**Ejercitación 1 - Responder el siguiente cuestionario**

1. Qué es HTML, cuando fue creado, cuáles fueron las distintas versiones y cuál es la última?

2. ¿Cuáles son los principios básicos que el W3C recomienda seguir para la creación de documentos

con HTML?

3. En las Especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿y

obsoleto?

4. Qué es el DTD y cuáles son los posibles DTDs contemplados en la especificación de HTML 4.01?

5. Qué son los metadatos y cómo se especifican en HTML?

1- **HTML** es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la siglas que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto.

Es el lenguaje de publicación usado por la World Wide Web que todas las computadoras pueden comprender potencialmente.

*El HTML da a los autores las herramientas para:*

* *Publicar documentos en línea con encabezados, textos, tablas, listas, fotos, etc.*
* *Obtener información en línea a través de vínculos de hipertexto, haciendo clic con el botón de un ratón.*
* *Diseñar formularios para realizar transacciones con servicios remotos, para buscar información, hacer reservas, pedir productos, etc.*
* *Incluir hojas de cálculo, videoclips, sonidos, y otras aplicaciones directamente en sus documentos.*

El HTML fue desarrollado originalmente por **Tim Berners-Lee** mientras estaba en el CERN, y fue popularizado por el navegador Mosaic desarrollado en el NCSA. Durante los años 90 ha proliferado con el crecimiento explosivo de la Web. Durante este tiempo, el HTML se ha desarrollado de diferentes maneras. La Web depende de que los autores de páginas Web y las compañías compartan las mismas convenciones de HTML. Esto ha motivado el trabajo colectivo en las especificaciones del HTML.

Debido al rápido crecimiento de la web, surgió la necesidad de crear un estándar para que tanto los programadores como los navegadores pudieran basarse en unas mismas normas para escribir HTML. Cada versión de HTML establece unas normas respecto a cuáles son las etiquetas válidas y cómo se deben escribir.

**Los estándares oficiales HTML** son el HTML 2.0, el HTML 3.2, el HTML 4.0, el HTML 4.01 y el HTML 5. **El HTLM 5 es la última especificación oficial** y se espera que continúe evolucionando a lo largo de los próximos años. El XHTML, una forma más avanzada del HTML que se suponía iba a sustituir a éste, podemos considerar que ha quedado integrado dentro del HTML 5.

2- Principios básicos que la W3C recomienda seguir para la creación de documentos:

### **Separar estructura y presentación**

El HTML tiene sus raíces en SGML, que siempre ha sido un lenguaje para la especificación de código estructural. A medida que el HTML madura, un número cada vez mayor de sus elementos y atributos presentacionales ha sido reemplazado por otros mecanismos, en particular las hojas de estilo. La experiencia ha demostrado que separando la estructura de un documento de sus aspectos presentacionales se reduce el coste de servir a un amplio espectro de plataformas, medios, etc. y se facilitan las revisiones del documento.

### **Considerar la accesibilidad universal a la Web**

Para hacer la Web más accesible a todos, en especial a aquéllos con discapacidades, los autores deberían considerar cómo pueden representarse sus documentos en diferentes plataformas: navegadores basados en voz, lectores braille, etc. No estamos recomendando a los autores que limiten su creatividad, sólo que consideren representaciones alternativas de sus diseños. El HTML ofrece un número de mecanismos con este fin.

Además de esto, los autores deberían recordar que sus documentos pueden llegar a una audiencia muy lejana con diferentes computadoras y configuraciones. Para que los documentos sean correctamente interpretados, los autores deberían incluir en sus documentos información sobre el idioma natural y la dirección del texto, cómo está codificado el documento, y otras cuestiones relacionadas con la internacionalización.

### **Ayudar a los agentes de usuario con la representación incremental**

Mediante un diseño cuidadoso de las tablas y haciendo uso de las nuevas características de las tablas de HTML 4, los autores pueden ayudar a los agentes de usuario a representar los documentos más rápidamente. Los autores pueden aprender a diseñar tablas para su representación incremental. Los implementadores deberían consultar las [notas sobre tablas](http://html.conclase.net/w3c/html401-es/appendix/notes.html#notes-tables) del apéndice para obtener información sobre algoritmos incrementales.

3- En las especificaciones de HTML:

Un elemento o atributo **desaprobado** es aquel que 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.

Los agentes de usuario deberían seguir dando soporte a los elementos desaprobados por razones de compatibilidad con versiones anteriores.

Un elemento o atributo **obsoleto** es aquél para el cual no hay garantía de soporte por parte de un agente de usuario. Los elementos obsoletos han dejado de estar definidos en la especificación, pero se enumeran por motivos históricos en la sección de cambios del manual de referencia.

4- El **SGML** es un sistema para definir lenguajes para dar formato a documentos (markup languages). Los autores utilizan un código de formato (en inglés markup) en sus documentos para representar información estructural, presentacional y semántica junto con el contenido. El HTML es un ejemplo de lenguaje de formato de documentos.

Cada lenguaje de formato de documentos definido con SGML se llama aplicación SGML. Una aplicación SGML se caracteriza generalmente por:

* Una declaración SGML. La declaración SGML especifica qué caracteres y delimitadores pueden aparecer en la aplicación.
* **Una definición del tipo de documento (document type definition, DTD)**. El DTD define la sintaxis de las estructuras de formato. El DTD puede incluir definiciones adicionales, tales como referencias a entidades de caracteres.
* Una especificación que describe la semántica que se debe conferir al código de formato. Esta especificación también impone restricciones de sintaxis que no pueden expresarse dentro del DTD.
* Documentos que contienen datos (contenido) y código (markup). Cada documento contiene una referencia al DTD que debe usarse para interpretarlo.

**HTML 4.01 especifica tres DTDs**, de modo que los autores deben incluir una de las siguientes declaraciones del tipo de documento en sus documentos. Los DTDs varían en cuanto a los elementos que soportan.

1. **El DTD HTML 4.01 Estricto (Strict DTD)** incluye todos los elementos y atributos que no han sido desaprobados o que no aparecen en documentos con marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"

"http://www.w3.org/TR/html4/strict.dtd">

1. **El DTD HTML 4.01 Transicional (Transitional DTD)** incluye todo lo que incluye el DTD estricto más los elementos y atributos desaprobados (la mayoría de los cuales están relacionados con la presentación visual). Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"

"http://www.w3.org/TR/html4/loose.dtd">

1. **El DTD HTML 4.01 para Documentos con Marcos (Frameset DTD)** incluye todo lo que incluye el DTD Transicional más los marcos. Para los documentos que usen este DTD, utilice esta declaración del tipo de documento:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"

"<http://www.w3.org/TR/html4/frameset.dtd>">

5- **Metadatos:**

El HTML permite a los autores especificar metadatos -- información sobre un documento más que contenido del propio documento -- de diferentes de maneras.

#### **Especificación de metadatos:**

En general, la especificación de metadatos implica dos pasos:

1. Declaración de una propiedad y de un valor para esta propiedad. Esto puede hacerse de dos maneras:
   1. Desde dentro de un documento, por medio del elemento [META](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-META).
   2. Desde fuera de un documento, vinculando los metadatos por medio del elemento [LINK](http://html.conclase.net/w3c/html401-es/struct/links.html#edef-LINK) (véase la sección sobre [tipos de vínculos](http://html.conclase.net/w3c/html401-es/types.html#type-links)).
2. Referencia a un [perfil](http://html.conclase.net/w3c/html401-es/struct/global.html#profiles) en el que se definen la propiedad y sus valores legales. Para designar un perfil, se usa el atributo [profile](http://html.conclase.net/w3c/html401-es/struct/global.html#adef-profile) del elemento [HEAD](http://html.conclase.net/w3c/html401-es/struct/global.html#edef-HEAD).

**Ejercitación 2**

HTML está compuesto por un conjunto de elementos que son la base de su estructura. Los elementos están compuestos por dos tags (el de apertura y el de cierre) y el contenido en el medio (con excepción de los elementos vacíos ). Cada tag puede tener atributos (proporcionan ciertas características como altura, ancho, color, etc.) y eventos (asocian un script que se ejecuta cuando el evento ocurre).

*Analizar los siguientes segmentos de código indicando en qué sección del documento HTML se colocan, cuál es el efecto que producen y señalar cada uno de los elementos, etiquetas, y atributos (nombre y valor), aclarando si es obligatorio.*

**2.a)**

\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<!--Código controlado el día 12/08/2009 —>

Apertura/Cierre de declaración de etiqueta = Obligatorio

Contenido = No Obligatorio

Este segmento de código es un comentario. Se puede colocar en cualquier sección del documento HTML.

No se permite espacio en blanco entre el delimitador de apertura de declaración de etiqueta ("<!") y el delimitador de apertura de comentario ("--"), pero sí se permite entre el delimitador de cierre de comentario ("--" y el delimitador de cierre de declaración de etiqueta (">").

**2.b)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<div id="bloque1">Contenido del bloque1</div>

Apertura/Cierre de declaración de etiqueta = Obligatorio

Atributo = No Obligatorio

Valor = No Obligatorio

Contenido = No Obligatorio

Elemento = Obligatorio

Este segmento de código es un div, es un contenedor genérico. Se utiliza con propósitos estilísticos en conjunto con los atributos class y style .

Debe usarse como último recurso, cuando no puedan usarse otras etiquetas y debe colocarse dentro de las etiquetas <body> </body> .

**2.c)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

longdesc="detalles.htm" />

Apertura/Cierre de declaración de etiqueta = Obligatorio

Atributo = No Obligatorio

Valor = No Obligatorio

Elemento = Obligatorio

El elemento img representa una imagen, es un recurso externo que puede incrustarse en el documento HTML. Debe colocarse dentro de las etiquetas <body> </body> .

En el atributo src -> Debe especificarse la ubicación del recurso.

El atributo alt -> Está pensado para proveer una versión alternativa del contenido o propósito de la imagen para aquellas situaciones en las que éstas no pueden ser accedidas apropiadamente (usuarios con discapacidades visuales, navegadores sin soporte,etc).

El atributo name -> Desaprobado HTML4.01, Obsoleto HTML5. Se refiere al nombre para el elemento. Soportado en HTML 4 sólo para compatibilidad con versiones anteriores. En su lugar, usa el atributo id.

El atributo longdesc -> La URL como descripción de una imagen mostrada, complementa al texto de alt.

Los atributos width y height -> El ancho y alto de la imagen, respectivamente, en píxeles CSS HTML5, o píxeles o porcentaje en HTML 4.

**2.d)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<meta http-equiv="expires" content="16-Sep-2019 7:49 PM" />

Apertura/Cierre de declaración de etiqueta = Obligatorio

Atributo = No Obligatorio

Valor = No Obligatorio

Elemento = Obligatorio

El elemento meta, (Metadatos) sirve para aportar información acerca del documento. Sus etiquetas <meta/> (Solo tiene una). Puede ser contenido por elementos <head>.

Los atributos debe indicarlos el autor.

Atributo name -> Nombre al que se asocia la metainformación.

Atributo lang -> Información sobre el idioma del contenido del elemento y del valor de sus atributos.

Atributo content -> Los datos que se quieren asociar a name.

Atributo http-equiv -> Aporta información sobre los encabezado de respuesta HTTP, puede usarse en lugar de name.

**2.e)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<a href="http://www.e-style.com.ar/resumen.html" type="text/html" hreflang="es"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

charset="utf-8" rel="help">Resumen HTML </a>

Apertura/Cierre de declaración de etiqueta = Obligatorio

Atributo = No Obligatorio

Valor = No Obligatorio

Contenido = No Obligatorio

Elemento = Obligatorio

Este segmento de código Anchor <a> crea un enlace a otras páginas de internet, archivos o ubicaciones dentro de la misma página, direcciones de correo, o cualquier otra URL.

Atributo href -> Contiene una URL o un fragmento de URL al cual apunta el enlace.

Atributo type -> Especifica el tipo de medio (media type) en la forma de MIME type para la URL enlazada. Esto es únicamente informativo, sin ninguna funcionalidad incorporada.

Atributo hreflang -> Este atributo indica el lenguaje humano del recurso al que se enlaza. Este es únicamente informativo, sin ninguna funcionalidad incorporada. Los valores permitidos están determinados por BCP47.

Atributo charset -> (Obsoleto HTML5) Este atributo define la codificación de caracteres (character encoding) de la URL enlazada. El valor debe de ser una lista delimitada por espacio y/o coma de caracteres definidos en RFC 2045. El valor por defecto es ISO-8859-1.

Atributo rel -> Especifica la relación del objeto de destino con el objeto de enlace. El valor es una lista separada por espacios de tipos de enlace tipos de enlace (link types).

**2.f)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<table width="200" summary="Datos correspondientes al ejercicio vencido">

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

<caption align="top"> Título </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">1o</th>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

<tr>

<th scope="row">2o</th>

<td>&nbsp;</td>

<td>&nbsp;</td>

<td>&nbsp;</td>

</tr>

</table>

Apertura/Cierre de declaración de etiqueta = Obligatorio

Atributo = No Obligatorio

Valor = No Obligatorio

Contenido = No Obligatorio

Elemento = Obligatorio

El **Elemento de Tabla** HTML <table> representa datos en dos o más dimensiones.

Las etiquetas de inicio y de fin son obligatorias.

El atributo summary -> Define un texto alternativo para describir una tabla en un usuario incapaz de mostrarlo. (Obsoleto)

width -> Este atributo define el ancho de una tabla, pudiendo contener una longitud de píxeles o un porcentaje, que representa un porcentaje de anchura del contenedor que la tabla debiera usar. No usar este atributo en HTML dado que es obsoleto: el elemento <table> debe ser estilizado usando CSS.

El **Elemento Caption** es el encargado de darle un título descriptivo a las tablas.

Sus etiquetas son: <caption> y </caption> (ambas obligatorias).

align -> Indica la alineación en relación a la tabla('top', 'bottom', 'left', 'right').

Puede contener: Texto y/o cero o más elementos en línea.

El **Elemento fila de tabla (table row)** <tr> define una fila de celdas en una tabla. Estas pueden ser una mezcla de elementos <td> y <th>.

El **Elemento <th>** define una celda como la cabecera de una columna en una tabla. La naturaleza exacta de esta cabecera está definida por los atributos scope y headers.

El atributo Scope -> Un grupo de celdas para las cuales esta celda brinda información de encabezado. Los valores posibles (insensibles a mayúsculas/minúsculas) son:

**row**: todas la celdas en la misma fila.

**col**: todas la celdas en la misma columna.

**rowgroup**: todas la celdas en el mismo grupo de filas (thead, tbody o tfoot).

**colgroup:** todas la celdas en el mismo grupo de columnas (colgroup).

**auto:** los navegadores deciden cuáles son las celdas afectadas basados en un análisis del contexto. Este es el valor por defecto.

El **Elemento <td>** define la celda de una tabla que contiene datos. Esta participa en el modelo de tablas.

**Ejercitación 3**

En cada caso, explicar las diferencias entre los segmentos de código y sus visualizaciones:

3) a)

Los primeros dos elementos <a> se ven iguales en el navegador, con la diferencia que el segundo tiene un atributo/valor target=”\_blank” lo que hace que cuando te redireccione a href=”<http://www.google.com.ar>” lo haga en una pestaña nueva del navegador y no en la actual.

Los elementos número 3 y 6 no se visualizan en el navegador, ya que no tienen contenido.

El elemento número 4 tiene la misma apariencia que los número 1 y 2 pero no realiza ninguna acción ya que su atributo href=”#” contiene un ‘#’ y no redirecciona a ningun enlace.

El atributo del elemento 5 href=”#arriba” lo que hace es redireccionar a otro enlace dentro de la misma página, un elemento que contenga el atributo name=”arriba” (elemento 6) pero como el elemento 6 no tiene contenido no se muestra en pantalla.

3) b)

En todos los segmentos el elemento <p> es padre de los elementos <a> y <img>.

1- En el primer segmento de código, los elementos <img> y <a> son independientes, por lo cuál se muestra una imagen a la izquierda y un enlace (a Google) a la derecha de ésta.

2- En el segundo segmento el elemento <img> se encuentra dentro del elemento <a> y resulta que al presionar en la imagen esta redirecciona a Google y no así el “Click aqui” que es el contenido del elemento <p>.

3- En el segmento 3, el elemento <a> es padre del elemento <img> y además “Click aquí” es su contenido. El elemento <a> redirecciona la página al presionar sobre la imagen o sobre el texto indistintamente.

4- El segmento 4 se trata de dos elemento <a>. Uno contiene al elemento <img> y el segundo contiene Texto “Click aqui”. Son dos enlaces distintos que apuntan a la misma dirección.

3) c)

La diferencia entre el ejemplo 1 y 2 es que el primero se trata de una lista no ordenada <ul> porque lo que sus ítems se muestran con un punto delante y no tienen órden y el segundo se trata de una lista ordenada <ol> y sus ítems se muestran por su correspondiente orden y número.

El ejemplo 3 son 3 listas ordenadas que tienen un único ítem cada una y sus valores están seteados manualmente mediante su atributo value=”n”.

El ejemplo 4 es un elemento <blockquote> que contiene un <p> el cuál simula tener la apariencia de una lista.

3) d)

El primer ejemplo es una tabla <table> que contiene elementos filas <tr>. En su primera fila contiene un encabezados “Columna 1” y “Columna 2” y en sus filas siguiente tienen celdas <td>.

El segundo tiene otra semántica que el del ejemplo 1, al no tener encabezado y utilizar elementos <div> dentro de las celdas para alinear el texto y elementos <strong>.

3) e)

La diferencia entre las dos tablas está en el título. En el ejemplo 1 el título de la tabla se realiza con un caption de una manera ordenada y en el segundo ejemplo el título se realiza con una celda normal que ocupa 3 columnas.

3) f)

Se pueden utilizar los atributos colspan y rowspan para que las columnas o filas ocupen más de un espacio ya que por defecto ocupan 1 espacio cada una.

En el ejemplo 1 se le atribuye dos espacios de manera vertical con el atributo rowspan=”2” a la primera celda, y en el ejemplo 2 se establece que la primera celda ocupará dos espacios de manera horizontal con el atributo colspan=”2”.

3) g)

En el primer ejemplo, la tabla tiene 3 columnas. En la primer fila la primer celda ocupa 2 columnas y 2 filas y luego hay una tercer celda. En la segunda fila la celda que está definida ocupa el lugar 3 y su width es de 50% por lo cuál toda la columna ocupa el 50% de la fila.

3) h)

- En el primer ejemplo el formulario tiene como action=”procesar.php” y method=”post” y target=”\_blank” por eso al presionar el botón Enviar se abre una nueva pestaña, además con las etiquetas <fieldset> se muestra un borde alrededor del formulario.

-El segundo el method=”get” y target=”\_blank” pero en action=”” por lo que se abre la misma página en una nueva pestaña pero como no tiene el código que procese la petición, no hace nada.

3) i)

La principal diferencia es que en el ejemplo 1 el botón se encuentra dentro de un label que contiene el texto y dentro del botón se encuentra un elemento <img> y <b> por lo tanto el botón abarcará lo que ocupe la imagen y el texto CLICK AQUÍ y en el ejemplo 2 dentro del label hay un elemento <input> con type=”button”.

3) j)

Cuando hay elementos <label> de type=”radio” deben tener el mismo nombre en el atributo name=”” para que puedas seleccionar solo una de las opciones.

3) k)

El elemento select (<select>) de HTML representa un control que muestra un menú de opciones.

En el ejemplo dos el elemento <select> tiene el atributo multiple: Este elemento booleano indica que se pueden seleccionar múltiples opciones de la lista. Si no está especificado, sólo se podrá seleccionar una opción cada vez.