
Anatomía Web

<HTML>



Dustin Menjivar Cornejo

Edición 21.06

Año: 2021

Edición: 21.06

Autor: Dustin Menjivar Cornejo

Licencia: Creative Commons (by, nc, nc)

Índice

Requisitos.....	6
Opcionales.....	6
Introducción a <HTML>.....	7
Manejo de archivos.....	8
Estructura de un elemento.....	9
Etiqueta.....	9
Atributo.....	10
Atributo básico.....	10
Atributo booleano.....	10
Valor.....	10
Modelo caja contenedor.....	11
Caja en bloque.....	11
Caja en línea.....	11
Elemento padre e hijo.....	11
Estructura de contenido HTML.....	12
Estructura básica de un proyecto.....	13
Estructura básica un proyecto (código).....	14
Elementos del contenido HTML.....	15
Etiquetas de texto.....	18
Comentarios.....	18
Niveles de encabezado.....	18
Párrafos.....	19
Fechas.....	19
Citas.....	19
Modificadores de texto.....	20
Símbolos y Caracteres en HTML.....	22
Enlaces.....	24

Etiquetas de enlaces.....	24
Atributos de enlaces.....	25
Listas.....	27
Etiqueta de lista.....	27
Elementos de lista.....	27
Listas anidadas.....	28
Multimedia.....	29
Imágenes.....	29
Optimizando carga de imágenes.....	29
Video.....	33
Elementos de video.....	33
Atributos de video.....	33
Audio.....	35
Elementos de audio.....	35
Atributos de audio.....	35
Iframe.....	37
Figure y Figcaption.....	39
Tablas.....	40
Etiqueta de tabla.....	40
Elementos de una tabla.....	40
Atributos de celda.....	42
Agrupando columnas.....	42
Formularios.....	44
Etiquetas de formulario.....	44
Atributos de un formulario.....	44
Elementos de formulario.....	45
Atributos de los elementos del formulario.....	47
Atributos del input.....	49
Metadatos.....	53

Atributos de los metadatos.....	53
Open Graph Protocol.....	55
Fuentes.....	56
Mensaje para el lector.....	57

Requisitos

Ordenador o dispositivo móvil

Editor de código o IDE

- Para PC, sugerimos el programa visual studio code
- Para el móvil, sugerimos la aplicación Acode

Visualizador

- En el caso de usar PC, es un navegador (No recomendamos IE)
- En el caso de usar teléfono, puede ser un navegador o la misma aplicación hemos descargado

Para el desarrollo de este material, utilizaremos, el programa “Visual Studio Code” y la aplicación “Acode”. Si ya ha trabajado con un software diferente, sientase en libertad de usar el que le parezca más cómodo.

Opcionales

- Traductor de nuestro idioma a inglés (ej google translate)
- Curso de computación básica

Introducción a <HTML>

HyperText Markup Language o HTML, es el lenguaje de marcado utilizado para estructurar los proyectos y aplicaciones web.

El uso de HTML es un estándar, este lenguaje es supervisado y establecido por la W3C (World Wide Web Consortium), la cual se encarga de establecer, actualizar y definir las estructuras y las pautas sobre el uso del lenguaje.

La estandarización del lenguaje nos permite disponer y de mayor compatibilidad entre dispositivos y de ese modo, todo proyecto puede ser soportado en el dispositivo desde el cuál lo visualicemos, o en su defecto, disponer de una forma sencilla de hacerlo compatible con los dispositivos que no lo soporten, que a decir verdad hoy en día, es poco probable que no dispongamos de algún dispositivo que no sea compatible con este estándar web.

Cabe aclarar que HTML no es un lenguaje de programación en sí, sin embargo, es un lenguaje de marcado (maquetado o estructurado), el cual define la estructura de los proyecto web y sin este estandar sería complejo realizar un proyecto que sea compatible con la gran cantidad de resoluciones y dispositivos desde los cuales nos conectamos hoy en día.

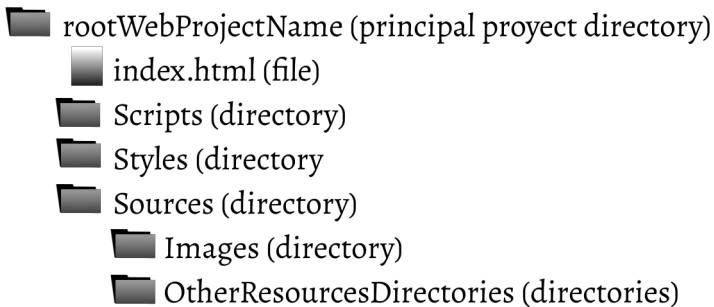
Manejo de archivos

Antes de comenzar debemos definir lo esencial, y es saber cómo será la organización de nuestros archivos y directorios.

Al organizar nuestros archivos obtendremos una estructura fácil de utilizar y escalar, debido a que haremos una estructura de tal modo que permita que el proyecto, pueda adquirir nuevas funcionalidades, directorios y archivos de una manera sencilla.

<!--Aconsejamos no utilizar el caracter espacio, guión alto o bajo para nombrar los directorios o archivos del proyecto.-->

Ejemplo de una estructura de manejo de archivos:



Estructura de un elemento

HTML se conforma en su totalidad por lo que entenderemos como etiquetas. A continuación este primer concepto.

Etiqueta

La etiqueta, también llamada elemento, caja, o contenedor, es un bloque de espacio definido en el DOM, al cual se le asigna el contenido que poseerá y de ese modo poder maquetarse.

Podemos obtener resultados similares con diferentes etiquetas, pero debemos aprender a utilizar la adecuada para cada situación, teniendo en cuenta su sintáctica y semántica.

Estructura general de una etiqueta

<code><tagName</code>	<code>attribute=</code>	<code>"value"></code>	<code>content</code>	<code></tagName></code>
Opening				Closing

Estructura particular de una etiqueta

<code><tagName</code>	<code>attribute=</code>	<code>"value"/></code>
Opening and Closing		

Atributo

Es una propiedad asignada a la etiqueta, un atributo puede aportar sentido semántico para que el elemento posea una mejor estructura, puede definir un nombre el cual será utilizado como referencia en el servidor, o puede ser una propiedad que simplemente cambie el aspecto visual del elemento, aunque por lo general realizaremos esto último con otro lenguaje.

Atributo básico

Es aquel que al ser declarado requiere un valor para poder aplicar la propiedad al elemento.

Atributo booleano

Es aquel que simplemente requiere ser declarado para aplicar su efecto en el elemento.

Valor

El valor es el parámetro que recibe un atributo, y su función es indicarle al atributo, la cualidad o dato específico a aplicar en el elemento.

Modelo caja contenedor

Antes de comenzar a aprender sobre los elementos de html, debemos conocer algo que nos ayudará a saber de qué forma estos serán maquetados.

Caja en bloque

Los elementos que tienen por defecto la propiedad en bloque, son aquellos los que utilizarán el ancho total de elemento padre, independientemente del tamaño del contenido que este posea, además que realizará un salto de línea.

Caja en línea

Los elementos que tienen por defecto la propiedad en en línea, son aquellos los que utilizarán únicamente ancho del tamaño del contenido que este posea, además estos elementos se maquetarán uno al lado del otro.

Elemento padre e hijo

En HTML, se hablará ocasionalmente de elemento padre y elemento hijo, esto se refiere sencillamente a un elemento que se encuentran dentro de otro.

Estructura de contenido HTML

Toda estructura de proyecto, que se trabaje con HTML, debe contener, únicamente, etiquetas y atributos con sus respectivos valores.

Los estilos, funcionalidad u otra característica extra, debe ser implementada posteriormente mediante archivos que posean ya sea sólo funcionalidad o sólo estilos (archivos externos o locales).

Los elementos utilizados al estructurar el código, en algunos casos es diferente a lo antes mencionado, esto es debido a que un elemento puede categorizarse como general o vacío.

<tagName attribute= "value"> content </tagName>

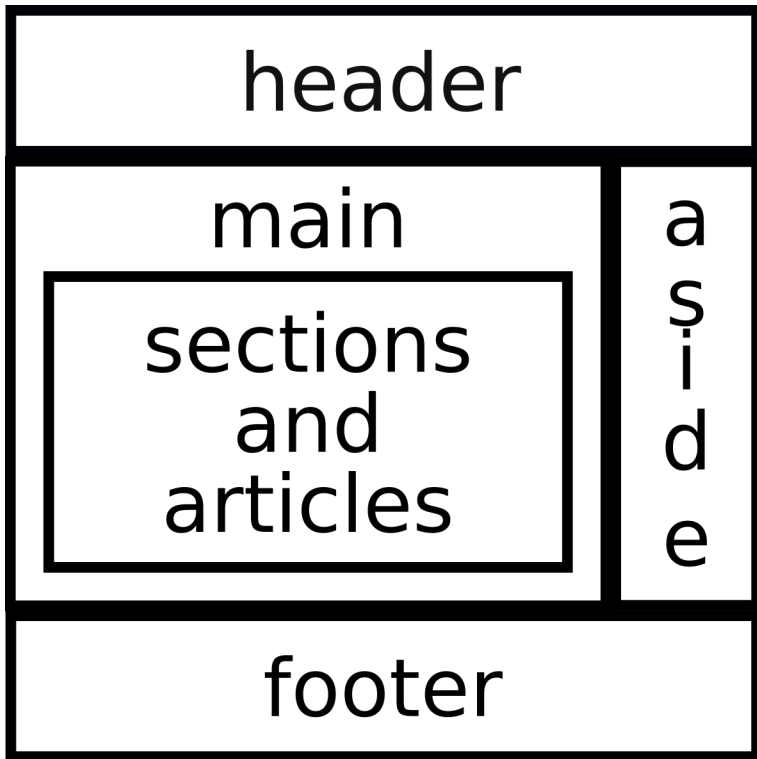
Elemento general. Son los elementos más comunes de encontrar, pues de la totalidad de etiquetas en HTML, este grupo son su gran mayoría.

<tagName attribute= "value" />

Elemento vacío. Son los elementos que no posee contenido y además, no poseen etiqueta de cierre y en ocasiones, ni atributos.

Estructura básica de un proyecto

Ejemplo de una estructura web visible por el usuario final.



<!--Podemos utilizar artículos dentro de secciones, como viceversa, solamente debemos tomar en cuenta el contexto en el cual los utilizemos-->

Estructura básica un proyecto (código)

```
<!doctype html>
<html lang= "languageAbrebiation" >
<head>
    <!--metaViewport and metadataTags-->
    <!--sourcesLinksTags and Icons links -->
    <title>ProjectName</title>
</head>
<body>
    <header></header>
    <nav></nav>
    <main>
        <!--Sections and Articles-->
    </main>
    <aside></aside>
    <footer></footer>
    <!--scriptsLinks -->
</body>
</html>
```

Elementos del contenido HTML

Todo documento HTML, posee una estructura básica, a continuación se explica la función de cada uno de ellos.

<!doctype html>

Estandar del documento. Etiqueta, su función es informarle al visualizador, que trabajaremos el documento con el estandar del lenguaje HTML 5.

<html> </html>

HTML. Etiqueta la cuál indica que, dentro de las mismas, se encontrará toda nuestra estructura y contenido del proyecto.

<head> </head>

Cabeza del documento. Esta etiqueta poseerá información de utilidad para el navegador.

<meta />

Etiqueta de metadatos (Uso explicado en su respectivo tema).

Enlaces a recursos (Usos explicado en su respectivo tema)

<body> </body>

Cuerpo del proyecto. Posee el contenido visible (maquetado) de nuestro sitio web.

<header> </header>

Encabezado del proyecto. Elemento que indica el área del encabezado de nuestro sitio, por lo general contine el logo de nuestro sitio y el nav.

<nav> </nav>

Navegación principal. Etiqueta que indica el área donde se encuentra el menú de navegación y sus enlaces.

<main> </main>

Contenido principal del sitio (información relevante al usuario).

<section> </section>

Sección. Sección de contenido relacionado al elemento padre.

<div> </div>

Contenedor simple. Contenedor sin semántica.

<article> </article>

Artículo. Sección de contenido independiente; Podemos visualizar este elemento independiente al sitio, y aún así no perderá su significado.

** **

Elemento simple. Elemento sin semántica

<aside> </aside>

Contenido secundario. El contenido dentro de este elemento no es relevante para el sitio web, o es complementario, pudiendo ser desde un menú secundario, hasta publicidad.

<footer> </footer>

Pié de página. Contiene información complementaria, ya sea derechos de autor, información de contacto u otra información de interés para el usuario.

Enlaces a recursos “script”. Por lo general, se enlazan los JavaScript (Su uso estará explicado en su respectivo tema)

Etiquetas de texto

Comentarios

`<!--textToBeCommented-->`

Los comentarios se utilizan generalmente para documentar nuestro código, estos no afectan el contenido del sitio, debido a que estos son, simplemente, textos de apoyo (anotaciones). Las anotaciones serán visualizadas únicamente por el programador.

Niveles de encabezado

`<h#> titleText </h#>`

(La almohadilla representa el número del nivel de encabezado).

Las etiquetas de encabezado, son utilizadas para indicar un título en nuestra página. La etiqueta, sólo admite un valor entero entre uno y seis. Además sólo podremos tener un único “h1” por página.

Este elemento es utilizado para definir los títulos ya sea de secciones principales, sub secciones o artículos.

Párrafos

<p> paragraphText </p>

La etiqueta de parrafo es la más útil, debido a que con ella podremos definir la gran mayoría de textos en nuestros proyectos

Fechas

<time datetime="dateOrHourOrBoth">dateOrHour</time>

Las fechas las podemos utilizar para indicar fechas, horas o ambas, útil cuando poseemos artículos o secciones las cuales presentan como un dato de interés una fecha u horario

Citas

Las citas son de apoyo cuando utilizamos referencias de libros, individuos, sitios web, personajes u otras fuentes.

<blockquote> appointmentText </blockquote>

Definimos una cita, la etiqueta es en formato bloque

<q> appointmentText </q>

Definimos una cita, la etiqueta es en formato en línea

<cite> citationAuthor </cite>

Autor de cita. Este modificador se agrega junto a la cita, útil para nombrar al autor correspondiente del texto citado

Modificadores de texto

Los modificadores deben estar dentro de alguna etiqueta de texto, estos modificadores, dentro de su cuerpo, tendrán el texto el cual se desea modificar.

** textToBeBold **

Negrita. (modificador visual)

** textToBeStrong **

Modificador similar a negrita, con la característica de indicar al navegador que el texto será relevante para el motor de búsqueda.

<i> textToBeItalic</i>

Itálica o cursiva. (modificador visual)

textToBeEmphasized

Énfasis. Modificador similar a itálica o cursiva, con la característica de indicar al navegador que el texto será relevante para el motor de búsqueda, pero un poco menos que el texto dentro de un

<small>text</small>

Pequeño. Texto menos relevante que un texto normal.

<pre>textToBePreformatted</pre>

Texto preformateado. El texto que tipeemos dentro del elemento se visualizará exactamente de la misma manera que fué escrito, útil por ejemplo cuando queramos mostrar ejemplos de código.

<code>codeText</code>

Texto de código. El texto dentro del elemento corresponde a texto de código, se puede utilizar junto al texto preformateado.

**
**

Salto de línea.

<wbr/>

Salto de línea predefinido. Realiza un salto de línea, sólo cuando este sea necesario. El guión '-', también cumple esta función.

^{text}

Superíndice. Un ejemplo es en ecuaciones matemáticas.

_{text}

Subíndice. Un ejemplo descripciones de compuestos químicos

<hr/>

Línea horizontal. Inserta una línea horizontal en el maquetado (utiliza el ancho completo).

<strike>textToBeStriked</strike>

Tachado. El texto dentro de esta etiqueta, obtendrá un tachado mediante una línea horizontal en el centro del texto

Símbolos y Caracteres en HTML

Para mostrar caracteres especiales, recurriremos a su código.

Utilizando códigos ascii

La primera forma de agregar simbolos en nuestros textos es mediante la utilización de la combinación de teclas y el código ASCII del simbolo correspondiente

Símbolos mediante código hexadecimal

Esta forma es la forma alternativa a agregar simbolos mediante combinación de teclas y el código ASCII, este método por lo general es utilizado en ordenadores con gnu/linux

Símbolos explicito en HTML mediante códigos ASCII.

En esta forma definimos una forma explicita para incluir un simbolo en nuestros textos, esta forma tiene la ventaja de mostrar el simbolo que exactamente necesitamos.

&#asciiCode;

Estructura para insertar un simbolo explicitamente mediante un código ascii (En ocaciones puede no funcionar)

<!--Para ver otras formas de representar símbolos en HTML, podemos investigar “html símbolos y caracteres”-->

Enlaces

Etiquetas de enlaces

` textDisplayedToUser `

Enlace simple (Utilizado para enlazar a sitios web o archivos)

` textDisplayedToUser `

Enlace hacia un elemento del sitio

` textDisplayedToUser `

Enlace hacia un elemento en un sitio o página diferente.

**`<link rel="icon" type="image/imageExtentionType"
href="sourceRoute/fileName.imageExtentionType ">`**

Enlace a ícono. Ícono mostrado en la pestaña del visualizador o mostrado al instalar nuestro sitio mediante pwa. Recomendamos enlazar varios tamaños (16x16, 32x32, 57x57, 60x60, 72x72, 76x76, 96x96, 114x114, 120x120, 144x144, 152x152, 180x180, 192x192) tener en cuenta que la extensión de la imagen sea soportada.

<link rel= "stylesheet" href= "sourceRoute"/>

Enlazar recursos de estilos (CSS)

<script src= "sourceRoute" > </script>

Enlace a un archivo JavaScript.

Atributos de enlaces

En “href” el valor de una almohadilla '#', es utilizado para indicar que no se accederá a ninguna dirección.

./sourcesDirName/fileName.extention

Cuando el enlace dirija hacia nuestro sistema, la ruta puede ser relativa al directorio actual, o desde el directorio raíz.

https://absoluteSourceRoute

Si el enlace envía hacia una web, debe poseer la ruta absoluta (URL) y esta debe incluir su respectivo protocolo. Por cuestiones de seguridad es aconsejable no utilizar direcciones que no posean el protocolo **https://**

** textDisplayedToUser **

Atributo **target= “_blank”**, es utilizado para indicar que el enlace debe abrirse en una nueva pestaña.

Atributo **rel= “noopener”**, Es útil cuando se abren enlaces que no son de confianza, para indicar que no puedan alterar el documento de origen a través de la propiedad `Window.opener`

Atributo **rel= “norereferrer”**, Evita que el navegador, envíe la dirección de página o cualquier otro valor al navegar a otra página, como referente a través del encabezado `Referer: HTTP`.

** textDisplayedToUser **

Descargar. Este atributo booleano, tiene la función de permitir descargar un archivo con ruta relativa. (funciona cuando el sitio está ejecutándose en el servidor). Si agregamos un valor, este será el nombre del archivo que descarguemos.

El uso de **target = “_blank”** sin **rel = “noopener”** y **rel = “norereferrer”** hace que el sitio web sea vulnerable a los ataques de explotación a través de la API de **window.opener**

Listas

Etiqueta de lista

** listItems **

Lista no numerada (Lista no ordenada)

** listItems **

Lista numerada (Lista ordenada)

<dl> listTermsAndDefinitions </dl>

Lista de definiciones, un uso podría ser al hacer un diccionario

Elementos de lista

** listItemValue **

Agregar un nuevo elemento a la lista (ordenada o no ordenada).

<dt> listTermItem </dt>

Para agregar elementos (o términos) a una lista de definiciones, utilizaremos la siguiente etiqueta.

<dd> listTermDefinition </dd>

Agregar una descripción al término de la lista.

Listas anidadas

Las listas anidadas son, simplemente, listas que estan dentro de un elemento “li” de otra lista, ya sea ordenada o no ordenada. Las listas anidadas son, en general, una combinación de listas ordenadas y listas no ordenadas.

<listTag>

** listItems **

<listTag> listItems </listTag>

</listTag>

Multimedia

Imágenes

Al trabajar con imágenes los formatos que nos podremos encontrar son: PNG, GIF, WEBP y JPEG o JPG, aunque el más recomendado es el formato WEBP.

```

```

Estructura básica para incrustar imagen a nuestra web

Optimizando carga de imágenes

Este tema puede ser estudiado posteriormente. No obstante, no hay inconveniente alguno, en estudiarlo en este momento.

srcset= "sourceRoutes"

Conjunto de rutas del recurso. Atributo con el cuál podemos indicar al visualizador, que por ejemplo, en función del tamaño del visualizador "Viewport", cargue una u otra imagen

**srcset= “ sourceRoute1/ imageName.extention,
sourceRoute2/ imageName.extention,
sourceRoute3/ imageName.extention”**

En el atributo, agregamos las diferentes rutas de las imágenes y junto a ellas, debemos agregar una valor (lo veremos en breve)

En ocasiones un visualizador no soporta un tipo de extensión en concreto, para solucionar esto, incluimos recursos de respaldo, con extensión de archivo diferente

**srcset= “ sourceRoute1/ imageName.extention,
sourceRoute1Backup1/ imageName.extention1,
sourceRoute1Backup2/ imageName.extention2,
sourceRoute2/ imageName.extention,
sourceRoute2Backup1/ imageName.extention1,
sourceRoute2Backup2/ imageName.extention2”**

Para indicar al conjunto de rutas del recurso, qué ruta deseamos tome, en un momento dado, debemos agregar un parametro extra en la ruta del recurso, para ello tenemos dos opciones:

Utilizar srcset mediante DPR

Indicarlo mediante el DPR “Device Pixel Ratio” (la almohadilla indica el valor del DPR)

srcset= “ sourceRoute1/ imageName.extention #x”

Utilizar srcset mediante ancho del viewport

Indicarlo mediante el ancho de la pantalla. (El valor de los píxeles los podemos modificar a nuestro criterio o conveniencia)

srcset= “ sourceRoute1/ imageName.extention”

media=“width: 320px,

sourceRoute1/ image2Name.extention”

media=“width: 360px,

sourceRoute1/ image3Name.extention”

media=“width: 480px”

Al trabajar con srcset mediante el viewport width, debemos agregarle un atributo extra a la imagen, el cual nos sirve para indicar al visualizador, en que momento deberá cargar la imagen

sizes= “(max-widt:320px) 320w,

(max-widt:360px) 360w,

(min-widt:361px) 480w”

loading= “lazy”

Atributo extra para una imagen. Lo que nos permite es indicarle al visualizador que sólo cargue la imagen cuando se este próximo a encontrarla, de lo contrario, el visualizador no la cargará

Ejemplo de la estructura utilizando el ancho del viewport

```

```

<picture></picture>

Contenedor opcional, este elemento lo podremos utilizar para indicar más parametros para que una imagen sea cargada

```
<picture>
    <source srcset= “sourceRoute” media= “(max-width:
screenWidthValue)” />
    <img src= “defaultImageRoute”/>
</picture>
```


Video

<video> videosSources </video>

Estructura base, contiene las rutas del video con distinta extensión, esta etiqueta la podremos usar con archivos locales.

Elementos de video

<source src= "videoRoute" type= "video/videoExtention">

Añadir un recurso de video, deberíamos incluir varios tipos de formatos del mismo archivo para aumentar la compatibilidad con los diferentes tipos de visualizadores

Atributos de video

autoplay

Reproducción automática. Atributo booleano

controls

Atributo booleano que activa los controles para el usuario.

loop

Repetir video. Atributo booleano que indica que el video, una vez finalizado vuelva a reproducirse

muted

Silenciar el video. Atributo booleano

poster= “sourceRoute”

Miniatura. Agregar una imagen de miniatura al video

preload= “propertyValue”

Indica qué deseamos precargar del video, para el usuario

Valores que puede definirse en el atributo precargar.

- **none**

No precargar ninguna información

- **metadata**

Sólo los metadatos

- **auto**

Permitir precargar el video

Audio

<audio> audioSources </audio>

Estructura base, contiene las rutas del audio con distinta extensión, esta etiqueta la podremos usar con archivos locales.

Elementos de audio

<source src="sourceRoute" type= "audio/audioExtention">

Añadir un recurso de audio, deberíamos incluir varios tipos de formatos del mismo archivo para aumentar la compatibilidad con los diferentes tipos de visualizadores

Atributos de audio

autoplay

Reproducción automática. Atributo booleano

controls

Controles. Atributo booleano que indica la activación de los controles para el usuario.

loop

Repetir audio. Atributo booleano que indica que se debe reproducir nuevamente el audio, una vez finalizado

muted

Atributo booleano . Silenciar el audio. Atributo booleano

poster= “sourceRoute”

Atributo booleano . Agregar una imagen de miniatura al audio

preload= “propertyValue”

Indicar qué deseamos precargar del audio, para el usuario, el valor puede tomar diferentes valores

Valores de la propiedad precargar.

- **none**

No precargar ninguna información

- **metadata**

Sólo los metadatos

- **auto**

Permitir precargar el audio

Iframe

Elemento utilizado, generalmente. para embeber contenido de otros sitios web, ya sea videos, mapas u otro elemento, por lo general no tendremos que estructurar este código debido a que los sitios, ya nos lo proporcionan.

<!--Se recomienda no utilizar demasiados iframe debido a que tienen un fuerte impacto en el uso de recursos-->

<iframe src="url"></iframe>

Estructura básica de la etiqueta.

Si deseamos embeber videos de youtube a nuestro sitio web recomendamos utilizar el código siguiendo el ejemplo que mostramos a continuación (Nos será útil para luego controlar el video con JavaScript)

De nuestra url del video, solamente necesitaremos el id del mismo, el cual, generalmente el el texto que se encuentra luego de la palabra **watch=**

Como ejemplo, tenemos la siguiente dirección de youtube, que obtenemos mediante la forma común de copiar la url de un video

<https://www.youtube.com/watch?v=jNQXAC9IVRw>

Entonces siguiendo el ejemplo **OurVideoId= jNQXAC9IVRw**

Entonces, nuestro iframe debería verse de la siguiente manera

```
<iframe src="https://www.youtube.com/embed/OurVideoId?  
enablejsapi=1&html5=1" allowfullscreen=""  
frameborder="0"></iframe>
```

El texto **?enablejsapi=1&html5=1** es un atributo el cual nos habilita el uso de la api iframe de youtube y de esa manera poder controlar el video mediante el uso de scripts con JavaScript

<!--Lamentamos el no poder traer más contenido sobre este tema. Si es necesario, en ediciones posteriores de este manual se agregará mas contenido sobre el uso del elemento "iframe"-->

Figure y Figcaption

Elemento figure es utilizado para contener multimedia, poemas, citas, código de algún lenguaje u otras funciones y usos.

El figure también se puede explicar, como un elemento que posee contenido relacionado opcional, lo que quiere decir, que si el contenedor padre posee, o no, al figure, este no perderá sentido y siempre transmitirá su idea de forma completa.

<figure>

mainMultimediaContent

<figcaption>Content</figcaption>

</figure>

<figcaption>complementaryContent</figcaption>

Contenido complementario. Utilizado para agregar información de interés o información complementaria relacionada con el contenido del “figure”, por lo general es un texto que complementa al contenido del “figure”.

Tablas

Las tablas, como se puede presuponer, las utilizaremos únicamente para la tabulación de datos de una manera ordenada y bien estructurada.

Etiqueta de tabla

`<table> tableContent </table>`

Elementos de una tabla

`<tr> tableColumns </tr>`

Fila. Agregar una fila a la tabla

`<td> cellData </td>`

Datos de tabla. Agregar un dato de tabla a la fila

En la tabla, es necesario agregar información de utilidad, para reconocer sobre qué tratan los datos mostrados. Añadiremos dicha información con las siguientes etiquetas:

<caption> describedTableTitle </caption>

Título de tabla. Describe el contenido que contiene la tabla, esta etiqueta se utiliza como primer hijo de la tabla.

<thead> headerCells </thead>

Encabezado de la tabla. Ubicada luego del título de la tabla, esta poseerá las celdas de encabezado.

<th> nameOfValueType </th>

Celdas de encabezado. Estas celdas, poseen el nombre de los datos que la columna contendrá.

<tbody> dataRowsAndColumnsCells </tbody>

Cuerpo de tabla. Dentro de este contenedor, tendremos todos los datos que tabulemos.

<tfoot> tableRowAndTableData </tfoot>

Pié de tabla.

Atributos de celda

rowspan= " intValue "

Tamaño de celda. Especificar a una celda, qué cantidad de filas utilizará (Se expande verticalmente).

colspan= " intValue "

Tamaño de celda. Especificar a una celda, qué cantidad de columnas utilizará (Se expanden horizontalmente).

Agrupando columnas

Visualmente tenemos una tabla completa y esta correcta, pero semánticamente, solamente tenemos un conjunto de filas con celdas. Para agregar la semántica correspondiente a las columnas, debemos utilizar un par de elementos los cuales agrupan las celdas en forma de columna. Añadir estos elementos es totalmente opcional, pero ayuda a tener mejor semántica en la tabla

<colgroup> tableColumns </colgroup>

Indicando el agrupado de columnas. Esta etiqueta es hija de la tabla (dentro de ella posee los elementos de columna de la tabla)

<col/>

Columnas de la tabla. Con este elemento indicamos la cantidad de columnas semánticas que poseemos en nuestra tabla.

Debemos utilizar una etiqueta por cada columna semántica que poseamos en la tabla.

span= “ numberOfColumns “

Si necesitamos que una columna semántica utilice más de una columna visual, utilizaremos el siguiente atributo en el elemento columna de tabla (ingresar un valor entero).

Formularios

Etiquetas de formulario

`<form> formContent </form>`

Estructura base de la etiqueta de formulario.

Atributos de un formulario

`action= "sourceRoute"`

Atributo del formulario con el cual indicamos el elemento el cuál se encargará de procesar los datos del formulario, por lo general es la url de un archivo que se encuentra en nuestro servidor.

`method= "methodType"`

Atributo del formulario, definido luego del “action”, utilizado para enviar dichos datos. (hay dos métodos).

- POST
- GET

Elementos de formulario

Todas (o la gran mayoría) de las entradas de datos en un formulario, serán recibidas mediante el mismo tipo de etiqueta. Para diferenciar el tipo de dato ingresado, definiremos el valor del tipo de dicha entrada.

```
<input id="inputIdName" type="inputType" name=  
"referentialNameOfTheInput" value="inputValue">
```

Estructura base de la etiqueta de entrada de datos. Tener en cuenta que depende de la funcionalidad que utilizaremos con la entrada, podremos omitir o agregar alguno de los atributos

```
<label for="inputIdName"> labelText </input>
```

Para agregar un texto el cual estará relacionado a la entrada, podremos usar un elemento, el cual debe poseer un atributo “for”, y su valor deberá ser el identificador de la entrada, así las relacionaremos entre sí.

```
<select> optionsItems </select>
```

Lista de selección

<option value="optionValue" > optionText </option>

Agregar un nuevo elemento (item) a la lista de selección

multiple

Atributo booleano. Para indicar que podemos hacer selección múltiple, añadimos el atributo “multiple” a la lista desplegable

<optgroup label= “text” > optionsToBeGrouped </optgroup>

Grupo de opciones. Al poseer listas de selección extensas, podemos agrupar sus opciones para facilitar su comprensión, con el atributo “label” agregamos el texto referente al grupo

<datalist list= “inputListAttributeValue” > optionsItems

</datalist>

Lista de datos. Lista de selección con filtro por coincidencia de caracteres. Para su uso requiere una entrada de tipo lista y con el atributo lista

Las opciones dentro de la lista de datos, son estructuradas de la misma forma que la lista de selección (opciones y grupos de opciones), debido a ello no creemos necesario repetir su explicación (Si desea la agregamos en siguientes versiones)

<fieldset> groupedFormElements </fieldset>

Agrupar elementos dentro de un formulario.

<legend> fieldSetTitle </legend>

Agregar un texto que indique el contenido del “fieldset”

<textarea> </textarea>

Área de texto. Área para que el usuario ingrese texto, lo podemos utilizar como cuerpo del mensaje cuando hacemos un formulario de contacto

<button> </button>

Botón. Generalmente utilizado añadir funcionalidad mediante JavaScript, el tema de funcionalidad mediante JavaScript lo podemos aprender con nuestro manual y curso de JavaScript

Atributos de los elementos del formulario

required

Para cualquier tipo de dato que el usuario ingresará, podremos indicarle que es un dato requerido.

placeholder= “textDisplayedToUser”

Marcador de posición. Agrega un texto similar al label, este tiene la ventaja de estar como contenido, y además se auto elimina al seleccionar la entrada (sería como un texto fantasma). Útil en las entradas de cadenas de caracteres.

readonly

Sólo lectura. La entrada debiese poseer un valor establecido antes de aplicar este atributo. El valor de la entrada, se enviará al servidor, pero el valor no podrá ser modificado.

disable

Sólo lectura y deshabilitado. La entrada debiese tener un valor antes de aplicar este atributo. El valor de la entrada no se enviará.

min= “value” max= “value”

Atributos para definir el rango mínimo y máximo de una entrada numérica (ingresar un valor entero)

minlength= “value” maxlength= “value”

Atributos para definir el mínimo y máximo de caracteres en una entrada tipo cadena de texto (ingresar un valor entero)

autofocus

Foco. Con este atributo simulamos un click del usuario en una entrada específica. Se activa al cargar el formulario, su función es que nos dirige directo para rellenar el campo indicado

Atributos del input

value= "inputTextValue"

Los elementos, ya que son elementos de entrada, tendrán valores o textos que correspondan al tipo de dato que la entrada requiere, si deseamos modificarlo podemos utilizar el siguiente atributo.

Placeholder. Se recomienda utilizar siempre que se nos sea posible, esta forma de indicar el tipo de entrada, en lugar de mostrar un texto con el atributo valor.

(Este atributo sólo lo recordaremos, debido a que ya ha sido explicado previamente en este tema).

(Se encuentra en el tema atributos de los elementos del formulario, puede regresar un poco para encontrar su uso)

type= "inputType"

Para las entradas, lo más importante que debemos conocer es su tipo, pues con ello, identificaremos qué tipo de dato recibiremos.

Valores posibles de la propiedad en base al dato a recibir:

type= "text" Texto

type= "password" Contraseña

type= "number" Número

type= "tel" Número de teléfono

type= "email" Correo electrónico

type= "date" Fecha

type= "time" Tiempo, Tiempo en formato hora

type= "month" Mes

type= "week" Semana

type= "url" Dirección web

type= "checkbox" Casilla de selección (una o más)

type= "radio" Casilla de selección única.

<radiogroup> radioInputs <\radiogroup> Es importante que las casilla de selección única, estén agrupadas mediante este elemento para que tengan la funcionalidad esperada

type= "file" Cargar archivos. Requiere funcionalidad con JS.

type= "color" Recuadro para seleccionar un color

type= "range" min= "minValue" max= "maxValue" step= "valuePerStep" Range, muestra un "slider" donde podremos seleccionar uno de los valores definidos.

type= "submit" value= "text" Agrega un botón que envía los datos ingresados en el formulario hacia el servidor

type= "button" value= "text" Botón. Mostrar un botón (utilizado para implementar funcionalidad con JS)

type= "reset" Reiniciar. Entrada que tiene la función de restablecer los valores de los campos del formulario.

maxlength= "maximumInputLength" Atributos que ayudan a definir un límite máximo de caracteres para las entradas, podremos utilizar el siguiente atributo

name= "inputVarName" Si trabajaremos con servidores, necesitamos enviar los datos e indicarle una referencia sobre qué tipo de información está recibiendo, lo cual es solucionado mediante el siguiente atributo

<!--Tomemos en cuenta que al trabajar con formularios que se conectarán a servidores, los elementos del formularios deberían poseer los atributos de nombre y valor-->

Metadatos

Los metadatos es información opcional (pero muy recomendada)

<meta attribute="value"/>

Estructura básica de la etiqueta de metadatos

<meta name="viewport" content="width=device-width, initial scale=1, user-scalable=no" />

Uno de los más importantes metadatos. Indica que la página se ajuste al ancho de la pantalla, opcionalmente agregamos los atributos **minimum-scale=1, maximum-scale=1**.

Atributos de los metadatos

charset= "characterCodification"

Charset. Utilizado para indicar la codificación de caracteres de la página, por lo general se utilizará "utf-8"

name="metadateType"

Nombre. Utilizado para indicar el tipo de metadato a trabajar

content="nameAttributeDescriptionText"

Contenido. Este atributo se encontrará junto a cualquier etiqueta name, su función es poseer el texto o palabras que describen al atributo name

Valores del atributo "name" (tipo de metadato)

name="copyright"

Derechos. Indicaremos que una página está bajo el derecho de alguna entidad.

Indicar que la página sea indexable o no indexable. (Permite que los motores de búsqueda encuentren una página en la web)

- **name="robots" content="index"**
- **name="robots" content="noindex"**

name="description" content="pageDescriptionText"

Descripción de nuestro sitio web. Realizar una descripción del sitio con un máximo (aproximado) de 170 caracteres.

name=" author " content=" authorName "

Nombre del autor, empresa o propietario del sitio web.

name=" keywords " content="pageKeywods"

Palabras claves acerca de nuestro sitio web.

Open Graph Protocol

Open graph protocol, es una herramienta creada por los desarrolladores de facebook, para poder estandarizar la elementos de un sitio al ser compartida su dirección.

Este protocolo nos permite es que al compartir una dirección con un conjunto de datos como por ejemplo un título, una imagen y una descripción, previamente definida por nosotros

Este protocolo es de uso opcional (Si gustas en versiones posteriores agregamos y explicamos su uso en el manual)

El sitio oficial para saber como usar este protocolo es:

<https://opengraphprotocol.org> (está en inglés)

Fuentes

Toda versión y mejora posterior de este manual, será realizada en base a la retroalimentación y apoyo que nuestra comunidad nos aporte, por lo tanto, tendrá información a la vanguardia y de alta calidad

La version 21.06 de este manual utilizó las fuentes:

- DorianDesings https://www.youtube.com/watch?v=eesyGnJwfAY&list=PLROIqh_5RZeB92ME1GFyeqDVOa-gLoYbd
- FalconMasters https://www.youtube.com/watch?v=cqMfPS8jPys&list=PLhSj3UTs2_yVHt2DgHky_MzzRC58UHE4z
- Fazt <https://www.youtube.com/watch?v=rbuYtrNUxg4>
- SoyDalto <https://www.youtube.com/watch?v=kN1XP-Bef7w>
- yacklyon https://www.youtube.com/watch?v=PvTRboH_iqg&list=PLg9145ptuAijps8rns9muRGmxlhO1vOe1
- yacklyon <https://www.youtube.com/watch?v=Qla9orIkOFc&list=PLg9145ptuAijj9GoHPTcYT8IoQAOc hM-n>

Mensaje para el lector.

Nosotros, Kixne™, nos preocupamos en poder entregarle la mejor información posible, por ende, nos esforzaremos por mantener todo nuestro contenido, a la vanguardia y con alta calidad.

Nuestro contenido es gratuito. Pero si nos desea apoyar de alguna manera, para poder continuar impartiendo conocimiento, puede hacerlo de diversas formas, ya sea mediante sugerencias, mediante aportes utilizando nuestros contactos y sitios oficiales, o si gusta, puede observar más formas de apoyo en nuestro sitio

<https://kixne.github.io/web/spanish>

El desarrollo del conocimiento, es una de las decisiones más importantes para todas las personas. Le agradecemos que nos haya elegido, y nos haya confiado su desarrollo personal y profesional. Esperamos poder volver a tenerlo junto a nosotros.

Le deseamos un buen día. ¡Hasta pronto!