# 1. Introducción Universidad Desarrollo Web

Página Web complementaria: https://www.w3schools.com/html/default.asp

1. ¿Qué es internet y cómo funciona?

Internet nos permite compartir información entre computadoras. Podemos tener equipos que contienen información y equipos que consultan información.

Los sitios web los podemos ubicar de dos maneras: Nombre del sitio web, Dirección de IP (Internet Protocol). La IP es una combinación de números que nos permite recuperar la información que contiene el servidor, de igual manera funciona poner el nombre del sitio web. DNS Checker (Domain Name Service: es un servicio para encontrar el nombre de los equipos), es un sitio que nos devuelve el IP según el nombre que le damos.

La forma en la cual se envía la información entre el servidor y los dispositivos normalmente es información de tipo HTML (HyperText Markup Language).

2. Herramientas a utilizar

-Google Chrome

-Visual Studio Code

3. Extensiones Visual Studio Code

-Prettier

-Live Server

-Path Intellisense

-HTML5

4. Estructura Básica HTML

Con la etiqueta <!DOCTYPE html> estamos indicando que estamos trabajando con la última versión de HTML.

5. Ejecutar un archivo HTML

URL significa Uniform Resource Locator. Podemos abrir nuestros archivos HTML con live server o directamente desde el archivo que creamos.

6. Etiqueta Meta y Charset

Con la etiqueta meta y el atributo charset vamos a especificar el tipo de juego de caracteres que vamos a poder desplegar y utilizar en nuestra página web.

Lo que modifica a la etiqueta se lo conoce como atributa y lo que se contiene entre las etiquetas se lo llama valor de la etiqueta.

7. Generación Automática de HTML

Visual Studio Code nos da una herramienta para generar un esquema similar al visto en index.html de manera automática. Agregando un ! al comienzo del documento y tabulando nos va a generar la estructura básica.

<meta http-equiv="X-UA-Compatible" content="IE-edge"/> || Nos agrega esta extensión para que soporte HTML en navegadores de tipo internet explorer.

<meta name="viewport" content="widht=device-width, intial-scale=1.0"/> || Etiqueta para indicar el manejo responsivo de nuestras paginas HTML. Sirve para acomodar de manera responsiva los estilos CSS y los framework como BootStrap.

Dependiendo del tamaño de la pantalla o del dispositivo en el que estemos trabajando, los elementos HTML se pueden acomodar de manera automática. Este concepto es conocido como Paginas Responsivas HTML. No solo se utiliza HTML, sino que también se utilizan hoja de estilo CSS o frameworks CSS como BootStrap-->.

8. Propio - head tag HTML

The HTML <head> element is a container for the following elements: <title>, <style>, <meta>, <link>, <script>, and <base>.

The HTML <head> Element

The <head> element is a container for metadata (data about data) and is placed between the <html> tag and the <body> tag.

HTML metadata is data about the HTML document. Metadata is not displayed.

Metadata typically define the document title, character set, styles, scripts, and other meta information.

# 2. Elementos Básicos HTML

1. Títulos en HTML

Los <hX> generan un "salto de línea" (En realidad es un elemento en bloque que ocupa toda nuestra pagina HTML). Según el tamaño que deseamos utilizar podemos elegir entre h1 y h6.

Los que no genera un "salto de línea" son considerados elementos en línea.

2. Párrafos en HTML

El texto o los elementos con los que vamos manejar no necesariamente se manejan siempre con etiquetas. Sin embargo, las recomendaciones y mejores prácticas es que siempre utilicemos una etiqueta que del significado al elemento o texto que estamos agregando.

Es muy importante el uso de etiquetas para después utilizar CSS.

3. Links en HTML

Utilizamos el tag a (anchor) para movernos entre distintas páginas web o distintos lugares dentro de nuestra misma página.

Con el atributo target="_blank" podemos hacer que se abran en una nueva pestaña.

4. Imágenes en HTML

Podemos agregar imágenes directamente desde una carpeta de nuestro proyecto o podemos linkearlas directamente desde internet.

Si la imagen no tiene una ruta correcta nos mostrar un icono con la "imagen rota".

Podemos asignarle un ancho, alto y un texto alternativo con atributos, entre otras cosas. El texto alternativo (alt) sirve por si se rompe la imagen, nos muestre el texto entre los alt.

Es recomendable solo ajustar el ancho o el alto, ya que la otra propiedad se ajusta sola, si utilizamos los dos se puede deformar la imagen.

5. Manejo de Atributos en HTML

El atributo <title> nos permite agregarle un título a la imagen. A diferencia del atributo <alt> se muestra si se carga de manera correcta la imagen y como el <alt> cuando no se muestra de manera correcta. El <title> se despliega cuando tenemos el curso por encima de nuestra imagen. Se aplican las mismas condiciones para otras etiquetas que no son <img>

Si bien podemos poner un atributo style es una buena práctica que todo lo que modifique el estilo de nuestra página se haga a través de CSS3 o un framework afín.

Los párrafos no respetan los enter ni los espacios, con el tag <pre> (dentro del atributo <p>) si se va a respetar tal cual lo estamos agregando. La etiqueta <pre> significa preformated.

6. Aplicar estilos CSS en HTML

Si bien podemos aplicar CSS a nuestros tags de HTML, es recomendable utilizar un archivo aparte CSS y vincularlo al tag. Vamos a ver como se aplica en las tags pero NO es una buena práctica hacerlo en el mismo archivo HTML, hacerlo como se menciona anteriormente.

Puedo agregar más de un atributo dentro del style dentro de mi tag si separo por ;.

7. Formato de texto en HTML

Si bien podemos dar formato a nuestros tags en el mismo HTML, es una buena práctica y se hace con regularidad con hojas de estilos CSS.

Con el tag  Podemos aplicar negrita a nutro texto.

La etiqueta  nos va a dar el mismo resultado visual que usar la etiqueta  pero en la teoría/semántica significa que es mucho más importante.

La etiqueta  nos da el texto en itálica.

La etiqueta  da el mismo resultado visual que la itálica pero se utiliza para enfatizar mucho más en ese tipo de texto.

La etiqueta <small></small> es un texto normal, pero más pequeño.

La etiqueta <mark></mark> es un resaltar el texto.

La etiqueta <del></del> es para indicar que nuestro texto se elimina. Dibuja una línea horizontal recta por encima del texto.

La etiqueta <ins></ins> es para indicar que nuestro texto se está insertando. Subraya el texto.

La etiqueta <sub></sub> es para indicar que nuestro texto es un subíndice. Sirve como para mostrar de manera correcta la fórmula de H2O.

La etiqueta <sup></sup> es para indicar que nuestro texto es un superíndice.

Las etiquetas mencionadas anteriormente no son solo para dar formato sino también para darle un sentido semántico al texto que estamos agregando.

8. Referencias de caracteres en HTML

La etiqueta <code></code> da formato al texto como si fuese un código. Muy útil si queremos mostrar lenguajes de programación.

Si nuestro teclado o nuestra codificación no permite el acceso a ciertos caracteres podemos ingresar a la página https://dev.w3.org/html5/html-author/charref y poner el carácter que queramos copiando el código y poniéndolo en el tag. Ejemplo: ® | Podemos utilizar el nombre del carácter, el valor hexadecimal o el valor decimal.

Esta práctica es muy útil ya que puede dar error de sintaxis utilizar !<>, con el código que provee la página anterior este problema se resuelve fácilmente.

Si tenemos problemas con las tildes podemos utilizar el siguiente formal &(letra)accute;

9. Link para enviar un Email en HTML

Utilizamos la etiqueta <a> pero dentro del atributo href especificamos que enviamos en mail de la siguiente manera: href="mailto:(MAIL)"

Podemos agregar un título y un cuerpo de inicio del mensaje de la siguiente forma: href="mailto:(MAIL)?subject=(TITULO)&body=(MENSAJE)" | El carácter & nos permite concatenar otro parámetro.

# 3. Introducción a HTML en CSS

## 1. Manejo de Colores en CSS y HTML y 2. Manejo de Colores en CSS y HTML parte 2

Pagina para ver los colores que soporta por defecto HTML y otros tipos de anotaciones (Name, Hexadecimal, RGB, hsl): https://htmlcolorcodes.com/es/nombres-de-los-colores/

Para aplicar color a un tag debemos agregar un atributo style="color:(COLOR);".

rgba para aplicar transparencia a un borde, ejemplo: Color por RGBA con transparencia .

También podemos aplicarlo de la misma forma con hsla (obviamente respetando que no utiliza un numero de 0 a 255 sino su propia regla)

## 3. Introducción a CSS (Cascading Style Sheet)

Como es muy impráctico poner código CSS en cada tag, vamos a utilizar las facilidades que nos da CSS para desarrollar el diseño para una o varios tags/paginas HTML.

Podemos relacionar nuestro CSS al HTML con la etiqueta link. Ej: <link rel="stylesheet" href="css/estilos.css">.

## 4. Cascadeo en las Hojas de Estilo CSS

Se llaman hojas de estilo en cascada porque si aplicamos a un nivel superior un estilo, entonces este estilo es aplica desde el elemento padre hasta el elemento hijo. Siendo nuestra hoja CSS nuestro padre y los tags de HTML los hijos.

Relación padre-hijo(s). Algo así como un virtual en C#. Browser Default Styles -> External Styles -> Embedded Styles -> Inline Styles

## 5. Box Model en CSS

El model de caja es como una mamushka, de afuera hacia dentro sería algo asi: Margin->Border->Padding->Content. A su vez tenemos los 4 ejes cartesianos para modificar. Esto se ve cuando se inspecciona un elemento con la herramienta Chrome en la sección computed.

6. Manejo de Padding, Border y Margin en CSS

Como se menciona en el punto 5. Podemos editar 1, 2, 3 o los 4 ejes cartesianos.

El orden en caso de querer hacerlo solo con la propiedad margin es: top-right-bottom-left

margin-top: 40px;

margin-right: 10px;

margin-bottom: 20px;

margin-left: 60px;

Es lo mismo que:

margin: 40px(top) 10px(right) 20px(left) 10px(bottom);

También si sabemos que el right y el left va a ser el mismo podemos tener la siguiente configuración del margin:

margin: 40px(top) 10px(right-left) 20px(bottom);

Podemos aplicar el mismo margin de izquierda a derecha eliminando los márgenes superiores e inferiores:

margin:auto (Pixeles)px;

7. Ejemplo de Paleta de Colores en CSS

https://coolors.co/palettes/trending

8. Estado y colores de los links en HTML y CSS

Podemos modificar el estado (azul si no lo clikeo, morado si lo clickie y rojo si lo mantengo presionado, etc) con CSS.

# 4. Links en HTML y CSS

1. Estado y colores de los links en HTML y CSS

Podemos modificar el estado (azul si no lo clikeo, morado si lo clickie y rojo si lo mantengo presionado, etc) con CSS.

2. Rutas relativas y absolutas en HTML

Ruta relativa = "contenido.html" -> Solo especificamos una parte y no la URL completa

Ruta absoluta = "http://google.com.ar" -> URL completa

# 5. Tablas en HTML

## 1. Tablas en HTML

Con la etiqueta table declaramos una tabla de HTML.

La etiqueta tr (table row) define un nuevo renglón de nuestra tabla.

Podemos (o no) agregar a nuestra primera etiqueta tr una cabecera con la etiqueta th (table head).

Si definimos cabecera debemos definir un nuevo tr y agregar el atributo td (table data).

El td es de izquierda a derecha. tr agrega un renglón de arriba hacia abajo, th es de izquierda a derecha

## 2. Tablas en HTML con CSS

Por default los table head se ponen en negrita y centrado en nuestra tabla.

Por default los table data se cargan hacia la izquierda

border-collapse: collapse; -> para colapsar las cajitas dentro de la tabla por cada td,th,tr

border-spacing: 5px; -> para espaciar las cajitas dentro de la tabla por cada td,th,tr

## 3. Atributos colspan y rowspan en Tablas HTML

Para que la tabla ocupe todo el lugar posible podemos asignar el siguiente estilo: width: 100%;

Si queremos ocupar elementos de tablas adyacentes de izquierda a derecha podemos utilizar el atributo colspan="" dentro de mi th o td para indicar cuantas celdas de izquierda a derecha va a ocupar.

Para expandir renglones tenemos la propiedad de rowspan="" dentro de mi td o th para para indicar cuantas celdas de de arriba hacia abajo va a ocupar.

4. Estilos CSS a Tablas HTML

Si queremos agregar un título a nuestra tabla podemos agregar el tag de caption. La etiqueta debe estar inmediatamente después de la etiqueta de table.

tr :nth-child(even) -> Nos permite seleccionar los renglones pares o impares (even pares, odd impares)

Los renglones no distigune de th y td.

# 6. Listas en HTML

1. Listas en HTML

Existen 3 tipos de listas: listas ordenadas<ol>; listas no ordenadas <ul>; listas de definicion<dl>;

Para agregar elementos a este tipo de listas utilizamos el tag <li> (list item).

Podemos cambiar los iconos por default del tipo de las listas con CSS. (style="list-style-type:"";).

También podemos anidar listas.

2. Listas de Descripcion

Utilizamos la etiqueta dl. Se relación a través de una key y un value.

La etiqueta dt (description term) para describir nuestro termino y la etiqueta dd (description data) para agregar la información de ese término.

# 7. Elementos inline y block en HTML y más temas

1. Elementos inline y block en HTML

Para los elementos en línea vamos a ver la etiqueta span y para los elementos en bloque la etiqueta div.

La diferencia entre ambos es que los elementos de bloque se inician en una nueva línea, en cambio los elementos en línea no inician una línea sino que se aplican dentro de la misma línea en donde estamos trabajando.

La división abarca todo el espacio-ancho posible.

2. Atributo class en HTML y CSS

Si por alguna razón solo queremos aplicar nuestros estilos a ciertos elementos podemos utilizar el concepto de class y aplicarlo solamente a los elementos CSS que nosotros queramos.

Necesitamos crear una clase para aplicar CSS a los elementos que necesitemos. Se definen dentro del archivo css con un ".(nombre)" y para vincularlo con el archivo HTML dentro de la etiqueta utilizamos el atributo class="claseCSS".

Con el * indicamos que se va aplicar a todo mi HTML el CSS que definamos dentro.

Como funciona en cascada si volvemos a definir un elemento luego de otro el primero se va a sobrescribir dentro del tag que tenga mi estilo CSS (algo así como variables locales vs variables globales en programación).

Se recomienda que nuestros títulos en CSS no deben contener ni espacios ni caracteres especiales, se recomienda utilizar minúsculas y si tengo varias palabras separarlas por guion.

Para agregar más de una clase a nuestro HTML simplemente espaciamos con la otra clase y ponemos el nombre.

## 3. Atributo id en HTML

Si queremos aplicar ciertos estilos solamente a un elemento de nuestra página HTML y este sea un elemento único dentro de nuestra página HTML utilizaremos el concepto de ID (similar al de class).

Utilizamos el carácter #(nombre) para indicar que estamos utilizando un elemento de tipo id.

Es importante que tengamos un solo elemento HMTL afectado por identificador (id). Si bien visualmente no notaremos ningún error, esto nos puede traer problemas cuando utilizamos JavaScript.

## 4. Marcadores o Bookmarks en HTML

El atributo de ID no solo sirve para asociar un identificador CSS y mejorar visualmente el elemento. También sirve para otras cuestiones, por ejemplo, para agregar el concepto de bookmarks o marcadores en HTML

Podemos generar un id="(Nombre)" y vincularlo con un anchor href para ir hacia ese elemento especifico si clickeamos en el link.

El identificador distingue de mayúsculas y minúsculas.

Se mueve nuestra página hacia el id marcado y a nuestra URL se le agrega el identificador que utilizamos para que ocurra este salto (ej: index.html#Java).

Es muy útil esta herramienta del URL ya que desde cualquier página podemos ir directamente a esa URL que nos proporciona el marcador

## 5. iFrames en HTML

Un iFrame en HTML es un marco HTML en el cual vamos a poder agregar información HTML. Así que vamos a poder agregar desde paginas HTML, otros sitios web directamente dentro de una página HTML, videos, documentos, etc.

Podemos utilizar el tag anchor para cargar un sitio web a nuestro iframe.

Ejemplo: <a href="index.html" target="marco1">Cargar index</a> -> Nuestro target tiene que coincidir con el atributo name que le otorguemos a nuestro iframe.

# 8. Formularios en HTML

1. Formularios en HTML

Un formulario nos va a permitir capturar información del usuario.

Para generar un formulario utilizamos la etiqueta de form. Para que se visualice en el navegador debemos agregar elementos que compongan nuestro formulario.

Los elementos de tipo input nos permiten capturar información de parte del usuario.

Con el atributo type vamos a especificar que tipo de elemento vamos a manejar.

Con el atributo id vamos a darle un nombre a ese elemento.

El atributo value sirve, por ejemplo, para darle un nombre a un botón cuando se visualice o darle un valor inicial a un elemento de tipo texto.

El atributo name se utiliza para especificar el nombre que se va a mostrar como parte del parámetro que se va a enviar hacia el servidor. Es recomendable que tenga el mismo nombre que el id. También utilizamos el name para vincular uno o más elementos de nuestro form.

El atributo placeholder nos va a permitir agregar un texto descriptivo dentro de nuestro elemento para realizar indicaciones.

La información se va a enviar a un servidor o la vamos a procesar con JavaScript.

2. Validar Formularios en HTML

Con el atributo de required dentro del tag podemos indicar que es un campo requerido. required="true".

Si queremos que un formulario no se valide, tenemos el atributo novalidate que podemos agregar dentro del tag de form. También podemos agregar el atributo novalidate="formnovalidate" dentro de un tag del form para que no haga falta validación (por default no hace falta validar).

3. Metodo GET y POST HTTP

Los parámetros después del signo '?' en la URL son los parámetros que enviamos al servidor. A esto se lo conoce como Query String.

Si tenemos información sensible (un campo de password por ejemplo) podemos cambiar el tipo de método para enviar la información al servidor (en la etiqueta form con el atributo method). Por default se utiliza el método GET, pero podemos utilizar el método POST para enviar la información sin que se agregue a la URL (Se envía en el request HTTP).

## 4. Tipos Email y Number en Formularios HTML

Los caracteres especiales sufren una codificación según el juego de caracteres que nosotros indicamos en el head de nuestra página HTML, para después ser enviados al servidor (HTML URL Encoding).

Si queremos que acepte números flotantes nuestro input type number, debemos agregar el atributo step="any".

## 5. RadioButtons y Checkboxes en Formularios HTML

Para poder seleccionar un solo radiobutton el valor de name debe ser el mismo, de lo contrario podemos seleccionar más de una opción.

El valor que se envía del radiobutton es el que especificamos en el atributo value.

La diferencia entre los radiobutton y los checkboxes es que los radiobutton se acostumbra que solo seleccionemos una opción, en cambio los checkboxes una o varias opciones.

## 6. Elemento Select y TextArea en HTML

Con el tag select podemos generar una lista desplegable de elementos. Para generar esta lista dentro del select utilizamos el tag de option.

No importa el texto de la etiqueta, importa el atributo value, que es lo que se va a enviar al servidor.

Con el atributo multiple="true" dentro del tag select indicamos que podemos seleccionar mas de una opción.

Con el atributo selected="true" dentro de una option indicamos que ese value va a ser el primero en ser seleccionado para mostrar apenas se carga el formulario.

El tag textarea nos va a permitir enviar una caja de texto a nuestro servidor. Sirve, por ejemplo, para enviar comentarios, observaciones, problemas, etc. Si escribimos un texto entro de los tags de apertura y cierre, este se va a agregar al textarea tal cual se lo ingreso (respeta espacios, tabulaciones, etc).

## 7. Fieldset y Legend en Formularios HTML

El Fieldset nos permite generar un marco al formulario. Es una buena práctica que el tag se coloque luego del tag form y que todo lo que vaya en el formulario se encuentre dentro del fieldset.

Con la etiqueta legend dentro del fieldset vamos a poder agregarle un título a nuestro formulario.

Ambos tags no afectan a la información que se envía al servidor, simplemente mejora visualmente el formulario.

## 8. Atributos del element form en HTML

La información se va a procesar con JS o con un servidor. Pero cuando llegue el momento precisamos indicar cual es la URL que va a procesar este formulario.

Con el atributo de action="(URL)" vamos a poder indicar donde se va a procesar esa información. (Si queremos ver el método utilizado, post o url, podemos enviar el formulario a una dirección invalida).

Con el atributo de target podemos indicar si la respuesta (Query) se envía en la misma pestaña o se abre una nueva pestaña (target="_self o _blank").

Con el autocomplete="off" anulamos la posibilidad de que el usuario utilice información que ya a enviado anteriormente (Por ejemplo cuando nos aparecen nombres que ya hemos ingresado con anterioridad).

9. Mas elementos de tipo input

El input reset nos va a permitir reiniciar todos los campos de nuestro formulario.

El atributo readonly="true" nos va a permitir que el usuario no modifique la información del campo. Esto se utiliza comúnmente si el servidor envía información a la página web y queremos que el usuario simplemente vea esa información que nos dio. De igual manera, este campo readonly se envía igual junto con los otros elementos de nuestro formulario.

Los inputs de tipo hidden nos permite enviar parámetros ocultos. Son ocultos para el usuario. Si tenemos el método get este elemento hidden se va a poder ver en la URL.

El atributo de disabled nos va a permitir que el usuario no pueda interactuar con algun elemento del formulario y no se va a enviar al servidor.

# 9. Elementos Semánticos en HTML

1. Elementos semánticos en HTML

Tiene el beneficio de que les es más fácil a los buscadores de páginas web clasificar cada una de las secciones de nuestra página web.

Los elementos semánticos no tienen una diferencia visual. Simplemente agregan un significado al elemento o etiqueta HTML.

2. Etiquetas estructurales en HTML

<section> -> Podemos agregar una sección en lugar de utilizar un elemento de tipo div.

<article> -> Son elementos independientes

<aside> -> Es para agregar un elemento en la barra lateral

<header> y <footer> -> Agrega un cabecero y un pie de pagina.

<hgroup> -> Podemos agrupar diferentes titulos (h1-h6)

<nav> -> Para definir una barra de navegacion.