Alumnos (Legajo):

Cuello Alejo (45572)

Jimenez Dana (29694)

Piccoli Enzo (42850)

Entornos Gráficos: Práctica 01- HTML

**Ejercitación 1**

**1.1) ¿Qué es HTML, cuando fue creado, cuáles fueron las distintas versiones y cuál es la última?**

HyperText Markup Language, es un lenguaje de marcado para la elaboración de páginas web mediante etiquetas predefinidas que contienen información adicional de la estructura del texto y para la definición de contenido como imágenes, videos, juegos, entre otros. Su principal diferencia con los lenguajes de programación es que éstos últimos poseen funciones aritméticas o variables.

Actualmente es un estándar, a cargo del World Wide Web Consortium (W3C) o Consorcio WWW, que se ha impuesto en la visualización de páginas web y es el que todos los navegadores actuales han adoptado.

El HTML fue desarrollado originalmente por Tim Berners-Lee y fue popularizado por el navegador Mosaic.

Originalmente, HTML fue diseñado principalmente como un lenguaje para describir semánticamente documentos científicos.

Durante los años 90 ha proliferado con el crecimiento explosivo de la Web. El HTML se ha desarrollado de diferentes maneras y gracias a su diseño general ha permitido que se adapte, en los años posteriores, para describir varios otros tipos de documentos e incluso aplicaciones.

Entre *1990-1995*, HTML pasó por una serie de revisiones y experimentó una serie de extensiones, alojadas principalmente en el CERN (Organización Europea para la Investigación Nuclear), y luego en el IETF (Internet Engineering Task Force). Luego con la creación del W3C, que es donde actualmente se encuentran sus recomendaciones y estandarizaciones.

Un primer intento frustrado de extender el HTML en *1995* conocido como **HTML 3.0** luego dio paso a un enfoque más pragmático conocido como **HTML 3.2**, que se completó en *1997*. **HTML 4.01** siguió rápidamente más tarde ese mismo año.

Al año siguiente, la membresía del W3C decidió dejar de evolucionar HTML y en su lugar comenzó a trabajar en un equivalente basado en XML, llamado XHTML. Este esfuerzo comenzó con una reformulación de HTML 4.01 en XML, conocido como XHTML 1.0, que no agregó nuevas funciones, excepto la nueva serialización, y que se completó en 2000. Después de XHTML 1.0, el enfoque del W3C se volvió más fácil para otros grupos de trabajo para extender XHTML, bajo el título de XHTML Modularization. Paralelamente, el W3C también trabajó en un nuevo lenguaje que no era compatible con los lenguajes HTML y XHTML anteriores, llamándolo XHTML 2.0.

El WHATWG se basó en varios principios básicos, en particular, que las tecnologías deben ser compatibles con versiones anteriores, que las especificaciones y las implementaciones deben coincidir incluso si esto significa cambiar la especificación en lugar de las implementaciones, y que las especificaciones deben ser lo suficientemente detalladas como para que las implementaciones interoperabilidad completa sin ingeniería inversa entre sí. Este último requisito en particular requiere que el alcance de la especificación HTML incluya lo que se había especificado previamente en tres documentos separados: HTML 4.01, XHTML 1.1 y DOM Nivel 2 HTML. También significó incluir muchos más detalles de los que se habían considerado anteriormente como la norma.

En 2006, el W3C indicó su interés en participar en el desarrollo de HTML 5.0 después de todo, y en 2007 formó un grupo de trabajo autorizado para trabajar con WHATWG en el desarrollo de la especificación HTML.

HTML 5 es la quinta revisión más importante que se hace al lenguaje HTML.

En este trabajo, se introducen nuevas características para ayudar a los autores de aplicaciones Web, y se ha prestado especial atención a la definición de claros criterios de conformidad para los agentes de usuario (navegadores) en un esfuerzo por mejorar la interoperabilidad (la habilidad de dos o más sistemas o componentes para intercambiar información y utilizar la información intercambiada)

**1.2) ¿Cuáles son los principios básicos que el W3C recomienda seguir para la creación de documentos con HTML?**

**Separar estructura y presentación**

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 la presentación 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 representar sus documentos en diferentes plataformas: navegadores basados en voz, lectores braille, etc.

El HTML ofrece un número de mecanismos con este fin (p.ej., el atributo alt, el atributo accesskey, etc.)

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 con la representación incremental**

Mediante un diseño cuidadoso de las tablas y haciendo uso de las nuevas características de representación incremental, los autores pueden ayudar a los agentes de usuario a representar los documentos más rápidamente.

**1.3) En las Especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿y obsoleto?**

Un elemento o atributo **desaprobado** es aquel que ha quedado anticuado por la presencia de estructuras nuevas. Los elementos desaprobados pueden declararse obsoletos en el futuro.

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

Esta especificación incluye ejemplos que ilustran cómo evitar el uso de elementos desaprobados. En la mayoría de los casos estos ejemplos dependen de que el agente de usuario soporte hojas de estilo. Por ejemplo, los atributos de presentación de HTML han sido desaprobados cuando existen alternativas con hojas de estilo.

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.

**1.4) ¿Qué es el DTD y cuáles son los posibles DTDs contemplados en la especificación de HTML 4.01?**

DTD es la declaración del tipo de documento (en inglés, Document Type Declaration). Es una sección que se ubica en la primera línea del archivo HTML, es decir antes de la marca html. Las posibles DTDs contempladas en la especificación de HTML 4.01 varían en cuanto a los elementos que soportan, y son las siguientes:

* DTD estricto: incluye todos los elementos y atributos que no han sido desaprobados o que no aparecen en documentos con marcos
* DTD transicional: incluye todo lo que incluye el DTD estricto más los elementos y atributos desaprobados.
* DTD para documentos con marcos: incluye todo lo que incluye el DTD transicional más los marcos.

**1.5) ¿Qué son los metadatos y cómo se especifican en HTML?**

Los metadatos son información sobre el documento, y en HTML se pueden especificar utilizando el elemento *meta* para identificar dichas propiedades de un documento (p.ej., el autor, la fecha de caducidad, una lista de palabras clave, etc.) y para asignar valores a esas propiedades. Puede utilizarse el atributo content para especificar un valor correspondiente a otro atributo especificado dentro de la misma etiqueta meta (por ejemplo, el lenguaje del documento.

**Ejercitación 2**

**2.a)** <!-- Código controlado el día 12/08/2009 -->

Es un comentario, se puede ubicar en cualquier parte de un documento HTML. No producen ningún efecto solo el documento, tiene un fin meramente informativo para quienes tengan acceso al código. Suelen ser utilizados para explicar el funcionamiento de diferentes partes del código o para hacer aclaraciones sobre el mismo. Un comentario comienza con los caracteres *<!-*- y termina con los caracteres -->

**2.b)** <div id="bloque1">Contenido del bloque1</div>

Se colocan dentro del body de un documento HTML. Se utiliza para definir un bloque contenedor de otras etiquetas. Está compuesta por una etiqueta de apertura con el nombre *div,* el contenido a mostrar dentro del bloque y una etiqueta de cierre con el mismo nombre. En este ejemplo, cuenta con el atributo de nombre id y su respectivo valor “bloque1”. Dicho atributo no es obligatorio, pero es una buena práctica utilizarlo.

**2.c)** <img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32"

longdesc="detalles.htm" />

Se colocan dentro del body de un documento HTML. Se utiliza para mostrar una imagen en el dispositivo que visualice el documento HTML. Consta de una etiqueta de apertura con nombre img, y los siguientes atributos:

* src: en este caso, el valor es un string vacío. Este atributo es obligatorio ya que contiene la ruta del archivo a mostrar en el documento HTML.
* alt: en este caso, el valor es un string vacío. Es un texto alternativo en caso que no se visualice la imagen en el agente del usuario. Este atributo no es obligatorio.
* id: su valor es “bloque1”. Dicho atributo no es obligatorio, pero es una buena práctica utilizarlo.
* name: su valor es “im1”. Dicho atributo es utilizado por el servidor para identificar los campos en el envío de formularios. No es obligatorio.
* width: su valor es 32. Especifica el ancho que tendrá la imagen (en pixeles). No es obligatorio, teniendo como valor por defecto el ancho original de la imagen.
* height: su valor es 32 especifica el alto que tendrá la imagen (en pixeles). No es obligatorio, teniendo como valor por defecto el ancho original de la imagen.
* longdesc: sirve para especificar una URL con información más detallada de la imagen. No es obligatorio.

**2.d)** <meta name="keywords" lang="es" content="casa, compra, venta, alquiler " />

La etiqueta meta define metadatos sobre un documento HTML. Los metadatos son datos (información) sobre datos. Las etiquetas <meta> siempre van dentro del elemento <head> y normalmente se utilizan para especificar el juego de caracteres, la descripción de la página, las palabras clave, el autor del documento y la configuración de la ventana gráfica. En el ejemplo de arriba es utilizada para describir las palabras clave mediante el atributo name con valor keywords. El atributo lang con valor es, indica que las palabras descritas en el valor del atributo content, corresponden al idioma español.

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

El atributo de nombre http-equiv con valor expires se utiliza para establecer la fecha de expiración para definir cuando el navegador debe mostrar la página en producción, la que se aloja en el servidor, en vez de mostrar la que está almacenada en caché.

**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>

La etiqueta define un hipervínculo, que se utiliza para enlazar una página con otra. El atributo más importante del elemento <a> es el atributo href, cuyo valor indica el destino del enlace. En caso de no especificarlo, el contenido dentro de las etiquetas de apertura y de cierre, no cumplirá la función de vincular documentos. El atributo type indica mediante su valor el tipo de medio del documento vinculado. El atributo hreflang indica mediante su valor, el lenguaje del documento vinculado (en este ejemplo, español). El atributo rel indica el tipo de relación entre el documento actual y el vinculado, siendo en este caso, de ayuda para el contexto del actual documento. El atributo charset de este ejemplo, debería especificarse dentro de una etiqueta meta.

**2.f)** <table width="200" summary="Datos correspondientes al ejercicio vencido">

El tag de nombre table indica la definición de una tabla entre su etiqueta de apertura y de cierre. El atributo width especifica el ancho de la tabla en pixeles. El atributo de nombre summary se utiliza para proporcionar también una descripción larga.

<caption align="top"> Título </caption>

El tag caption indica entre sus etiquetas de apertura y de cierre, el título que contendrá la tabla. El atributo align indica la alineación vertical del mismo, en este caso superior.

<tr>

Este tag indica entre sus etiquetas de apertura y cierre, la definición de una fila de la tabla. Dentro de esta etiqueta deberán definirse etiquetas <th> o <td> para definir celdas que tengan contenidos.

<th scope="col">&nbsp;</th>

Indica la definición de una celda de encabezado, especificando su contenido entre sus etiquetas de apertura y de cierre. Mediante el atributo scope, indica si es un encabezado para una columna, fila o grupo de columnas o filas. En este caso particular, el contenido &nbsp indica un espacio de no separación (lo que sería equivalente a un string vacío ‘’)

<td>&nbsp;</td>

Indica la definición de una celda de datos.

**Ejercitación 3**

**3.a)** <a href="http://www.google.com.ar">Click aquí para ir a Google</a>

Consulta el documento ubicado en la URL: <http://www.google.com.ar>, abriéndolo en la pestaña actual.

<a href="http://www.google.com.ar" target="\_blank">Click aquí para ir a Google</a>

Consulta el documento ubicado en la URL: <http://www.google.com.ar>, abriéndolo en una nueva pestaña.

<a href="http://www. google.com.ar" type="text/html" hreflang="es" charset="utf-8" rel="help">

Al ser solo una etiqueta de apertura, no cuenta con contenido que nos permita establecer el vínculo con la URL especificada en el atributo href.

<a href="#">Click aquí para ir a Google</a>

Establece un vínculo con el documento actual

<a href="#arriba">Click aquí para volver arriba</a>

Establece un vínculo con la etiqueta a que tenga el atributo name con el mismo valor que se especifica en el atributo href.

<a name="arriba" id="arriba"></a>

Establece el destino del vínculo especificado en la línea anterior

**3.b)**

<p><img src="im1.jpg" alt="imagen1" /><a href="http://www.google.com.ar">Click aquí</a></p>

Crea un link a la ruta especificada mediante el texto especificado, sin incluir la imagen.

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> Click aquí</p>

Crea un link a la ruta especificada mediante la imagen. Además, la imagen tiene un texto alternativo diferente al ejemplo anterior.

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" />Click aquí</a></p>

Crea un link a la ruta especificada mediante la imagen y el texto.

<p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> <a

href="http://www.google.com.ar">Click aquí</a></p>

Crea un link mediante la imagen, y crea otro link a través del texto.

**3.c)**

<ul>

<li>xxx</li>

<li>yyy</li>

<li>zzz</li>

</ul>

Crea una lista ordenada con 3 ítems, enumerados desde el valor 1 hasta el 3.

<ol>

<li>xxx</li>

<li>yyy</li>

<li>zzz</li>

</ol>

Crea una lista desordenada, diferenciando cada item pero sin enumerarlos

<ol>

<li>xxx</li>

</ol>

<ol>

<li value="2">yyy</li>

</ol>

<ol>

<li value="3">zzz</li>

</ol>

Crea 3 listas ordenadas distintas con distintos valores iniciales para los items que contienen cada una de ellas. El aspecto visual es similar al ejemplo anterior, diferenciando que hay más espacio entre los items ya que pertenecen a listas distintas.

<blockquote>

<p>

1. xxx<br />

2. yyy<br />

3. zzz

</p>

</blockquote>

La etiqueta blockquote crea citas en bloque, marca las citas a otros autores o documentos. El contenido dentro de este bloque es un párrafo con varios saltos de líneas, logrando el mismo aspecto que una lista ordenada.

**3.d)** Ambas tablas tienen el mismo aspecto virtual, aunque se aplican mejores prácticas en la primera. En la segunda, se busca mediante otras etiquetas, solo lograr el mismo aspecto visual que una celda de encabezado.

**3.e)** En ambos ejemplos tenemos tablas con el mismo aspecto visual. En la primera tenemos un título fuera de la tabla, y luego dos filas. En la segunda, el texto “título” es incluido dentro de una fila de la tabla.

**3.f)** La diferencia es que en la primer tabla, una celda ocupa dos filas. Y en la segunda, una celda ocupa dos columnas.

**3.g)** En el primer ejemplo se define una tabla de 3 filas por 3 columnas, con celdas que ocupan más de un espacio. El segundo ejemplo es visualmente similar, con la diferencia que el espacio entre las celdas es 0, unificando los bordes de las mismas. Además esta tabla es de 3 filas por 2 columnas.

**3.h)** El primer formulario utiliza el método post (especificado en el atributo method), por lo que los datos obtenidos de los distintos inputs que lo componen, irán en el body de la petición. El formulario será enviado a un archivo llamado procesar.php, que debe estar dentro de la misma carpeta que el documento HTML. Cuenta con dos inputs con un valor inicial “xxx” en ambos. El formulario es enviado mediante el botón “enviar” ya que es de tipo submit.

El segundo formulario tiene una diferencia visual marcada por la ausencia del tag fieldset, que en el ejemplo anterior recuadra los inputs del formulario (además de esto, el fieldset no aporta ninguna funcionalidad). Los inputs no cuentan con valores iniciales, y el botón Enviar cumple la misma función que el ejemplo anterior. Este formulario utiliza el método post, por lo que la información irá en la URL a enviar. En este caso no tiene ninguna acción asociada.

El tercer ejemplo envía la información del formulario mediante un método get, al mail especificado en el atributo action. El botón de dicho formulario cumple una función de reseteo del mismo, por lo que se limpiarán todos los inputs una vez clickeado.

3.i) El primer botón contiene una imagen dentro del mismo, junto con un texto “Click aquí”. Dicho contenido es especificado entre las etiquetas de apertura y de cierre.

El segundo botón es un input de un formulario, que solo cuenta con el texto “Click aquí”, especificado mediante el atributo value. Ninguno de los dos botones tiene una funcionalidad definida.

**3.j)** En el primer ejemplo, los dos radio buttons tienen el mismo valor en el atributo name, por lo que permite al usuario seleccionar solo uno de ellos. En el segundo ejemplo, tienen un valor distinto, por lo que puede seleccionarse ambos a la vez.

**3.k)** En el primer ejemplo, se muestra un select sin opción múltiple, se despliegan sus opciones una vez que se hace click sobre dicho elemento. En cambio, el segundo ejemplo cuenta con un listado deslizable que muestra todas las opciones disponibles, permitiendo seleccionar más de una. El valor del atributo name en los dos ejemplos no genera una diferencia visual ni funcional.