= "http://www.geograf.es"
xmlns:bolsa= "http://bolsa.com"
```

Podríamos escribir el párrafo anterior de la siguiente manera:

```
< geograf:pais geograf:nombre="España">
<geograf:capital>Madrid</geograf:capital>
<bolsa:inversion>
<bolsa:capital>200000</bolsa:capital>
</bolsa:inversion>
</geograf:pais>
```

Eliminando de esta forma las ambigüedades.

- El espacio de nombres o namespace que utilizan todas las páginas XHTML es <http://www.w3.org/1999/xhtml>.

5 .- Comentarios.

- Pueden escribirse comentarios en una página XHTML, que no serán tenidos en cuenta por el navegador, encerrándolos entre las marcas `<!--` y `-->`

Ejercicio 401:

Crear con Notepad++ el siguiente documento (Usar el código de caracteres siguiente: *Codificación* → *Juego de caracteres* → *Europeo occidental* → *iso-8859-15*).

```
<?xml version="1.0" encoding="iso-8859-15"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Mi primera página web</title>
</head>
  <body>
    <p>
      <!--Este es mi primer documento HTML-->
      Texto normal. <b>Texto en negrita</b>. Normal
otra vez.</p>
    </body>
  </html>
```

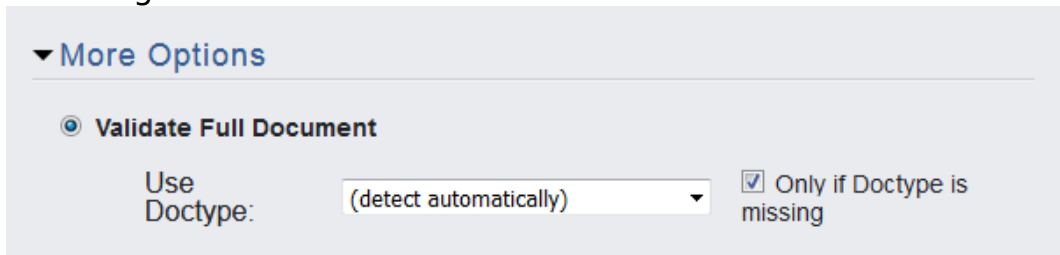
Almacenar el documento con el nombre [ejercicio301.htm](#).

Acceder al servicio de validación del W3C.

Podemos validar el documento escribiendo su dirección (*Validate by [URI](#)*), cargando el fichero (*Validate by [File Upload](#)*) o mediante entrada directa (*Validate by [Direct input](#)*). Elegimos la 3ª pestaña.

Copiamos el contenido del documento en la ventana que se nos muestra.

En [More Options](#) marcamos la casilla **Validate Full Document** del modo siguiente:



▼ More Options

☒ **Validate Full Document**

Use Doctype: (detect automatically) ☒ Only if Doctype is missing

Pulsamos el botón [check](#).

Una vez que el documento sea válido, abrirlo con el navegador Firefox.

6.- Caracteres especiales.

- Existen caracteres que no pueden incluirse directamente en el texto de una página HTML y otros que, aunque se pueden incluir, puede que el navegador no los interprete correctamente, dado que dependerá del código de caracteres que esté utilizando.
- Por ejemplo los acentos, la letra Ñ, los símbolos `<` y `>` y otros muchos.
 - o Para escribir estos caracteres, así como otros que no aparezcan en el teclado, hay que utilizar la tabla de códigos HTML.
 - Páginas 3 y 4 del manual de referencia.
 - o También pueden referenciarse mediante códigos numéricos.
 - Para ver una tabla de códigos numéricos visitar la página
<http://ascii.cl/es/codigos-html.htm>
- Debemos que tener en cuenta que HTML sustituye todos los espacios en blanco por uno sólo, por lo que si queremos que aparezcan varios espacios en blanco tenemos que utilizar el símbolo ` `;

Ejercicio 402.

Crear un documento XHTML que muestre la siguiente frase:

Los caracteres <, > y & han de escribirse como símbolos especiales

- a) Utilizando la tabla de símbolos del manual de referencia.
- b) Utilizando códigos numéricos

7.- LOS COLORES EN XHTML.

- En cualquier monitor un color se construye a partir de unir diferentes proporciones de los 3 colores primarios: rojo, verde y azul.
- En XHTML cada color se expresa mediante un número hexadecimal de 6 cifras precedido del símbolo #.
 - o Las 2 primeras cifras corresponden a la cantidad de rojo, las 2 siguientes a la de verde y las 2 últimas a la de azul (formato RGB ó RVA).
 - o Por ejemplo, el color amarillo, definido como todo el rojo, todo el verde y nada de azul, se expresaría en formato RGB, como #FFFF00.
- 2 cifras en hexadecimal permiten $16 \times 16 = 256$ números distintos, por lo que con este formato pueden identificarse $256 \times 256 \times 256 = 16.777.216$ colores posibles.
- Algunos colores, los más comunes, están identificados con un nombre (página 4 del manual de referencia).
- Existen multitud de tablas con códigos de colores en Internet, como ejemplo, visitar la página

<https://html-color-codes.info/codigos-de-colores-hexadecimales/>

- A pesar de lo anterior, no todas las combinaciones de colores se muestran igual en cualquier navegador. Si queremos asegurarnos de que nuestra página se mostrará siempre con los mismos colores, independientemente del navegador con el que se abra, es aconsejable utilizar únicamente las que se denominan "Combinaciones Seguras" que son las combinaciones de cantidades de color básico que sólo utilizan los valores 00, 33, 66, 99, CC, FF.

- o Para ver una tabla de códigos de colores seguros visitar la página

<http://htmlcolorcodes.com/es/tabla-de-colores/tabla-de-colores-web-seguros/>

- Para establecer el color de fondo y el del texto de un documento debemos incluir en la etiqueta **<body>** los atributos **bgcolor** y **text**, respectivamente.

Ejercicio 403.

Modificar el fondo y el color del texto del ejercicio 402 utilizando colores seguros.

8.- Etiquetas de edición de texto.

8.1.- Títulos **<h?>**

- **<h?>** permite especificar un tamaño de letra para el texto que contiene y provoca un salto de línea.
- **<h1>** indica el mayor tamaño de letra y **<h6>** el menor.

8.2 - Párrafos **<p>**

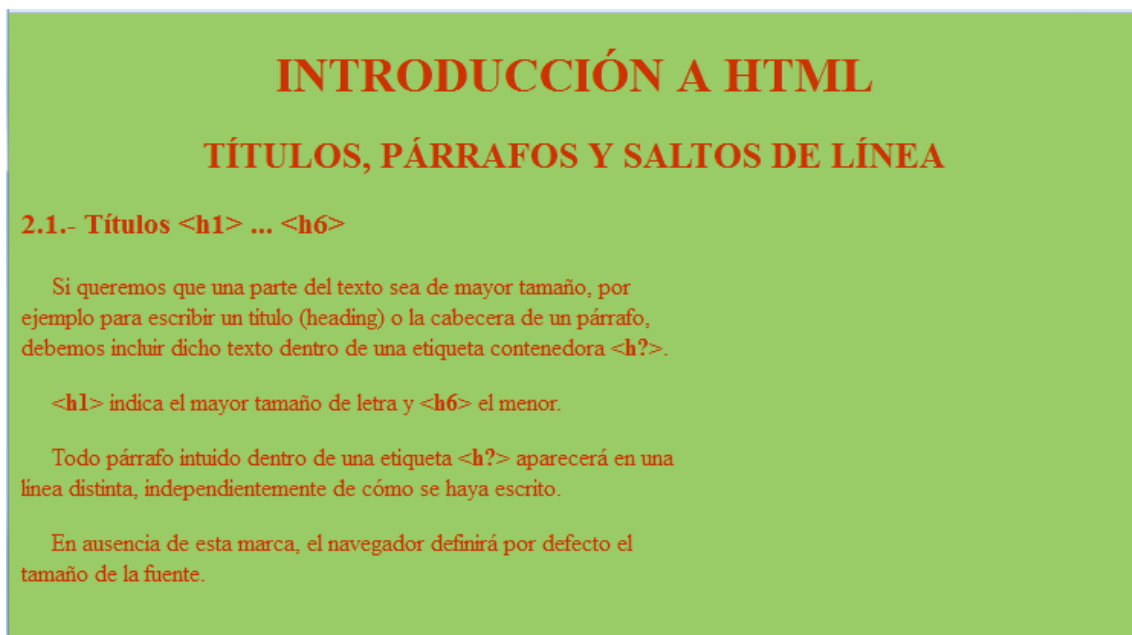
- Todo lo que escribamos dentro de la etiqueta **<p>** (paragraph) aparecerá separado por una línea en blanco con respecto al resto de la página.
- Atributo **align**, indica cómo se alineará el párrafo.
 - Posibles valores: **Left**, **Center** o **Right**

8.3.- Saltos de línea **
**.

- Etiqueta vacía que provoca un salto de línea.
 - Debemos tener en cuenta que los navegadores no reconocen el carácter "ENTER" por lo que escriben todo el texto en una sola línea mientras no se les indique lo contrario.

Ejercicio 404.

Teniendo en cuenta que la etiqueta **** pone en negrita un texto y **<center>** lo centra, crear un documento XHTML que muestre lo siguiente:



INTRODUCCIÓN A HTML

TÍTULOS, PÁRRAFOS Y SALTOS DE LÍNEA

2.1.- Títulos <h1> ... <h6>

Si queremos que una parte del texto sea de mayor tamaño, por ejemplo para escribir un título (heading) o la cabecera de un párrafo, debemos incluir dicho texto dentro de una etiqueta contenedora **<h?>**.

<h1> indica el mayor tamaño de letra y **<h6>** el menor.

Todo párrafo incluido dentro de una etiqueta **<h?>** aparecerá en una línea distinta, independientemente de cómo se haya escrito.

En ausencia de esta marca, el navegador definirá por defecto el tamaño de la fuente.

Modificar el tamaño de la ventana que contiene la página y observar que el formato del texto se ajusta a su tamaño.

Ejercicio 405.

Utilizar la etiqueta **** (página 6 del manual de referencia) para modificar el tamaño, el tipo y el color de letra de los diferentes párrafos del texto del ejercicio 404.

8.4.- Etiqueta de formato **<pre>.**

- Dado que los navegadores no respetan ni varios espacios en blanco (los convierten en uno sólo), ni saltos de línea, ni líneas vacías, podemos obligarle a que respete el formato del texto incluyendo éste dentro de la etiqueta **<pre>**.
- El inconveniente de esta etiqueta es que no se adapta a los cambios de tamaño de la ventana que muestra la página, pudiendo desaparecer parte del texto a mostrar.
- Otro inconveniente es que el estándar XHTML desautoriza la inclusión de la mayoría de las etiquetas de formato dentro de una etiqueta **<pre>**.
 - o Debido a lo anterior esta etiqueta suele utilizarse con un selector de clase que será definido en una hoja de estilo CSS.

Ejercicio 406.

Modificar el ejercicio 404 incluyendo todo el texto en una etiqueta **<pre>**

Modificar el tamaño de la ventana que contiene la página y observar que parte del texto desaparece.

8.5.- Etiqueta **<div>.**

- La etiqueta contenedora **<div>** se utiliza para definir un bloque de contenido o sección de una página web para poder aplicarle un estilo determinado.
- Los atributos que contenga **<div>** afectarán lógicamente a todo su contenido.

- La aplicación del estilo puede hacerse mediante el atributo **style** o, más comúnmente, definiendo un selector de clase que será descrito en una hoja de estilos CSS.
- Su uso se verá más adelante.

9.- Imágenes ****.

- Se utiliza para insertar una imagen dentro de una página.
- Sintaxis:

- Los atributos **src** y **alt** son obligatorios.
 - o Atributo **src** (source): Se le asigna el nombre del fichero que contiene la imagen (.gif, .jpg o .png). En caso necesario debe indicarse la Uri.
 - o Atributo **alt** (alternative) se le asigna un texto que verán los usuarios que visiten el sitio con un navegador que sólo permita texto o que tenga desactivada la opción de descarga de imágenes.
- Esta etiqueta posee varios atributos que permiten formatear la imagen (páginas 11 y 12 del manual de referencia).

Ejercicio 407.

Crear una página HTML para visualizar dos imágenes con pie de foto.

Formatear las imágenes.

10.- Enlaces (links).

10.1.- Links a otra página **<a href>**

- Esta etiqueta define un enlace o hipervínculo a otra página web.
- Cuando el usuario pulsa sobre el espacio ocupado por esta etiqueta el navegador abre la página web en ella especificada.
- Dicho espacio puede contener texto (hipertexto) o una imagen. En ambos casos dicho espacio será sensible al puntero del ratón.
 - o Por defecto, los navegadores señalan un área linkada subrayándola si es texto o poniéndole un borde si es una imagen, ambas de color azul.
- Sintaxis

**** Texto o imagen ****

- o Si la página a cargar se encuentra en un servidor distinto, en el valor que se le asigne al atributo **href** habrá que indicar su Uri correspondiente. (href="http:\\www.nombresitio\\ ...)

Ejercicio 408.

Crear dos páginas HTML, con los títulos ORIGEN y DESTINO, de manera que desde cada una de ellas se pueda acceder a la otra.

Ejercicio 409.

Página que muestre dos imágenes como links.

Al ser presionados deben llamar cada uno a una página distinta.

Dentro de cada una de estas páginas se incluirá un link, también con una imagen, a la página principal.

Ejercicio 410.

Crear una página HTML con el siguiente contenido:

APUNTES SOBRE HTML

<http://www.jmarshall.com/easy/html/spanish/#toc>

Índice:

- 1.Introducción.
2. ¿Qué es HTML?
3. Anatomía de una página Web.

Los 3 puntos contenidos en el índice deben ser enlaces a sendas páginas denominadas **punto1.html**, **punto2.html** y **punto3.html** respectivamente con el contenido que aparece más adelante. Estas páginas deberán tener al final un link a la página principal.

Dar un color de fondo y texto diferentes a cada una de las páginas.

Utilizar la etiqueta **** para modificar el color, tipo y tamaño de letra de los diferentes párrafos.

punto1.html

Introducción.

El HTML es muy fácil de usar; fue diseñado en este sentido.

No tienes que ser un programador para usarlo. Si puedes editar un archivo de texto, entonces puedes escribir HTML.

Si tu navegador soporta marcos (frames) ejecuta esta [Cama de Prueba HTML \(versión sin marcos\)](#), donde puedes escribir HTML en un marco y ver el resultado en otro. Ajusta los marcos de entrada y salida para una mejor vista.

Si tu navegador no soporta marcos o si estás haciendo páginas reales, querrás usar un editor de texto real. Arranca Notepad en Windows, o uno mejor si lo tienes.

Acuérdate de escribir la extensión **".html"** (o **".htm"**) a tus archivos HTML. Usa tu navegador para ver los archivos HTML que vayas creando con el menú **"Archivo/Abrir"** o algo similar; usa la función "Recargar" (**Reload**) después de cada cambio.

punto2.html

¿Qué es HTML?

Aunque HTML significa **Lenguaje de Marcas de HiperTexto**, no es en realidad un lenguaje de programación como Java, Perl, C, o BASIC; es mucho más simple.

Es una manera de describir cómo un conjunto de texto e imágenes deben desplegarse en el navegador, similar en concepto a las marcas que hace un editor de periódico.

punto3.html

Anatomía de una Página Web

Una página Web consiste en un archivo HTML más los archivos de imagen usados en la página.

El archivo HTML (un archivo de texto común) contiene todo el texto a desplegar, y también actúa como el pegamento para sostener el texto e imágenes juntas en los lugares correctos, y los despliega en el estilo correcto.

Escribir un archivo HTML significa componer el texto que quieres desplegar, entonces insertar las **etiquetas** que quieras en los lugares correctos.

Las etiquetas comienzan con un carácter `<` y terminan con un carácter `>`, y le dicen al navegador que haga algo especial como mostrar texto en *itálicas* o **negritas**, o en una fuente más grande, o mostrar una imagen, o hacer un enlace a otra página Web.

Aunque el HTML tiene muchas etiquetas que puedes usar, no necesitas saberlas todas para usar HTML. Puedes arreglártelas con unas cuantas.

Algo excelente sobre el aprendizaje del HTML es que puedes ver cómo otros lo han hecho revisando su código fuente. Puedes ver el código fuente de cualquier página que estés visitando (En Internet Explorer View/Source).

No necesitas un "Editor HTML" para escribir HTML, basta un simple editor de texto.

10.2.- Anclas a otro lugar de la misma página `<a name>`.

- HTML permite establecer una referencia (ancla) dentro de una página para poder posteriormente disponer un hipervínculo a dicho ancla.

- Es una práctica común cuando queremos desplazarnos dentro de una página de gran tamaño: Se disponen hipervínculos a diferentes lugares de la página y estos lugares se marcan mediante referencias o "anclas".
- La sintaxis para definir un ancla a un lugar de la misma página es:

``

- o No debemos confundir un ancla con un link, más allá de que se utiliza la misma marca **`<a>`**.
 - o Para un ancla inicializamos el atributo **name** con un identificador, para un link utilizamos **href** con una dirección.
 - o Un ancla se define en el lugar de la página que queremos mostrar al clicar en el hipervínculo.
- La sintaxis para ir a un ancla desde un hipervínculo es la siguiente:

**`Título
`**

- o En el atributo **href** indicamos el nombre del ancla precedido del carácter #.

Ejercicio 411.

Crear una página con el texto que aparece más abajo.
Deberán establecerse anclas a cada uno de los apartados y, al final de cada apartado, un ancla para volver al comienzo.

Tutorial de MySQL

- 1.- Introducción.
- 2.- show databases
- 3.- Creación de una tabla y mostrar sus campos.

4.- Carga de registros a una tabla y su recuperación.

1.- Introducción.

SQL, Structure Query Language (Lenguaje de Consulta Estructurado) es un lenguaje de programación para trabajar con base de datos relacionales como MySQL, Oracle, etc.

MySQL es un intérprete de SQL, es un servidor de base de datos.

MySQL permite crear base de datos y tablas, insertar datos, modificarlos, eliminarlos, ordenarlos, hacer consultas y realizar muchas operaciones, etc., resumiendo: administrar bases de datos.

Ingresando instrucciones en la línea de comandos o embebidas en un lenguaje como PHP nos comunicamos con el servidor. Cada sentencia debe acabar con punto y coma (;).

La sensibilidad a mayúsculas y minúsculas, es decir, si hace diferencia entre ellas, depende del sistema operativo, Windows no es sensible, pero Linux

sí. Por ejemplo Windows interpreta igualmente las siguientes sentencias:

```
create database administracion;  
Create DataBase administracion;
```

Pero Linux interpretará como un error la segunda.

Se recomienda usar siempre minúsculas. Es más el sitio mysqlia.com.ar

está instalado sobre un servidor Linux por lo que en todos los ejercicios

deberán respetarse mayúsculas y minúsculas.

2.- Show databases

Una base de datos es un conjunto de tablas.

Una base de datos tiene un nombre con el cual accederemos a ella.

Vamos a trabajar en una base de datos ya creada en el sitio, llamada "administracion".

Para que el servidor nos muestre las bases de datos existentes, se lo solicitamos enviando la instrucción:

show databases;

Nos mostrará los nombres de las bases de datos, debe aparecer en este sitio "administracion".

3.- Creación de una tabla y mostrar sus campos

Una base de datos almacena sus datos en tablas.

Una tabla es una estructura de datos que organiza los datos en columnas y filas; cada columna es un campo (o atributo) y cada fila, un registro. La intersección de una columna con una fila contiene un dato específico, un solo valor.

Cada registro contiene un dato por cada columna de la tabla.

Cada campo (columna) debe tener un nombre. El nombre del campo hace referencia a la información que almacenará.

Cada campo (columna) también debe definir el tipo de dato que almacenará.

4.- Carga de registros a una tabla y su recuperación

Usamos "insert into". Especificamos los nombres de los campos entre paréntesis y separados por comas y luego los valores para cada campo, también entre paréntesis y separados por comas.

Es importante ingresar los valores en el mismo orden en que se nombran los campos, si ingresamos los datos en otro orden, no aparece un mensaje de error y los datos se guardan de modo incorrecto.

Note que los datos ingresados, como corresponden a campos de cadenas de caracteres, se colocan entre comillas simples. Las comillas simples son OBLIGATORIAS.

10.3.- Anclas desde otra página.

- Si queremos ir desde una página a un sitio concreto de otra página debemos utilizar la siguiente sintaxis:

`Título`

- o Donde *pagina2.htm* es el nombre de la página que queremos visitar y *#nombreancla* es el identificador del ancla de pagina2.htm al que queremos ir.

Ejercicio 412.

Con el mismo texto del ejercicio 411, crear una página web que contenga el índice y otra el contenido.

Desde la página índice deberá poderse acceder a cada uno de los apartados de la página contenido.

Asimismo, debe poder volverse a la página índice desde el final de cada uno de los apartados.

11.- Listas Numeradas o con Viñetas ``, ``

- HTML proporciona una manera simple de mostrar listas numeradas ("listas ordenadas") o con viñetas ("listas desordenadas") mediante las etiquetas `` y `` respectivamente.
- Sintaxis de lista ordenada|con viñetas:

```
<ol>|<ul>
  <li>elemento1</li>
  <li>elemento2</li>
  <li>elemento3</li>
  ...
</ol>|</ul>
```

- Los atributos que pueden utilizarse en las etiquetas `` y `` se encuentran en las páginas 8 y 9 del manual de referencia.

Ejercicio 413.

Crear una página web con el siguiente texto:

Esta es una lista ordenada:

1. Primer elemento
2. Segundo elemento
3. Tercer elemento

Esta es una lista desordenada:

- Primer elemento
 - Segundo elemento
 - Tercer elemento
- Los elementos de una lista, pueden contener links, anclas, imágenes, tablas o incluso otras listas.
 - o En este último caso estaremos hablando de "listas anidadas".
 - o Las listas anidadas son bastante comunes, útiles para resúmenes o menús en cascada.

Ejercicio 414.

Crear una página con el siguiente contenido:

LISTA ANIDADA CON VIÑETAS

- Ficheros HTML
 - o Ejemplos
 - Listas
 - Tablas
 - Estilos
 - Cita
 - Texto en énfasis
 - o En curso
 - o Prescritos
- Ficheros de imágenes

Ejercicio 415.

Crear una página web con el mismo texto del ejercicio 414 con listas anidadas numeradas.

- También pueden combinarse listas numeradas con listas con viñetas.

Ejercicio 416 .

Crear una página web con el siguiente contenido:

LISTA ANIDADA NUMERADA Y CON VIÑETAS

- Ficheros HTML
 1. Ficheros de prueba
 - [prueba1](#)
 - [prueba2](#)
 - [prueba3](#)
 2. Ficheros de ejemplo
 3. Ficheros del servidor
- Ficheros de imágenes
- Ficheros de sonidos
- Ficheros de vídeo

[prueba1](#), [prueba2](#) y [prueba3](#) deben ser links a sendas páginas xhtml y desde ellas debe poder volverse a la página principal.

12. - Tablas `<table>` `<caption>` `<tr>` `<td>`

- Una tabla muestra un conjunto de datos, en forma ordenada, organizados en filas y columnas.
- Sintaxis

```
<table border="num">
<caption> Título de la tabla </caption>
<tr>
    <td>dato_fila1_columna1</td>
    <td>dato_fila1_columna2</td>
    <td>dato_fila1_columna3</td>
    ...
</tr>
<tr>
    <td>dato_fila2_columna1</td>
    <td>dato_fila2_columna2</td>
    <td>dato_fila2_columna3</td>
    ...
</tr>
...
</table>
```

- Donde:
 - o `<table>` Etiqueta de tabla:
 - Atributo **border**. Permite especificar el grosor del borde mediante un número entero.

- o **<caption>** Su contenido aparecerá como título de la tabla. Debe escribirse inmediatamente después de **<table>**.
- o **<tr>** Etiqueta de fila.
- o **<td>** Etiqueta de celda.
- Los atributos que pueden incluirse en las etiquetas **<table>**, **<tr>** y **<td>** se encuentran en la página 12 y siguientes del manual de referencia.
 - o Lógicamente, los atributos incluidos en **<table>** afectarán a todas las celdas de la tabla, los incluidos en **<tr>** a todas las celdas de esa fila y los incluidos en **<td>** a esa celda.
 - o Si un mismo atributo está incluido en más de una etiqueta prevalece el más interno.
- Debe definirse el mismo número de celdas en cada fila, aunque algunas no tengan contenido.
 - o Si se definen filas con distinto número de celdas el resultado dependerá del navegador (o versión) utilizados.

Ejercicio 417.

- a) Crear una página HTML que muestre la siguiente tabla:

TABLA DE DATOS

Dato11	Dato12	Dato13	Dato14
Dato21	Dato22	Dato23	
Dato31	Dato32		
Dato41			

- b) Utilizando el menor número de atributos **bgcolor** posible, modificarla para que muestre el siguiente aspecto:

TABLA DE DATOS

Dato11	Dato12	Dato13	Dato14
Dato21	Dato22	Dato23	
Dato31	Dato32		
Dato41			

Ejercicio 418.

Documento XHTML que muestre la siguiente tabla

NOVIEMBRE 2017

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	SÁBADO	DOMINGO
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

Ejercicio 419:

Modificar la tabla creada en el ejercicio 417 para que muestre la siguiente forma:

Dato11	Dato12	Dato13	Dato14
Dato21	Dato22	Dato23	
Dato31	Dato32		
Dato41			

12.1.- Encabezados de tabla <th>.

- Si queremos que una celda adopte la forma de "Encabezado", es decir, con el texto centrado y en negrita, debemos sustituir la etiqueta **<td>** por **<th>**.
 - o Lo anterior puede hacerse para cualquier celda.

Ejercicio 420.

Utilizando etiquetas de encabezado, documento XHTML que muestre la siguiente página:

Pedido

Nombre producto	Precio unitario	Unidades	Subtotal
Reproductor MP3 (80 GB)	192.02	1	192.02
Fundas de colores	2.50	5	12.50
Reproductor de radio & control remoto	12.99	1	12.99
TOTAL		7	207.51