

Ejercitación 1:

1- HTML (HyperText Markup Language) es un lenguaje de marcado para la elaboración de páginas web. Se considera el lenguaje web mas importante del World Wide Web Consortium siendo su invención crucial en la aparición, desarrollo y expansión de la www.

VERSIONES:

- HTML 2.0 (1995)
- W3C (1996)
- HTML 3.2 (1997)
- HTML 4.0 (1998)
- HTML 4.0.1 (1999)
- HTML 5.0 (2008)

2- Principios de W3C para la creación de documentos con HTML

- **Perceptible:** tanto los contenidos como la interfaz de un sitio web debe poder ser percibida por todos sus usuarios.
- **Operable:** un sitio web debe tener muchas maneras, y todas muy claras y advertidas, para realizar una acción o buscar un contenido. Entre más alternativas existan, mejor será su accesibilidad.
- **Comprensible:** un sitio web debe ser: legible, comprensible y predecible
- **Robusto:** los sitios web o aplicaciones deben ser compatibles con todos los navegadores web (Internet Explorer, Mozilla, Chrome, Safari, entre otros), en todos los sistemas operativos (Windows, Mac, Android, iOS), y en todos los dispositivos (computadoras, teléfonos móviles, tabletas, e incluso televisores)

3- **Un atributo es desaprobado** cuando ha quedado anticuado por la presencia de estructuras nuevas. Los elementos desaprobados se definen en el manual de referencia en los lugares apropiados, pero claramente marcados como desaprobados. Los elementos desaprobados pueden declararse obsoletos en versiones futuras de HTML.

Un atributo es obsoleto cuando para el cual no hay garantía de soporte por parte de un agente de usuario.

4- Una DTD es un documento SGML que incluye las reglas sintácticas para un tipo de documento específico, incluye los elementos que se permiten y sus atributos, así como las reglas que afectan a la anidación de los primeros y a los valores de los segundos, contrastando un documentos con su DTD se puede comprobar si este es válido o no.

Por ejemplo:

- `<!ELEMENT PARAM - O EMPTY -- valor de propiedad con nombre -->`
- `<!ENTITY % fontstyle "TT | I | B | BIG | SMALL">`

Su función principal es aclarar qué versión de html estás usando, va en la primera línea del código. En html 5 estas ya no se utilizan mas y lo único que se solicita es lo siguiente.

- `<!DOCTYPE html>`

La palabra clave `<!ELEMENT` comienza una declaración y el carácter `>` la termina. Entre ellos se especifica:

- El nombre del elemento.

- Si las etiquetas del elemento son opcionales. Si aparecen dos guiones después del nombre del elemento tanto la etiqueta inicial como la final son obligatorias. Un guión seguido por la letra "O" indica que puede omitirse la etiqueta final. Un par de letras "O" indican que tanto la etiqueta inicial como la final pueden omitirse.
- Los contenidos del elemento, si los hay. El contenido admitido para un elemento se denomina su modelo de contenido. Los tipos de elementos diseñados de modo que no tengan contenido se llaman elementos vacíos. El modelo de contenido de tales tipos de elementos se declara con la palabra clave "EMPTY".

TDT's Contemplados en la especificación de HTML 4.01:

- **Estricto:** no permite el uso de ningún etiquetado de presentación visual o elementos obsoletos o conjuntos de marcos, valida el estilo de etiquetado HTML más libre, como los atributos minimizados y los atributos sin comillas.
- **Transicional:** Permite algunos elementos de etiquetado de presentación visual y elementos obsoletos, pero no los conjuntos de marcos.

5- Los metadatos son elementos html los cuales muestran información sobre la propia página web que los contiene. Son usados por los buscadores para definir la información principal de nuestra web (temática y descripción). Estos elementos se encontraran en el head del código.

La especificación de estos metadatos en la vista de un código es:

```
<meta name="nombre_del_elemento"
content="contenido_del_elemento"/>
```

Ejercitación 2

A- <!-- Código controlado el día 12/08/2009 →

B- <div id="bloque1">Contenido del bloque 1</div>

Se utiliza para crear secciones o agrupar contenidos. Estos se ubican en el body de nuestro sitio y en este caso la identificación o lo que le da un nombre al elemento que lo diferencia de todos los demás del documento es el id que en este caso el "bloque 1", dentro de las etiquetas div va el contenido que en este caso solamente es "Contenido del bloque 1". Por lo que en este ejemplo lo único que se define es un bloque de identificación bloque1, y de contenido "Contenido del bloque 1".

C-

Este es un elemento para insertar una imagen en el sitio, la etiqueta es img y como atributos está el src que en este caso está vacío, el cual indica dónde se encuentra alojada la imagen que queremos mostrar, el atributo alt el cual proporciona un texto alternativo para mostrar en su lugar en caso de que la imagen no se pueda mostrar, el atributo id el cual identifica a esa imagen en el documento, los atributos width y height para darle altura y anchura a la

imagen y por último el atributo longdesc que se utiliza como complemento al atributo alt para especificar un vínculo a una descripción larga. Esta se encuentra definida en el body ya que ahí se suelen ubicar las imágenes.

D- <meta name="keywords" lang="es" content="casa, compra, venta, alquiler"/>
<meta http-equiv="expires" content="16-Sep-2019 7:49 PM"/>

Esta etiqueta meta sirve para aportar información sobre el documento, se define en el head del html. Su atributo name identifica un nombre de propiedad o nombre al que se asocia la metainformación, lang define el lenguaje en el que va a estar escrito el contenido, el atributo content define los datos que se van a asociar a esa name, el http-equiv aporta información sobre los encabezado de respuesta HTTP y puede usarse en lugar de name.

E- Resumen HTML

La etiqueta a se utiliza para indicar la URL a la que apunta un enlace, en este caso el enlace va en el atributo href, el atributo type especifica el tipo de medio para la URL enlazada, hreflang el lenguaje del enlace, el atributo charset define la codificación de caracteres de la URL enlazada y el atributo rel especifica la relación del objeto de destino con el objeto de enlace. Esta etiqueta suele ir en el body del sitio.

F-

```
<table width="200" summary="Datos correspondientes al ejercicio vencido">
  <caption align="top"> Titulo </caption>
  <tr>
    <th scope="col">&nbsp;</th>
    <th scope="col">A</th>
    <th scope="col">B</th>
    <th scope="col">C</th>
  </tr>
  <tr>
    <th scope="row">1</th>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
  <tr>
    <th scope="row">2</th>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
  </tr>
</table>
```

La etiqueta table define una tabla para nuestro sitio, la cual va a estar ubicada en el body de la misma. Los atributos utilizados en esta son:

- width: define la anchura de la tabla en pixeles o porcentaje.
- summary: se utiliza para dar una descripción lo más precisa posible del contenido o tema tratado en nuestra tabla.
- caption: es al atributo de alineación de los elementos en la tabla, en este caso es hacia arriba (top) y lo que se va a alinear es el "Título"

- tr: este atributo especifica la alineación vertical del texto dentro de cada fila de las celdas de la cabecera de una tabla.
- th: representa a una celda de encabezado en una tabla, están pensadas para proveer información de encabezado para las celdas de los datos. Lo que hacen en el código es darle los valores que van a estar en cada celda de la tabla.

Ejercitación 3

A- Dentro de las diferencias de estas visualizaciones podemos diferenciar varios aspectos, primero que estos hipervínculos no todos hacen lo mismo, y si lo hacen, lo hacen de manera diferente.

Los dos primeros, su funcionalidad es la misma, ambos son hipervínculos en cuando los apretas te redirige a la página de google, la única diferencia es el atributo target en el segundo caso, pero ambas realizan la misma funcionalidad.

La tercera línea, no tiene visualización, cuando se la compara con las demás líneas lo único que agrega son los atributos type, hreflang charset y rel, pero la funcionalidad cambia relativamente.

El 4º hipervínculo se visualiza de la misma manera dentro del sitio que los primeros dos, pero cuando se selecciona no renderiza a ninguna página, esto se debe que le referencia lo hace a #, lo que implica que no se va a redirigir a ningún otro sitio al seleccionarlo.

La 5ª línea es muy similar a la 4ª lo único es que se visualiza otro texto en el hipervínculo y la referencia es otra, pero la funcionalidad es la misma, ya que no es una url válida para renderizar a otro sitio web, por lo que permanecerá en el mismo.

La última línea es la más se diferencia de los anteriores ya que el atributo name permite crear enlaces vacíos que hacen referencia a secciones dentro de la misma página, en este caso no se visualiza.

B- En este caso podemos visualizar 4 líneas de código las cuales son imágenes que hacen referencia al sitio de google. Las primeras dos son idénticas pero en distinto orden la primera tiene a la imagen antes del hipervínculo y la segunda al revés. La 3ª logra un solo hipervínculo uniendo a la imagen con el "click aquí" por lo que si apretas una o la otra va a ser apretar lo mismo. Y en la última línea la diferencia es que son dos hipervínculos separados donde la primera es la imagen la cual redirige y la segunda es el click aquí que también redirige.

C- En estas líneas vemos diferentes maneras de implementar las listas, la primera es con la etiqueta ul la cual representa una lista no ordenada, por lo que va a mostrar los elementos de la misma con un punto o asterisco para listarlos. La segunda ya lo realiza con la etiqueta ol, la cual ordena a la lista, en este caso del 1 al 3. La tercera también se realiza con la etiqueta ol, pero separa las etiquetas y le pone valores a las etiquetas internas li, por lo que la visualización es similar a la anterior pero hay un salto de línea entre ellas provocado por la separación de etiquetas. Y en la última visualizamos otra etiqueta, blockquote la cual permite a los autores insertar citas en forma de bloques de contenido lo que va a provocar una indentación en la visualización.

D- En este caso podemos ver dos tablas las cuales se visualizan de la misma manera en el sitio web, la única diferencia en cuanto al código yace en como se definen las columnas, en

la primera tabla se utiliza la etiqueta `th` solamente, en la segunda se utiliza la etiqueta `td` lo que conlleva que se deba hacer una alineación aparte y que se tenga que añadir la negrita manualmente con las etiquetas `div` y `strong`, por lo que se concluye que la etiqueta `th` es mas eficiente para este caso.

E- Estas dos tablas tambien se visualizan de igual manera, dos tablas con un titulo y tres columnas con dos filas con solo un fondo gris, la única diferencia en el código es la manera de implementar el titulo, la primera lo realiza con la etiqueta `caption` y la segunda con la etiqueta `td colspan` cuyo numero indica la cantidad de celdas que extiende a lo ancho y la cual tambien hay que alinearla todo esto `caption` lo realiza de manera automática.

F- Estas dos tablas estan realizadas con las mismas etiquetas que las anteriores con la diferencia que en este caso se utiliza la etiqueta `td rowspan = 2` en la primera tabla para unir las dos filas de la primera columna y en la segunda se utiliza la etiqueta `td colspan = 2` para unir las primeras dos columnas de la primer fila.

G- En este par de tablas se observan en lo que respecta al modelo idénticas, lo que cambia la segunda de la primera es el espacio que hay entre sus bordes internos, esto lo logra agregando los atributos `cellpadding = 0` y `cellspacing=0`. Por todo lo demás son iguales.

H- En este punto tenemos tres ejemplos de logins, las estructuras de los tres son idénticas, ya que todas utilizan la etiqueta `form`, pero la funcionalidad es distinta. El primer login tiene los campos usuario y clave para completar, la particularidad es que ya tienen un valor por defecto xxx, en ambos campos. El segundo es igual que el primero con la excepción de que no tiene precargado valores por defecto. Estos dos primeros tienen algo en común, y es que una vez que se oprime enviar, se redirige a otra pestaña con el mismo código, no así ocurre con el tercero, esto se debe a que el atributo `type` de la etiqueta `input` tiene como valor `reset` y no `submit` como las dos anteriores.

I- En este caso tenemos dos estilos de botones simples, el primero con el objetivo de tener una imagen como un botón y la segunda solamente un botón común con el valor `click` aquí. El primero utiliza la etiqueta `button` y el segundo utiliza la etiqueta `input` con `button` como valor del atributo `type`.

J- Aquí tenemos dos ejemplos de inputs de tipo radio, ambos son iguales y tienen la misma funcionalidad, la única diferencia entre ellos está en el código, y es el nombre que denomina a las opciones. En el primero los dos se llaman opción y en el segundo los nombres son `opcion1` y `opcion2`.

K- En este último caso se muestran dos listas de selección, en la primera esta es desplegable y en la segunda no lo es. Ambas tienen dos casos con Mayo y Junio. En el primero solo se puede seleccionar una opción y en la segunda se pueden seleccionar ambos meses ya que utiliza el atributo `multiple` que en el otro caso no lo hacía dentro de la etiqueta `select`.