

# Práctica N°1- HTML

## Ejercitación 1 - Responder el siguiente cuestionario

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

HTML es el lenguaje de marcado principal de la World Wide Web. Originalmente, HTML fue diseñado principalmente como un lenguaje para describir semánticamente documentos científicos.

El origen de HTML se remonta a 1980, cuando el físico Tim Berners-Lee, investigador del CERN (Organización Europea para la Investigación Nuclear) propuso un nuevo sistema de “hipertexto” para compartir documentos.

Los estándares oficiales HTML son HTML 2.0, HTML 3.2, HTML 4.0, HTML 4.01 y HTML5. HTML5 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 HTML5.

<b>HTML 2.0</b>	En 1995 se publica, es el primer estándar oficial de HTML, no soportaba tablas. Se simplificaba al máximo la estructura del documento para agilizar su edición, donde la declaración explícita de los elementos body, html y head es opcional.
<b>HTML 3.2</b>	Se publicó en 1997 y es la primera recomendación de HTML publicada por el W3C. Esta revisión incorporó applets de Java y texto que fluye alrededor de las imágenes.
<b>HTML 4.01</b>	Se publicó en 1999. Desde la publicación de HTML 4.01, el W3C se centró en el desarrollo del estándar XHTML. En el año 2004, las empresas Apple, Mozilla y Opera mostraron su preocupación por la falta de interés del W3C en HTML y decidieron organizarse en una nueva asociación llamada WHATWG (Web Hypertext Application Technology Working Group) que comenzó el desarrollo del HTML 5, cuyo primer borrador oficial se publicó en enero de 2008. En marzo de 2007 el W3C decidió retomar la actividad estandarizadora de HTML, dentro del cual decidió integrar el XHTML.
<b>HTML 5, HTML 5.1, HTML 5.2</b>	W3C liberó el HTML 5 como estándar oficial en 2014. HTML 5 incorpora nuevos elementos no contemplados en HTML 4.01. Hay nuevas etiquetas, se introduce la posibilidad de introducir audio y video de forma directa en la web sin necesidad de plugins o complementos en los navegadores, y otras novedades. El W3C irá lanzando progresivamente nuevas evoluciones del HTML 5.

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

Recomendaciones para creación de documentos con HTML

### 1. Separar estructura y presentación

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.

## **2. 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. considerar representaciones alternativas de sus diseños.

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.

## **3. 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.

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 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.

Las definiciones de elementos y atributos indican claramente cuáles son desaprobados.

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 de referencia.

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

DTD es un documento SGML(Standard Generalized Markup Language) (ISO 8879: 1986) que incluye las reglas sintácticas para un tipo de documento específico. Incluye los elementos que se permiten y sus atributos, así como reglas que afectan a la anidación de los primeros y a los valores de los segundos. Contrastando un documento con su DTD se puede comprobar si este es válido o no.

HTML 4.01 especifica tres DTDs, que varían en cuanto a los elementos que soportan:

- 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.
- 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).

- El DTD HTML 4.01 para Documentos con Marcos (Frameset DTD incluye todo lo que incluye el DTD Transicional más los marcos.

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

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

Por ejemplo, para especificar el autor de un documento.

`<META name="Author" content="Dave Raggett">`

Esta especificación no define un conjunto de propiedades legales de metadatos. El significado de una propiedad y el conjunto de valores legales para esa propiedad debería estar definida en un diccionario de referencia llamado perfil. Por ejemplo, un perfil diseñado para ayudar a los motores de búsqueda a indexar documentos podría definir propiedades tales como "author", "copyright", "keywords", etc.

### Especificación de metadatos

Implica dos pasos:

1. Declaración de una propiedad y de un valor para esta propiedad. Esto puede hacerse de dos maneras:
  - Desde dentro de un documento, por medio del elemento META.
  - Desde fuera de un documento, vinculando los metadatos por medio del elemento LINK.
2. Referencia a un perfil en el que se definen la propiedad y sus valores legales. Para designar un perfil, se usa el atributo profile del elemento HEAD.

Al estar definido un perfil por el elemento HEAD, se aplica el mismo perfil a todos los elementos META y LINK de la cabecera del documento.

Los agentes de usuario no necesitan soportar los mecanismos de metadatos. Para aquellos que opten por soportar metadatos, esta especificación no define cómo deberían interpretarse los metadatos.

## Ejercitación 2:

2-a) La línea de código representa un comentario, no es obligatorio, sirve al usuario para aclarar detalles.

2-b) Un div sirve para crear secciones o agrupar contenido, crea un bloque, se coloca en la sección del body.

Tiene un id llamado bloque1.

2-c) Es una imagen, que mide 32x32.

falta especificar la ruta donde se encuentra el archivo.

2-d) Son etiquetas HTML que se incorporan en el encabezado de una página web y que resultan invisibles para un visitante normal, pero son de gran utilidad para navegadores que puedan valerse de esta información.

Su propósito es el de incluir información (metadatos) de referencia sobre la página.

2-e) Es un enlace el cual lleva a <http://www.e-style.com.ar/resumen.html>

2.f) Es una tabla 3 con título llamado Título.  
En la primera fila contiene A, B y C.  
En su primera columna 1° y 2°.

## Ejercitación 3:

3.a)

a) Todas las líneas de código son enlaces.

- 1) Enlace al home de google
- 2) target (destino): dónde se abre una nueva pestaña.
- 3) Es todo tag de apertura; no tiene contenido, no podemos ver nada; no se puede ir a la página porque no aparece el enlace.
- 4) Cuando se hace click vuelve a la misma página (href="#").
- 5) Enlace a una sección de la misma página.
- 6) Indica cuál es la sección "arriba" para referenciarla con #arriba, nos llevará hasta donde esté el enlace con el id de arriba.

b) Son todos párrafos.

- 1) Tiene una imagen y un "click aquí" con un enlace.
- 2) Texto "click aquí" fuera del enlace, la imagen es la que está enlazada
- 3) Dentro del enlace están la imagen y el texto; puedo hacer click en la imagen o en el texto.
- 4) Dos enlaces, uno con una imagen y otro con un texto. Se pueden poner páginas diferentes.

c) Listas.

- 1) Lista no ordenada (con viñetas).
- 2) Lista ordenada (con números).
- 3) Tres listas ordenadas, cada una con un solo item.
- 4) Blockquote: con sangría.

d) Tablas.

- 1) 3 filas y 2 columnas. Encabezados con th.
- 2) 3 filas y 2 columnas. Encabezados a mano.

e) Tablas: bgcolor --> atributo presentacional, color de fondo.

- 1) Caption: aparece como una fila más para poner el título.
- 2) El título aparece en la primera fila dentro de la tabla.

f) Tablas:

colspan --> una columna;  
rowspan --> una fila.

g) Tablas.

- 1) Tabla normal con bordes normales
- 2) Cellpadding y cellspacing: modifican los bordes.

#### h) Formularios.

1) id y name: iguales; action (obligatorio): adonde va la info cuando aprieto botón submit (a un programa en php; si está vacío el programa en php, lo que hace mailto es enviar la info a una dirección de correo específica); method: forma de enviar el form --> get: veo la info en la url, post: no veo la info en la url; target: dónde se recibe la respuesta del servidor (cartel en página nueva); enctype: tipo de encriptación.

fieldset: agrupa datos; legend: título de la agrupación; input: campo para ingresar datos; value: valor por defecto.

2) Usa etiquetas label asociadas al campo para ingreso de datos.

3) Botón reset: borra el contenido de los campos. Botón button no hace nada. Botón submit envía datos.

#### i) Botones.

1) Botón con imagen, no hace nada, abajo hay un click aquí.

2) Botón con el click aquí adentro.

#### j) Radio Button: id único, el name se puede repetir.

1) Mismo name: array de botones, se puede elegir uno de los dos.

2) Distinto name: dos array distintos, se pueden elegir los dos a la vez.

#### k) Listas --> optgroup: grupos de opciones.

1) Se ve una sola opción.

2) Lista múltiple: se ven varias opciones.