Práctica 1 – HTML

Universidad Tecnológica Nacional – Facultad Regional Rosario

Asignatura: Entornos Gráficos

Año 2022

Alumno: Tadeo Schlieper

**Ejercitación 1**

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

HTML es un lenguaje de marcado que permite enunciar y publicar un documento o página web con contenido y estructura propios y enlazados, cuyas especificaciones están en constante revisión, y por ello nacen diferentes versiones HTML.

Su creación tuvo lugar en el año 1990, con la posteriormente considerada versión 1.0, pero no fue esa versión la que llegó a nuestros días.

La versión 1.0 tuvo que pasar por una serie de revisiones en busca de la convención de sus especificaciones para tener interoperabilidad entre diferentes sistemas de todo el mundo. Las versiones desarrolladas en trabajo conjunto entonces fueron las HTML 2.0 (1995), 3.2 (1997-2018), 4.0 (1997-2018), 4.01 (1999-2018), el conjunto de versiones HTML 5.X (2008/2014-2021) y el *Living Standard for HTML* (2019/2021 en adelante). Los navegadores intentan ser compatibles con varias de estas versiones. El término “HTML5” incluye a versiones tales como la 5.0 (2014-2018), 5.1 (2016-2021) y 5.2 (2017-2021), así como al *Living Standard for HTML*, tal como se explica a continuación:

En 2019 sucedió que los grupos encargados de trabajar las especificaciones HTML, WHATWG y W3C, firmaron un acuerdo para colaborar hacia adelante en una sola versión en constante evolución (“*Living Standard for*”) de HTML, con la cual WHATWG venía trabajando. Así es como en enero de 2021 W3C terminó por retirar las últimas versiones “numeradas” que venía trabajando de HTML5 (5.1, 5.2, y 5.3), y pasó a recomendar el nuevo “HTML”. Aunque desde ese momento y hasta ahora, W3C considera que dicha versión única está (todavía) bajo el paraguas de HTML5.

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

El W3C recomienda seguir unos principios para crear un documento HTML, aprovechando las herramientas que ofrecen las especificaciones, los cuales son:

1. Separar la estructura de la presentación. Es decir, hacer uso de la facilidad que ofrece actualmente HTML para enlazar un documento de hojas de estilo (comúnmente CSS) y depositar la mayor cantidad de estilos posibles allí, los cuales serán “la presentación”, para que en el documento HTML quede principalmente “la estructura”.
2. Considerar la accesibilidad universal a la web. Dado que a la web acceden millones de personas a través de diferentes sistemas con diferentes capacidades y culturas, entonces se recomienda apostar por la accesibilidad creando experiencias y diseños alternativos, y manejando propiedades relacionadas a la codificación del texto HTML que puede diferir de un idioma natural a otro.
3. Ayudar con la representación incremental. Por una cuestión de practicidad y velocidad, e incluso de accesibilidad por ancho de banda, se recomienda hacer uso de las características de representación incremental del HTML, que posibilita una carga de los documentos más rápida (fraccionada).
4. *En las especificaciones de HTML, ¿cuándo un elemento o atributo se considera desaprobado? ¿Y obsoleto?*

La especificación HTML define, justamente, los elementos y atributos que pueden utilizarse, pero también especifica el estado de algunos de ellos entre desaprobados y obsoletos.

Desaprobado es cuando hay alternativas (generalmente novedosas) en la especificación más óptimas para conseguir los mismos efectos y se insta con ejemplos a utilizar las alternativas, en lugar del elemento o atributo en cuestión; a su vez se recomienda a los navegadores seguir soportando su utilización, además de las alternativas, aunque quizás en vistas de que su estado de desaprobado pueda convertirse en obsoleto en un futuro.

A la vez, tenemos el estado obsoleto, bajo el cual el elemento o atributo en cuestión pierde la definición en la especificación (como puede ser en el DTD) y por lo tanto no habrá garantía de soporte en los agentes de usuario que sigan dicha especificación.

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

El DTD (del inglés *Document Type Declaration*) es una definición del tipo de documento SGML o XML, que se conforma con una sintaxis particular, y permite validar que un documento particular cumpla con un DTD particular. En el caso del HTML debemos considerar que hasta la versión 4.01 el lenguaje está basado en SGML, por lo que hereda y utiliza DTDs que están ligados a las especificaciones, pero desde la versión 5.0 ya no está basado en SGML y tampoco tiene DTDs, y en su lugar [se pueden utilizar validadores HTML que siguen la última especificación](https://github.com/whatwg/html/blob/main/FAQ.md#if-there-is-no-dtd-how-can-i-validate-my-page) para validar el cumplimiento de la misma.

En HTML 4.01 existen 3 DTDs posibles:

* DTD HTML 4.01 (Strict): define los elementos y atributos del HTML 4.01 que no están desaprobados (ni obsoletos) ni los necesarios para crear marcos.
* DTD HTML 4.01 Transitional: define los elementos y atributos del DTD Strict más los desaprobados.
* DTD HTML 4.01 Frameset: define los elementos y atributos del DTD Transitional más los necesarios para crear marcos.

1. *¿Qué son los metadatos y cómo se especifican en HTML?*

Los metadatos se definen, en general, como datos que describen o identifican a otros datos.

Un documento (sea o no HTML) posee propiedades sobre el mismo, más allá del contenido principal, como un nombre, un autor, o una fecha de creación. Esas propiedades se pueden almacenar, por ejemplo, dentro o fuera del documento; en el caso de HTML, algunas de estas se permiten indicarlas dentro del mismo documento en forma de elementos y atributos especificados con ese objetivo. El elemento usado para los metadatos recomendado por W3C desde 1999 es “meta”; por ejemplo, se podría escribir: <meta name="author" lang="es" content="Juan Pérez"> para indicar que el autor del documento se llama Juan Pérez y que tal nombre está escrito en español.

**Ejercitación 2**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ref. código | En qué sección se coloca | Efecto | Ele-mentos | Etiquetas | Atributos |
| 2.a) | Se coloca en cualquier parte del HTML que no sea dentro de ninguna etiqueta, ni dentro de elementos cuyo contenido no sea lenguaje HTML (como pueden ser los elementos “style” y “script”) | Todo su contenido es recibido pero ignorado por el agente de usuario, por lo que solo puede verlo el lector del código fuente del documento. | Co-men-tario | Apertura: <!—  Cierre: --> | No aplica |
| 2.b) | Se ubica dentro del único elemento “body”, a su vez dentro del (único) elemento “html”. | Genera un bloque (con salto de línea anterior y posterior) con id único a través del documento “bloque1”, que se muestra al usuario y cuyo contenido es un texto que dice “Contenido del bloque1” | div | Apertura: <div id="bloque1">  Cierre: </div> | id  Valor: bloque1  Es opcional |
| 2.c) | Se ubica dentro del único elemento “body”, a su vez dentro del (único) elemento “html”. | Genera un cuadro de imagen, de 32x32 píxeles, con texto alternativo visible por no tener imagen que dice “lugar imagen”, con id único “im1”. Y con nombre “im1” y una redirección a una descripción detallada en el archivo detalles.htm, aunque actualmente los navegadores ignoran los atributos “name” y “longdesc” en los elementos “img”, son obsoletos desde la primer versión de HTML5 ([enlace a la especificación actual](https://html.spec.whatwg.org/multipage/obsolete.html#non-conforming-features)). | img | Apertura y cierre: <img src="" alt="lugar imagen" id="im1" name="im1" width="32" height="32" longdesc= "detalles.htm" /> | src  Valor: ""  Es obligatorio |
| alt  Valor: "lugar imagen"  Es obligatorio |
| id  Valor: "im1"  Es opcional |
| name  Valor: "im1"  Es opcional y obsoleto |
| width  Valor: "32"  Es opcional |
| height  Valor: "32"  Es opcional |
| longdesc  Valor: "detalles.html"  Es opcional y obsoleto |
| 2.d) | Se ubican ambos elementos dentro del único elemento “head”, a su vez dentro del (único) elemento “html”. | El agente de usuario (generalmente un motor de búsqueda) reconoce el metadato de “keywords”, que contiene las palabras clave acerca del documento, las cuales también se indica que están en idioma español. | meta | Apertura y cierre: <meta name= "keywords" lang="es" content= "casa, compra, venta, alquiler" /> | name  Valor: "keywords" |
| lang  Valor: "es"  Es opcional |
| content  Valor: "casa, compra, venta, alquiler" |
| El agente de usuario reconoce como metadato la consigna de enviar en las peticiones HTTP junto con el documento la cabecera “Expires” con valor “16-Sep-2019 7:49 PM”. | meta | Apertura y cierre: <meta http-equiv= "expires" content="16-Sep-2019 7:49 PM" /> | http-equiv  Valor: "expires"  Es obligatorio colocar este atributo ó el atributo name ó el atributo charset |
| content  Valor: "16-Sep-2019 7:49 PM" |
| 2.e) | Se ubica dentro del único elemento “body”, a su vez dentro del (único) elemento “html”. | Se muestra al usuario el texto “Resumen HTML ”, que funciona como enlace de ayuda y lo dirige, al hacer click en éste, a la página web http://www.e-style.com.ar/resumen.html, de la cual se indica que es un documento HTML y en idioma español. Y con codificación UTF-8, aunque actualmente los navegadores ignoran el atributo “charset” en elementos “a”, es obsoleto desde HTML5 ([enlace a la especificación actual](https://html.spec.whatwg.org/multipage/obsolete.html#non-conforming-features)). | a | Apertura: <a href="http:// www.e-style.com.ar/ resumen. html" type= "text/html" hreflang="es" charset="utf-8" rel="help">  Cierre: </a> | href  Valor: " http:// www.e-style.com.ar/ resumen. html "  Es opcional. |
| type  Valor: "text/html"  Es opcional. |
| hreflang  Valor: "es"  Es opcional. |
| charset  Valor: "utf-8"  Es opcional y obsoleto. |
| rel  Valor: "help"  Es opcional. |
| 2.f) | Se ubica dentro del único elemento “body”, a su vez dentro del (único) elemento “html”, para definir una tabla. | Genera una tabla de 200 píxeles de ancho, la cual posee una leyenda sobre la misma con el texto “Título”, con 4 columnas y 3 filas, incluyendo las primeras diferentes (o cabeceras) columna (“1°” y “2°”) y fila (“A”, “B” y “C”), y el resto de celdas con un espacio de contenido. Sin embargo, los atributos que hacen que la tabla sea de 200 píxeles de ancho (“width”) y la leyenda se ubique sobre la misma (“align”), son obsoletos desde HTML5. Y actualmente los navegadores que siguen la última especificación ignoran el atributo también obsoleto “summary”, el cual mostraría un texto resumen “Datos correspondientes al ejercicio vencido”. | table | Apertura: <table width="200" summary= "Datos correspon-dientes al ejercicio vencido">  Cierre: </table> | width  Valor: "200"  Es opcional y obsoleto. |
| summary  Valor: "Datos correspon-dientes al ejercicio vencido"  Es opcional y obsoleto. |
| Se ubica siempre inmediata-mente luego de la etiqueta de apertura de un elemento “table”. | caption | Apertura: <caption align="top">  Cierre: </caption> | align  Valor: “top”  Es opcional y obsoleto. |
| Se ubica dentro de un elemento “table” para definir una fila de tabla. | tr (3 veces) | Apertura (3 veces): <tr>  Cierre (3 veces): </tr> | No tiene. |
| Se ubica dentro de un elemento “tr” para definir una celda de encabezado de fila o columna. | th (6 veces) | Apertura (4 veces): <th scope="col">  Apertura (2 veces): <th scope="row">  Cierre (6 veces): </th> | scope  Valor (4 veces): "col"  Valor (4 veces): "row"  Es opcional. |
| Se ubica dentro de un elemento “tr” para definir una celda de datos. | td (6 veces) | Apertura (6 veces): <td>  Cierre (6 veces): </td> | No tiene. |

**Ejercitación 3**

3.a)

1. <a href="http://www.google.com.ar">Click aquí para ir a Google</a>
2. <a href="http://www.google.com.ar" target="\_blank">Click aquí para ir a Google</a>
3. <a href="http://www.google.com.ar" type="text/html" hreflang="es" charset="utf-8" rel="help">
4. <a href="#">Click aquí para ir a Google</a>
5. <a href="#arriba">Click aquí para volver arriba</a>
6. <a name="arriba" id="arriba"></a>

El segmento de código 3 es una etiqueta de apertura de un elemento “a” del cual no se sabe el contenido y falta la etiqueta de cierre, mientras que el resto (1, 2, 4, 5 y 6) son elementos “a”.

1, 2 y 4 muestran el texto “Click aquí para ir a Google”, 5 muestra “Click aquí para volver arriba”, y 6 no se muestra porque no tiene contenido (y 3 no se sabe).

1, 2, 4 y 5 funcionan, y 3 funcionaría, como enlace, en el caso de 3 como enlace de ayuda; mientras que 6 funciona como ancla o marcador del fragmento del documento, fragmento llamado “arriba” indicado por el atributo “id”, y el atributo “name” es obsoleto.

En 2, el direccionamiento se hará en una nueva pestaña o ventana, mientras que en el resto no se especifica y será por defecto: en la misma pestaña.

1 y 2 dirigen, y 3 dirigiría, al hacer click en sus respectivos contenidos, a la página web http://www.google.com.ar, y en 3 se indica que dicha página es un documento HTML, está en idioma español y codificada en UTF-8 (aunque el atributo “charset” es obsoleto). En cambio, 4 y 5 dirigen, al hacer click en éstos, a fragmentos del documento HTML actual; 4 dirige al mismo lugar en que se encuentra el agente de usuario, pues apunta a un fragmento nulo (“#”), lo que no tiene más efecto que agregar el símbolo # a la ubicación actual; mientras que 5 dirige al lugar donde esté referenciado el fragmento “arriba” (“#arriba”), que podría existir o no existir, lo que en el último caso sería como dirigir a un fragmento nulo. 6 no dirige a ningún sitio dado que no funciona como enlace sino, como se explicó más arriba, como ancla.

3.b)

1. <p><img src="im1.jpg" alt="imagen1" /><a href="http://www.google.com.ar">Click aquí</a></p>
2. <p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" /></a> Click aquí</p>
3. <p><a href="http://www.google.com.ar"><img src="im1.jpg" alt="imagen1" />Click aquí</a></p>
4. <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>

Los 4 segmentos de código son elementos “p”, es decir, párrafos.

En los 4 segmentos, el elemento “p” contiene y muestran, mediante un elemento “img”, la imagen “im1.jpg”, ubicada en el mismo directorio donde se ubica el documento HTML y con texto alternativo “imagen1”, y luego de la imagen el elemento “p” contiene al texto “Click aquí”, el cual, por lo tanto, se muestra seguido de la imagen.

Todos contienen al menos un elemento “a”, un enlace que dirige a la página web “http://www.google.com.ar”.

La diferencia entre los segmentos es qué está contenido dentro de el/los elemento/s “a”, como se explica a continuación. En 2 y 4, la imagen está a su vez contenida dentro del mencionado elemento “a”. En 1 y 4, el texto “Click aquí” también está contenido en un elemento “a” idéntico al mencionado. Y en 3, ambos (el elemento “img” y el texto) están contenidos dentro de un único elemento “a” como el mencionado.

3.c)

1. <ul>…
2. <ol><li>xxx</li><li>…
3. <ol><li>xxx</li></ol>…
4. <blockquote>…

Los primeros 3 segmentos de código usan elementos apropiados para hacer listas, mientras que el segmento 4 es un elemento “blockquote” que representa una cita.

1 y 2 son listas, desordenada (sin numeración) y ordenada (con numeración) respectivamente, con los mismos 3 ítems “li” cada una: “xxx”, “yyy” y “zzz”. En el segmento 3, en cambio, hay 3 elementos separados de lista ordenada “ol”, cada uno con un ítem diferente: “xxx”, “yyy” y “zzz” respectivamente. Los ítems de la segunda y tercera lista tienen el atributo “value” con los valores “2” y “3” respectivamente, para simular una continuidad en el orden respecto a la lista anterior.

Por último, el segmento 4 es una cita cuyo contenido es un párrafo “p” que simula ser una lista, incluyendo 2 saltos de línea “br” entre los textos que simulan ser ítems de la lista: “1. xxx", “2. yyy" y “3. zzz”.

3.d)

1. <table border="1" width="300"><tr><th>…
2. <table border="1" width="300"><tr><td>…

Ambos segmentos de código son un elemento “table” y producen una tabla de 2 columnas por 3 filas con ciertos atributos y contenido. La diferencia radica en que el primer segmento utiliza las 2 celdas de la primera fila como celdas de encabezado “th”, en cambio el segundo segmento utiliza para estas celdas el elemento de celda de datos “td” y aplica sobre éstas estilos (el atributo de estilo obsoleto “align” y el elemento de estilo “strong”) que simulan los estilos por defecto de las celdas de encabezado “th”. Visualmente, típicamente ambos segmentos producen el mismo resultado, pero semánticamente el agente de usuario reconoce en el primer segmento las celdas de encabezado mientras que en el segundo segmento trata a todas las celdas por igual.

3.e)

1. <table width="200"><caption>…
2. <table width="200"><tr>…

Ambos segmentos son un elemento “table” y producen una tabla de 3 columnas por 3 filas con ciertos atributos y contenido. La diferencia radica el primer segmento utiliza una leyenda con el elemento “caption” a modo de título de la tabla por fuera de las celdas, mientras que en el segundo segmento coloca ese título dentro de una celda que ocupa las 3 columnas de ancho y se alinea al centro para que, con los estilos por defecto del navegador, pueda simular visualmente la leyenda hecha con “caption” en el segmento 1.

3.f)

1. <table width="200"><tr><td colspan="3"><div align="center">Título</div></td></tr><tr><td rowspan="2"…
2. <table width="200"><tr><td colspan="3"><div align="center">Título</div></td></tr><tr><td colspan="2"…

Ambos segmentos de código son un elemento “table” y producen una tabla de 3 columnas por 3 filas con ciertos atributos y contenido.

La diferencia es que, en el primer segmento, la 1° celda de la 2° fila tiene el atributo “rowspan” con valor “2”, lo que produce que dicha celda ocupe la 2° y 3° fila de la 1° columna, y a la vez para mantener la consistencia de la tabla hay una celda menos en la 3° fila; mientras que, en el segundo segmento, la misma 1° celda de la 2° fila tiene el atributo “colspan” con valor “2”, lo que produce que dicha celda ocupe la 1° y 2° columna de la 2° fila, y a la vez para mantener la consistencia de la tabla hay una celda menos en la 2° fila.

3.g)

1. <table width="200" border="1">…
2. <table width="200" border="1" cellpadding="0" cellspacing="0">…

Ambos segmentos de código son un elemento “table” y producen una tabla de 3 filas, de 200 píxeles de ancho y borde (con los atributos obsoletos “width” y “border”), pero en el segmento 1 la tabla tiene 3 columnas mientras que en 2 tiene 2 columnas.

En 1 se definen las celdas de la siguiente manera. La 1° celda de la 1° fila tiene el atributo “colspan” con valor “3”, lo que hace que ocupe las 3 columnas de la fila. Luego la 1° celda de la 2° fila tiene los atributos “colspan” y “rowspan” con valor “2”, lo que hace que la celda ocupe la 1° y 2° columna de la 2° y 3° fila. Para mantener la consistencia de la tabla, hay otra celda en la 2° fila que vendría a ocupar su 3° columna, y una única celda en la 3° fila que ocupa su 3° columna.

Por otro lado, en 2 la tabla tiene los atributos “cellpadding” y “cellspacing” con valor 0 para cambiar el estilo por defecto, además de las diferencias en sus celdas. La 1° celda de la 1° fila tiene el atributo “colspan” con valor “2”, lo que produce que dicha celda ocupe las 2 columnas de la fila. Luego la 1° celda de la 2° fila tiene el atributo “rowspan” con valor “2”, ocupando la 2° y 3° fila de la 1° columna. Para mantener la consistencia de la tabla, hay otra celda en la 2° fila que ocupa su 2° columna, más una única celda en la 3° fila que ocupa su 2° columna.

3.h)

1. <form id="form1"…
2. <form id="form2"…
3. <form id="form3"…

Los 3 segmentos de código son elementos “form” y producen un formulario con 2 campos para usuario y contraseña. Poseen un elemento “input” al final a modo de botón. En el caso de 1 y 2 este botón ejecuta la acción del formulario, mientras que en 3 reinicia los campos del formulario.

En 1 la acción del formulario es una petición HTTP que envía el contenido ingresado por el usuario en una pestaña nueva, vía método POST a la dirección “procesar.php”. En 2 la acción es igual excepto que es vía método GET y a la misma dirección donde se encuentra el documento HTML. En 3, la acción, aunque definida, no podría ser ejecutada por el usuario: sería una petición de redacción de mail en una pestaña nueva, conteniendo los campos ingresados por el usuario, a la dirección xx@xx.com.

La otra diferencia es que en 1 y 3 se usan los elementos “fieldset” y “legend”, para agrupar los campos de ingreso de datos y para mostrar un nombre a ese agrupamiento respectivamente.

3.i)

1. <label>Botón 1…
2. <label>Botón 2…

1 y 2 son elementos “label”, los cuales contienen un texto seguido de diferentes tipos de botones.

En 1 se usa el elemento “button”, el cual puede contener otros elementos HTML en su interior, y se le agrega a éste una imagen “logo.jpg” de 30x20 píxeles con texto alternativo “Botón con imagen ”, un salto de línea y un texto en negrita “CLICK AQUÍ”. En cambio, en 2 se usa el elemento de auto-cierre “input” con el atributo “type” igual a “button”, y el atributo “value” con valor “CLICK AQUÍ” el cual se muestra como texto en el botón generado.

3.j)

1. <p><label><input type="radio" name="opcion"…
2. <p><label><input type="radio" name="opcion1"…

Ambos códigos producen en un párrafo “p” 2 inputs de radio con valores “X” e “Y” y que muestran los textos “X” e “Y” respectivamente.

La diferencia es que en 1 ambos inputs usan el mismo nombre (atributo “name”) “opcion" por lo que hacen referencia a la misma entrada de datos y sólo puede seleccionarse uno, mientras que en 2 cada input hace referencia a una entrada diferente, única, con los nombres “opcion1” y “opcion2”, por lo que se seleccionan ambos si el usuario lo desea.

3.k)

1. <select name="lista">…
2. <select name="lista[]" multiple…

Ambos segmentos de código son un elemento “select”, un selector de opciones para que el usuario elija entre 4 opciones, agrupadas en 2 grupos, “Caso 1” y “Caso 2”, con las opciones “Mayo” y “Junio” en ambos grupos.

La diferencia es (además del nombre del selector) que, en 2, el selector tiene el atributo “multiple” con valor “multiple” (aunque es un atributo booleano), lo que permite que el usuario pueda elegir una o más opciones, a diferencia de en 1 donde sólo puede elegir una opción, por lo que visualmente el selector cambia y responde diferente.

**Ejercitación 4**

Archivos en carpeta:

* \ejercitacion4
  + \img
    - utn-frro-logo-recortado.jpg
  + **index.html**

**Ejercitación 5**

Archivos en carpeta:

* \ejercitacion5
  + \img
    - utn-frro-logo-recortado.jpg
  + Articulos.htm
  + Encabezado.htm
  + **index.html**
  + MasInfo.htm
  + Menu.htm