HTML5

Tags, Best Practice y Validators

Integrantes:

- Beccan, Constanza.
- Buzatto, Florencia.
- Aponte, Elifer.
- Garcete, Erica.
- Yusti, Zoe.

Introducción:

El presente trabajo pretende afianzar los conocimientos ya obtenidos y asentados sobre HTML. Mismos conocimientos que hemos obtenido durante la cursada en ADA ITW. Este trabajo es presentado por el grupo Nro 6 y su forma de presentación será en modalidad dinámica.

¿Qué es HTML?
HyperText Markup Language (lenguaje de marcas de hipertexto) es una forma de embeber texto plano. Este nació en 1980 a partir de una propuesta de un nuevo sistema de *"hipertexto"* para compartir documentos. Antiguamente este sistema se utilizaba en el ámbito informático para la visualización de documentos electrónicos. De cierta manera, los primitivos sistemas de *"hipertexto"* podrían asimilarse a los enlaces de las páginas web actuales.
El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre ***HTML Tags*** *(Etiquetas HTML*) y todavía hoy puede ser consultado online a modo de *reliquia informática* en el siguiente link : **https://tinyurl.com/oldhtmltags**

¿Cómo funciona el HTML?
Todo HTML se basa en un único concepto, las etiquetas. Estas son una forma de hablarle al navegador. Al crear un archivo con etiquetas e información, básicamente le estamos dando instrucciones al navegador de como mostrar dicha información. Con esto, si creamos una etiqueta diciendo “esto es un título” el navegador interpretará la información dentro de la etiqueta y mostrará la información en forma de título.

Etiquetas agregadas para HTML5:

Secciones:

<section>	Define una sección en un documento.
<nav>	Define una sección que solamente contiene enlaces de navegación
<article>	Define contenido autónomo que podría existir independientemente del resto del contenido.
<aside>	Define algunos contenidos vagamente relacionados con el resto del contenido de la página. Si es removido, el contenido restante seguirá teniendo sentido

<header>	Define la cabecera de una página o sección. Usualmente contiene un logotipo, el título del sitio Web y una tabla de navegación de contenidos.
<footer>	Define el pie de una página o sección. Usualmente contiene un mensaje de derechos de autoría, algunos enlaces a información legal o direcciones para dar información de retroalimentación.
<address>	Define una sección que contiene información de contacto.
<main>	Define el contenido principal o importante en el documento. Solamente existe un elemento <main> en el documento.

Agrupación de contenido:

<figure>	Representa una figura ilustrada como parte del documento.
<figcaption>	Representa la leyenda de una figura.

Semántica a nivel de texto:

<data>	Asocia un *equivalente legible por máquina* a sus contenidos. (Este elemento está sólamente en la versión de la WHATWG del estandar HTML, y no en la versión de la W3C de HTML5).
<time>	Representa un valor de *fecha* y *hora*; el equivalente legible por máquina puede ser representado en el atributo `datetime`.
<mark>	Representa texto resaltado con propósitos de *referencia*, es decir por su relevancia en otro contexto.
<ruby>	Representa contenidos a ser marcados con *anotaciones ruby*, recorridos cortos de texto presentados junto al texto. Estos son utilizados con

regularidad en conjunto a lenguajes de Asia del Este, donde las anotaciones actúan como una guía para la pronunciación, como el *furigana* Japonés.

<rt>	Representa el *texto de una anotación ruby* .
<rp>	Representa los *paréntesis* alrededor de una anotación ruby, usada para mostrar la anotación de manera alterna por los navegadores que no soporten despliegue estandar para las anotaciones.
<bdi>	Representa un texto que debe ser *aislado* de sus alrededores para el formateado bidireccional del texto. Permite incrustar un fragmento de texto con una direccionalidad diferente o desconocida.
<wbr>	Representa una *oportunidad de salto de línea,* es decir, un punto sugerido de envoltura donde el texto de múltiples líneas puede ser dividido para mejorar su legibilidad.

Contenido Incrustado:

<embed>	Representa un *punto de integración* para una aplicación o contenido interactivo externo que por lo general no es HTML.
<svg>	Define una *imagen vectorial* embebida.
<math>	Define una *fórmula matemática*.

Formularios:

<datalist>	Representa un *conjunto de opciones predefinidas* para otros controles.
<keygen>	Representa un control de *par generador de llaves.*

<output>	Representa el *resultado de un cálculo.*
<progress>	Representa el *progreso de finalización* de una tarea.
<meter>	Representa la *medida* escalar (o el valor fraccionario) dentro de un rango conocido.

Elementos Interactivos :

<details>	Representa un *widget* desde el que un usuario puede obtener información o controles adicionales.
<summary>	Representa un *resumen, título* o *leyenda* para un elemento <details> dado.
<command>	Representa un *comando* que un usuario puede invocar.
<menu>	Representa una *lista de comandos* .

Las Meta Etiquetas o Meta Tags son etiquetas de información que se añaden en el código html de cada página de una Web para aportar información relevante sobre la categorización de esa página web. Es una información que sólo podrá accederse a través del código fuente y que en realidad sólo resulta relevante para los buscadores, pero su finalidad es de gran utilidad ya que aportan gran información sobre la página como el autor, fecha, palabras clave, descripción, etc.

La calidad de una **meta tag** tiene dos aspectos importantes:

1. Como una persona evalúa la **meta tag**
2. Como un robot evalúa la **meta tag**

Para una persona, la **meta tag** necesita ser llamativa, interesante, informativa, curiosa y con un toque de "*call-for-action"* (instrucción que provoca una reacción inmediata).

Para un robot, la **meta tag** necesita de una dosis de palabras clave para que el algoritmo clasifique de que trata la página web.

Lo esencial en el desarrollo de la **meta tags** es encontrar un término medio entre estos dos puntos.

La Meta Etiqueta Title:
Estrictamente hablando, este no es ningún meta-tag, sino una etiqueta autónoma de HTML, aunque, debido a su significado a la hora de interactuar con los agentes de usuario es común que sea mencionada como parte de los metadatos. Actúa como título de la página en cuestión y debe estar formada por palabras clave presentes en la página web.

Codificación de caracteres:

Si la fuente no fue previamente definida en el header del archivo HTTP, es necesario hacerlo usando HTML. Así se evita, por ejemplo, que la ñ o las tildes no se muestren correctamente.

La Meta Etiqueta Description:

La etiqueta descripción o Meta Etiqueta Description actúa como una descripción de la página en cuestión y también debe estar formada por palabras clave y frases que resuman el contenido de la página web.

Esta información se muestra como snippet (una síntesis en dos líneas del tema de una página que aparece bajo la URL) en los buscadores de uso más generalizado como Google o Bing, por lo que se recomienda cuidar su redacción.

Palabras clave (Keywords)

Con esta etiqueta meta los administradores tienen la posibilidad de definir palabras clave para el buscador. Las keywords son aquellos criterios a los que responde un buscador para ofrecerle al usuario páginas HTML como respuesta, donde tales palabras clave son parte de los meta tags.

Autor (author) y copyright

Estos dos meta tags, de uso opcional desde el punto legal, permiten hacer referencia al diseñador de una página web y al propietario de los derechos del código fuente de una página HTML.

VIEWPORT

Básicamente, sirve para definir qué área de pantalla está disponible al renderizar un documento, cuál es nivel de escalado que puede realizar el usuario, así como si el navegador debe mostrarla con algún *zoom* inicial. Todo ello se indica a través de varios parámetros en la propia etiqueta META.

Lista completa de propiedades de viewport:

Atributo	Valores	Descripción
width	Valor integral (en pixeles) o constante device-width	Define el ancho del viewport
height	Valor integral (en píxeles) o constante device-height	Define el alto del viewport
initial-scale	Cualquier número real de 0.1 en adelante. 1 representa no escala.	Escala inicial del viewport
minimum-scale	Cualquier número real de 0.1 en adelante. 1 representa no escala.	Escala máxima del viewport
maximum-scale	Cualquier número real de 0.1 en adelante. 1 representa no escala.	Escala mínima del viewport
user-scale	"yes" / "no"	Permiso para que el usuario pueda hacer zoom

¿Cuales son las buenas prácticas en un Programador?

- ➔ Nunca perder el foco del proyecto.

Para esto podemos hacer un documento, un diagrama de flujo, mock ups y todas las herramientas que nos ayudan a esquematizar el proyecto. Para que durante el desarrollo no perdamos el objetivo o los elementos principales del mismo.

- ➔ Usar comentarios.

Por más tonto que parezca tu programa, siempre utilizar comentarios antes de cada conjunto importante de código ayuda a la hora de entender el programa. Mas si eres estudiante.

Ofuscación de código CSS

CSS Obfuscator es una forma única de proteger sus hojas de estilo

¿cómo funciona?
Este software transformará sus hojas de estilo de una manera, nadie querrá modificarlas. Pueden robarlo, pero ¿quién quiere usar (y ajustar) la hoja de estilo, que parece que está hecha del peor codificador del universo? ¿Lo tengo? ¡Intentalo!

Un ofuscador también minimiza, pero también hará modificaciones al programa, cambiando los nombres de variables, funciones y miembros, haciendo que el programa sea mucho más difícil de entender, y reduciendo aún más su tamaño en la negociación. Algunos ofuscadores son bastante agresivos en sus transformaciones. Algunos requieren anotaciones especiales en el programa de origen para indicar qué cambios pueden ser seguros o no.

Cualquier transformación conlleva el riesgo de introducir un error. Incluso si el ofuscador no causó el error, el hecho de que podría tener es una distracción que ralentizará el proceso de depuración. Las modificaciones al programa también aumentan significativamente la dificultad de depuración.

Después de minificar u ofuscar, debes usar GZIP. GZIP puede reducir aún más el tamaño del programa. GZIP es tan efectivo que la diferencia en la eficiencia entre la minimización y la ofuscación se vuelve insignificante. Así que prefiero la minificación con GZIP porque no

tengo tiempo para programar herramientas que puedan inyectar errores en buenos programas.

Bibliografía Web:

- https://developer.mozilla.org/es/docs/HTML/HTML5
- https://developer.mozilla.org/es/docs/HTML/HTML5/HTML5_lista_elementos
- http://librosweb.es/libro/xhtml/capitulo_1/breve_historia_de_html.html
- https://yuiblog.com/blog/2006/03/06/minification-v-obfuscation/
- https://www.1and1.es/digitalguide/paginas-web/desarrollo-web/los-meta-tags-mas-importantes-y-su-funcion/
- https://quiwiq.com/posicionamiento/importancia-meta-etiquetas-seo/952
- http://www.alhsis.com/las-meta-tags-y-su-importancia-en-el-seo/
- https://desarrolloweb.com/articulos/etiqueta-meta-viewport.html
- https://www.netvoluciona.es/blog/La-web-movil-y-la-utilidad-de-la-etiqueta--viewport--80