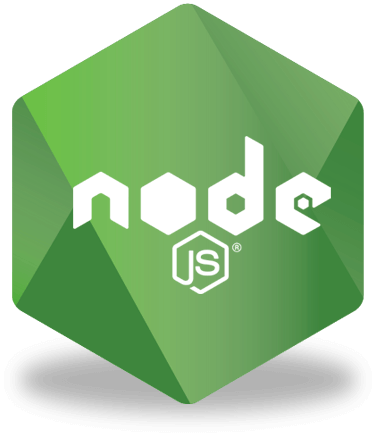
**Curso/Capacitacion de Javascript**

¿Qué se intenta enseñar en este curso?



1. Introducción a HTML5.
2. Se profundizará en el lenguaje de programación Javascript.
3. Se enseñará a utilizar Jquery, que es una librería fundamental para Javascript ya que está presente en la mayoria de los proyectos web.
4. Se explicará como hacer una maquetación web en HTML5 y CSS3.
5. También se explorará Typescript y su enfoque orientado a objetos.
6. Además, se trabajará con Angular, el framework más popular y demandado de los últimos tiempos.
7. Se realizará el backend con NodeJS.
8. Se utilizará el Módulo Express/REST.
9. Se combinará todo con MEAN Stack.



## **Introducción HTML ¿Qué veremos?**

1. Definiciones de HTML y que es una etiqueta.
2. Se explicará cómo está conformada la estructura de un documento HTML.
3. Tipos de etiquetas para textos en HTML (P, span, etc.)
4. Cómo realizar listas.
5. Mostrar imágenes en una página con HTML5.
6. Realizar tablas.
7. Realizar formularios.
8. Ejercicios con lo aprendido.
9. Javascript

**¿Qué es HTML?**

HyperText Markup Language, es un lenguaje de marcado o etiquetado que nos permite crear la estructura basica de una pagina web.

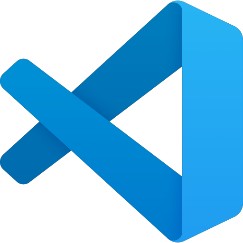
Luego para dar estilos o diseño a una web se utiliza un lenguaje de maquetación que es CSS, que significa Hojas de Estilos en Cascada.

Para este curso se recomienda usar alguno de estos navegadores Web:

Google Chrome Mozilla Firefox

Tambien se utilizará algún editor de texto, se recomienda:



Visual Studio Code

# 01-¿Qué es una etiqueta?

Una etiqueta es una marca que es utilizada para dividir distintas secciones de una página web, ya sea por encabezado, por cuerpo o por píe, entre otras.

Por ejemplo, si creamos un archivo con extencion .html qué es la extención utilizada para crear las paginas web y escribimos dentro de el:

<head>

</head>

<body>

<h1>Este es el titulo</h1>

<h2>Este es el cuerpo de la página</h2>

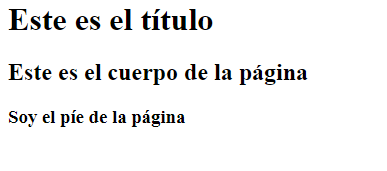
</body>

<footer>

<h3>Soy el píe de la página</h3>

</footer>

Al abrirlo se abrirá una nueva ventana del navegador web y se vería así:



# 02-Estructura HTML

Para hacer una página web deberia tener una cabecera y un cuerpo, primero se define que version de HTML estamos utilizando, esto se hace de la siguiente manera:

<!DOCTYPE HTML>

Una página web HTML deberia tener una estructura parecida a esta:

Primero se abre una etiqueta “html”:

<html>

Y se cierra la etiqueta

</html>

Luego se abre la etiqueta “head

<head>

</head>

Dentro de ella se insertan los script, y los estilos css (si fuesen pocos), en debido caso de que esos estilos fuesen muchos se hacen en un archivo .css a parte.

Luego va el Body que es el cuerpo de la página, donde basicamente entra todo lo que vemos en una pagina web, título, parrafos, detalles y demás.

<body>

</body>

Y si se requiere al final un footer, que es el píe de la página:

<footer>

</footer>

# 03- Etiquetas de texto

Existen distintas etiquetas de texto que se utilizan dependiendo de lo que se necesite, por ejemplo, para un titulo lo mas común sería usar la etiqueta <h1>Esto es un título</h1>

Se pasa a listar los dintintos tipos de etiqueta:

<h1> </h1> Se utiliza mayoritariamente para el encabezado o titulo de una página.

<h2> </h2> Es usado para subtitulos o textos grandes.

<h3> </h3> Es el texto que se utiliza en los parrafos.

<h4> </h4>

<h5> </h5>

<h6> </h6> Son aquellas etiquetas utilizadas para textos mas pequeños respectivamente (cuanto mas grande es el numero que acompaña a la “h” mas pequeña es la fuente.

<p> </p> Es para indicar parrafos.

<hr/> Esta etiqueta es utilizada para hacer un salto de linea junto a una linea que divide la el texto.

<br/> Es un salto de linea sin linea.

# 04 – Listas

En HTML tambien podemos hacer listas como esta:

* Elemento 1
* Elemento 2
* Elemento 3

La estructura de una lista así en HTML es asi:

<ul>

<li>Elemento 1</li>

<li>Elemento 2</li>

<li> Elemento 3</li>

</ul>

# 05 – Imágenes

Para añadir imágenes a una página HTML se utiliza la etiqueta “img”

La estructura es la siguiente:

<img src=”” alt=”” title=””/>

src es la ubicación de la imagen, ya sea en el directorio local o una url.

alt es por si falla la carga de la imagen dejar un texto que indique que habia ahí o que muestre un mensaje de error.

title es el titutlo de la foto si es que se le quiere indicar, al igual que alt, es opcional.

Ejemplo:

<img src=”imagenes/perro.jpg” alt=”Acá había una foto de un perro” title=”Foto de perro”/>

# 06 – Tablas

Las tablas nos sirven para dibujar en la pantalla una tabla tipica de contenido que contienen filas y columnas con valores.

Para hacer tablas en html se utiliza la etiqueta “table”

Para hacer una fila se utiliza la etiqueta tr y para hacer una columna la etiqueta td

Entonces un ejemplo de tabla sería:

<table>

<tr/>

<td>Documento</td>

<td>Nombre</td>

<td>Apellido</td>

<tr>

</table>

Entonces tenemos una tabla con tres columnas, pero sin datos.

Para añadirle datos a esa tabla simplemente se le agregarian otras tres columnas, pero con los datos de esas tres columnas de arriba.

<table>

<tr/>

<td>Documento</td>

<td>Nombre</td>

<td>Apellido</td>

<tr>

<tr/>

<td>123456</td>

<td>Lautaro</td>

<td>Biondi</td>

<tr>

<tr/>

<td>123455</td>

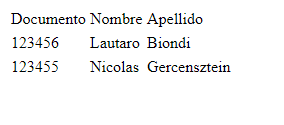
<td>Nicolas</td>

<td>Gercensztein</td>

<tr>

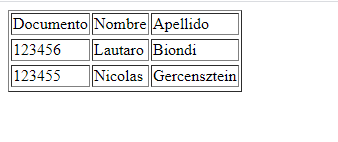
</table>

Si esto lo copiamos dentro de un archivo .html y lo abrimos nos quedaria una tabla así:



Por lo visto no tiene un formato de tabla si no de texto, para que tenga un formato mas de tabla se le añade la propiedad border a la etiqueta table asi: <table border=1px>

Entonces se terminaría viendo así:



# 07 – Formularios

Los formularios se utilizan para que el usuario cargue datos e interactue con la página.

La etiqueta de los formularios es “form”

<form> </form>

En ella hay distintos atributos:

action=”” se utiliza para indicar a donde se enviaran los datos del formulario.

method=”” el atributo method es para indicar que metodo se va a utlizar (POST para enviar, GET para obtener).

Entre las etiquetas form van las etiquetas de input para insertar datos con su respectivo tipo, por ejemplo:

<input type=”text”/>

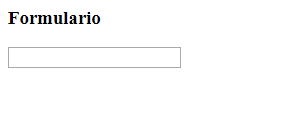
Si escribimos esto en el archivo HTML podremos ver un texto “Formulario” y un input de tipo texto:

<h3>Formulario</h3>

<form>

<input type=”text”/>

</form>



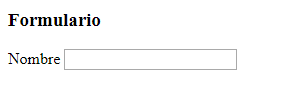
Dentro del input se puede escribir lo que se quisiera.

Para indicar que se debería escribir dentro del input se debe agregar una etiqueta “label” que es el nombre que va a tener ese campo en la página, por ejemplo:

<label>Nombre</label>

<input type=”text”/>

Entonces quedaría algo así:



**Ejercicio Completo con HTML**

Lo que se busca hacer en este ejercicio son distintas paginas web que estén enlazadas por un menú, esas páginas son:

1. Una pagina de inicio donde se muestre un titulo, un menu con las demas paginas para hacerle click al texto de cada página y dirigirse a ellas. Tambien que tenga una imagen y varios parrafos debajo ya sea escritos a mano o con un generador de texto como lorem ipsum por ejemplo.
2. Una página de contenidos que contenga una tabla con los contenidos que contiene esta web.
3. Y una página de contacto en la que el usuario pueda introducir un nombre, una descripcion e indicar si es mayor de edad o menor de edad y un boton que diga “enviar email”.

¡NOTA IMPORTANTE! Una buena practica es ir dejando comentarios en el codigo para indicar que hay debajo.

Paso 1: Crear una carpeta llamada web y dentro de ella crear un archivo que se llame index.html, este archivo será la página principal de la web.

Paso 2: Abrir ese archivo con un editor de texto para empezar a armar la estructura basica HTML en el:

<!DOCTYPE HTML/> <!—Acá se indica que tipo de documento es, en este caso HTML-->

<html>

<head>

<!--Acá van aquellos datos que no se ven a simple vista en la página web -->

<title>Mi primera web</title>

</head>

<body>

</body>

</html>

Paso 3: Añadir un título dentro del body con la etiqueta h1 que dentro diga “Mi primera web”

Paso 4: Debajo del titulo hacer un formulario dentro del body que tenga 3 elementos li con los nombres Inicio, Contenidos y Contacto, ese será el menú.

Paso 5: Agregar una etiqueta hr debajo del formulario para añadir una linea de separación en la página.

Paso 6: Crear una carpeta en la ubicación de index.html que se llame imágenes y en ella agregar la imagen que se quiera, en el ejercicio de este documento se utilizarán imágenes de animales.

Paso 7: agregar una etiqueta img dentro del body del documento, la cual tenga el atributo src=”imágenes/” y la imagen que se haya puesto en esa carpeta, luego agregar el atributo alt=”Acá había una imagen de un perro”, y el atributo title=”Un perro”.

Paso 8 : Insertar debajo varios parrafos con la etiqueta p con un texto ya sea escrito o generado.

El archivo quedaria algo así:

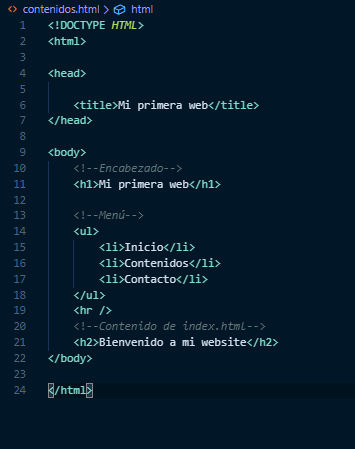


Paso 9: Crear una copia del archivo index.html y renombrarlo contenidos.html en la carpeta web.

Paso 10: Abrir ese archivo con el editor de texto.

Paso 11: Borrar todo el contenido desde el <img> hasta el ultimo </p>

Quedaría así:



Paso 12: Reemplazar el contenido de la etiqueta h2 por “Tabla de contenidos”

Paso 13: Añadir entre el </h2> y el </body> una etiqueta de tabla (<table>) y su cierre </table>

Paso 14: Entre la etiqueta <table> y su cierre </table> añadir una etiqueta <tr> junto a su cierre </tr> y dentro de ella añadir cuatro etiquetas <td> con sus respectivos cierres, quedaría algo así:

<table>

<tr>

<td></td>

<td></td>

<td></td>

<td></td>

</tr>

</table>

Paso 15: Entre cada etiqueta <td> y su cierre insertar el nombre de las cosas que tengan las cuatro imágenes, por ejemplo:

  <tr>

            <th>Perro</th>

            <th>Oso</th>

            <th>Caballo</th>

            <th>Gato</th>

      </tr>

Paso 16: Volver a hacer una etiqueta <tr> y su cierre debajo de la etiqueta de cierre </tr> y añadir dentro de ellas añadir cuatro etiquetas <td> junto a sus cierres.

Paso 17: Entre cada etiqueta <td> y su cierre </td> escribir una etiqueta <img> con atributo src=”imágenes/” y una imagen, tambien agregar el atributo alt=”” con la explicación de que habría ahí si llegase a fallar la carga de la imagén, además agregar el atributo height=”100px” y luego hacer lo mismo con las demas etiquetas <td>, quedaría así:



Entonces, abrimos contenidos.html en el navegador se vería algo así:



Y ahora hay que crear la página de contacto, para eso, crear una copia del archivo contenidos.html y renombrarlo contacto.html.

Paso 18: Abrir el archivo contacto.html con el editor de texto.

Paso 19: Borrar toda la tabla para que solo quede el título y el menú.

Paso 20: Agregar una etiqueta de <form> con su cierre </form>

Paso 21: Entre la etiqueta <form> y su cierre agregar una etiqueta <label> con su cierre </label>

Paso 22: Dentro de la etiqueta <label> agregar atributo for=”nombre” y entre la etiqueta <label> y su cierre</label> escribir “Nombre” sin comillas.

Paso 23: Añadir un salto de linea al lado del cierre de label </label> con la etiqueta <br/>

Paso 24: Debajo agregar una etiqueta <input> dentro de la etiqueta input insertar el atributo type=”text” y el atributo name=”name”

Paso 25: Encerrar las etiquetas <label> y <input> con una la etiqueta <p> y su cierre </p>.

Quedaría así:

    <form>

        <p>

            <label *for*="name">Nombre</label> <br/>

            <input *type*="text" *name*="nombre">

        </p>

    </form>

Paso 26: Hacer una copia del parrafo debajo de ese y cambiar el valor del atributo de label for=”name” a for=”biografia” y Nombre por Biografia

Paso 27: borrar el input y crear una etiqueta <textarea> y su cierre </textarea> y añadirle el atributo name=”biografia”, deberia ir quedando así:

    <h2>Bienvenido a mi website</h2>

    <form>

        <p>

            <label *for*="name">Nombre</label> <br/>

            <input *type*="text" *name*="nombre">

        </p>

        <p>

            <label *for*="biografia">Biografia</label> <br/>

            <textarea *name*="biografia"></textarea>

        </p>

    </form>

Paso 28: Volver a crear un parrafo debajo del ultimo parrafo y dentro agregar una etiqueta <label> y su cierre </label>

Paso 29: Añadir el atributo for=”edad” y escribir entre la etiqueta <label> y su cierre la palabra Edad.

Paso 30: Debalo del label insertar una etiqueta <select> y su cierre </select> y dentro agregar tres etiquetas <option> y sus cierres </option> y en la primera option escribir “Mayor de edad” y añadirle el atributo value=”mayor”, en la segunda option escribir “Adulto” y añadirle el atributo value=”adulto”, y en la tercera option escribir “Mayor de 65 años” y añadirle el atributo value=”65”, deberia quedar así:

    <h2>Bienvenido a mi website</h2>

    <form>

        <p>

            <label *for*="name">Nombre</label> <br />

            <input *type*="text" *name*="nombre">

        </p>

        <p>

            <label *for*="biografia">Biografia</label> <br />

            <textarea *name*="biografia"></textarea>

        </p>

        <p>

            <label *for*="edad">Edad</label>

            <select *name*="edad">

                <option *value*="mayor">Mayor de edad</option>

                <option *value*="adulto">Adulto</option>

                <option *value*="65">Mayor de 65 años</option>

            </select>

        </p>

    </form>

Y la página se debería ver así:



Paso 31: Abrir con el editor de texto index.html.

Paso 32: Agregar dentro del <li>Inicio</li> la etiqueta <a> de manera que la linea de codigo quede de la siguiente manera: <li><a>Inicio</a></li> y dentro de la etiqueta <a> añadirle el atributo href=”index.html”.

Paso 33: Agregar dentro del <li>Contenidos</li> la etiqueta <a> de manera que la linea de codigo quede de la siguiente manera: <li><a>Contenidos</a></li> y dentro de la etiqueta <a> añadirle el atributo href=”contenidos.html”.

Paso 34: Agregar dentro del <li>Contacto</li> la etiqueta <a> de manera que la linea de codigo quede de la siguiente manera: <li><a>Contacto</a></li> y dentro de la etiqueta <a> añadirle el atributo href=”contacto.html”.

Entonces quedaría:

    <ul>

        <li><a *href*="index.html">Inicio</a></li>

        <li *href*="contenidos.html">Contenidos</li>

        <li *href*="contacto.html">Contacto</li>

    </ul>

    <hr/>

Paso 35: Copiar la lista del menú y reemplazarla en los archivos contacto.html y contenidos.html

De esta manera ya quedaría por concluido el ejercicio.

**Introducción a Javascript**

# **¿Qué es Javascript?**

Javascript, también conocido como JS, es un lenguaje de programacion interpretado, es decir, que no precisa de compilación como otros lenguajes de programación como Java, ya que es interpretado por otros programas, como los navegadores web.

Este lenguaje es utilizado principalmente para ejecutar acciones del lado del cliente, aunque tambien existen formas de usar Javascript del lado del servidor.

Su principal objetivo es agregar interactividad a las paginas web, es decir, dotarlas de cierta inteligencia artificial, y que los eventos de ciertas acciones del usuario se representen en la página.

Javascript permite a los desarrolladores añadir efectos especiales dinamicos a las paginas para hacerlas mas vistosa.

Por otra parte, Javascript está definido como POO(Programación Orientada a Objetos), tambien se caracteriza por tener tipado debil, es decir que no hace falta especificar que tipo de dato se almacena como pasa con otros lenguajes.

Para empezar con la introducción a Javascript lo primero será crear una carpeta llamada ‘javascript’.

¡NOTA!

Lo que se recomienda hacer es crear una carpeta que se llame CapacitaciónJS y dentro de ella crear la sub-carpeta deseada (en este caso javascript).

# **Hola Mundo en JS**

Paso 1: Dentro de la carpeta javascript se debe crear otra llamada “hola-mundo”.

Paso 2: Dentro de esa carpeta crear un archivo index.html

Paso 3: Abrir index.html con un editor de texto.

Paso 4: Escribir la estructura básica de HTML:

<!DOCTYPE *html*>

<html *lang*="es">

    <head>

        <meta *charset*="utf-8">

        <title>Hola Mundo con Javascript</title>

    </head>

    <body>

    </body>

</html>

Paso 5: En el body agregar una etiqueta <h1> con su cierre y dentro el titulo de la página, en este caso se usará “Capacitacion JS”.

Paso 6: Debajo de la etiqueta de cierre </h1> abrir una etiqueta <p> y escribir “Mi primer Script en JS” y cerrar la etiqueta </p>

Paso 7: En el head debajo del <title> abrir una etiqueta <script> junto a su cierre </script> añadiendole el atributo type=”text/javascript”

Paso 8: Entre las etiquetas script escribir: Alert(‘Hola Mundo con JS’).

De esta manera al abrir el archivo index.html se mostrará una alerta de navegador que diga el mensaje escrito en el parentesis.

Paso 9: Debajo de la etiqueta de cierre </script> abrir otra etiqueta de <script> y su cierre </script> añadiendole el atributo type=”text/javascript” y el atributo src=”js/hola-mundo.js”.

Paso 10: Crear una carpeta llamada js y dentro crear un archivo llamado “01-hola-mundo.js” y abrirlo con un editor de texto.

Paso 11: Copiar la linea de codigo que contiene el alert de index.html y pegarla dentro de hola-mundo.js y debajo escribir console.log(‘Hola Mundo en JS’) de tal manera que el archivo hola-mundo.js contenga:

*alert*('Hola Mundo en JS')

console*.*log('Hola Mundo en JS')

Paso 12: Comentar la primera etiqueta <script>,su contenido y su cierre</script> utilizando <!— al principio y para cerrar el comentario se utiliza -->

Entonces quedaria así:

        <!--

*<script type="text/javascript">*

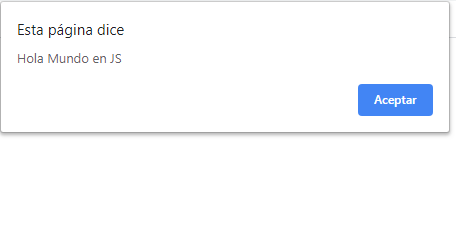
*alert('Hola Mundo en JS')*

*</script>*

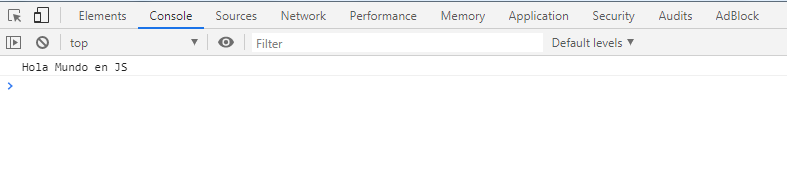
-->

        <script *type*="text/javascript" *src*="js/01-hola-mundo.js"></script>

Si se abre el archivo index.html se podrá comprobar que el script se ejecuta de todas formas pero desde el archivo js creado anteriormente.



Además, en la consola del navegador (La cual se puede ver presionando la tecla F12 (desde Google Chrome) se podrá ver que también en la consola dice “Hola Mundo en JS”



# **Comentarios y variables**

Comentarios:

Existen dos maneras para realizar comentarios en Javascript:

// es para comentar una sola línea. Por ejemplo:

console*.*log('Hola Mundo en JS')//*Esto es un console.log()*

/\* es para comenzar el comentario y \*/ es para terminarlo, por ejemplo:

/\*

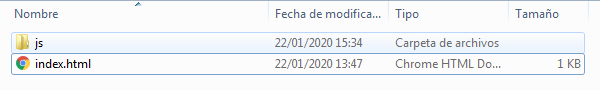
*Esto es un*

*comentario de*

*varias lineas*

\*/

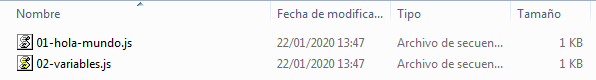
Para practicar con variables primero se debe crear una carpeta llamada js dentro de la carpeta llamada hola-mundo que se creó anteriormente; el directorio quedaría así:



Luego se debe crear un archivo llamado 02-variables.js dentro de la carpeta js.

Los archivos .js son propios de Javascript y se pueden ejecutar desde prácticamente todos los navegadores web.

Y dentro de la carpeta js quedaría así:



Ahora hay que abrir el archivo 02-variables.js con el editor de texto.

Luego escribir dentro del archivo el comentario //Variables para indicar con lo que se va a trabajar.

Variables:

Una variable es un contenedor de información, las cuales se pueden definir con las palabras var, let y const, según el tipo deseado.

Por el momento se usará var, luego se dará la diferencia entre ambas.

Ejemplo:

var pais = ‘Argentina’

index.html:

<!DOCTYPE *html*>

<html *lang*="es">

    <head>

        <meta *charset*="utf-8">

        <title>Hola Mundo con Javascript</title>

        <script *type*="text/javascript" *src*="js/02-variables.js"></script>

    </head>

    <body>

    </body>

</html>

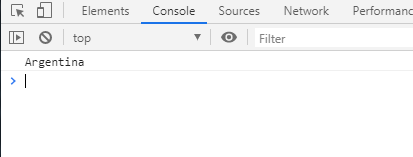
02-variables.js:

//*Variables*

varpais='Argentina';

console*.*log(pais);

Guardar y abrir el index.html:



Y se pueden definir más variables debajo, como nombre, provincia, etc

var pais=’Argentina’

var nombre=’Lautaro’

var provincia=’Buenos Aires’

Además, se puede hacer un console.log de las 3 a la vez separándolas por comas.

//*Variables*

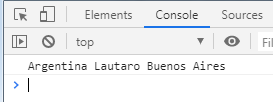
varpais='Argentina';

varnombre='Lautaro';

varprovincia='Buenos Aires'

console*.*log(pais,nombre,provincia);

y en la consola se veria asi:



También se pueden concatenar dos variables con el signo +

Ej:

var pais\_y\_provincia=pais+’ ’+provincia

//*Variables*

varpais='Argentina';

varnombre='Lautaro';

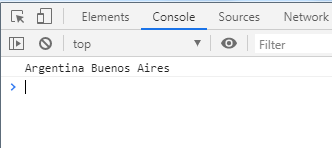
varprovincia='Buenos Aires';

varpais\_y\_provincia=pais+' '+provincia;

console*.*log(pais,nombre,provincia);

console*.*log(pais\_y\_provincia)

y en la consola se vería así:



**var, let y const**

En Javascript con los nuevos estándares se puede definir qué tipo de variable se va a usar:

- var es para variables globales (es decir que su valor es global)

- let es para variables que se encapsulan en bloques, ejemplo:

varmarca='nike'

function *zapatilla*(){

    letmarca='adidas'

    console*.*log(marca)

}

console*.*log(marca);

entonces lo que se busca demostrar es que dentro del bloque de la función se cambia el valor de la variable marca y cuando se hace el log dice adidas, pero cuando se le vuelve a hacer log fuera del bloque la marca vuelve a ser nike. Esto es porque las variables let solo aplican dentro de sus bloques, esto no quiere decir que no se pueda definir una variable let fuera de un bloque y funcione como si fuese una variable var.

A diferencia de var y let, la variable const tiene un valor constante, quiere decir que una vez que se le define un valor ya no se puede cambiar durante la función en ejecución.

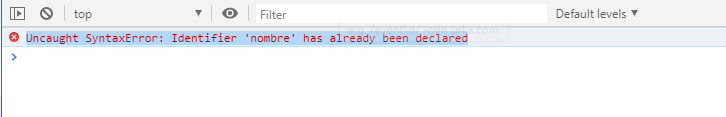
Ejemplo:

const*nombre*='Lautaro'

nombre = 'Juan'

console*.*log(nombre);

y al ver la consola del navegador aparece el siguiente error:



La variable nombre ya fue declarada y al intentar cambiarla falla.

**Tipos de datos en Javascript**

Los tipos de datos básicos que se usan en javascript son:

* String
* Number
* Boolean
* Array

Para definir el tipo de dato que va a recibir una variable se hace de la siguiente manera:

varnombre=*String*;

**Condicionales**

# **if**

La sintaxis del condicional if es la siguiente:

//*Ejemplo if*

varnombre='Lautaro'

if(nombre == 'Lautaro'){

    console*.*log(nombre)

}

# **Else**

Else va despues del condicional if y su sintaxis es la siguiente

//*Ejemplo else*

varnombre='Juan'

if(nombre == 'Lautaro'){

    console*.*log(nombre)

}else{

    console*.*log('El nombre no es Lautaro')

}

# **Operadores Logicos**

Los operadores logicos que se utilizan en Javascript son:

* && (and)
* || (or)
* ! (not)

Ejemplo de cada uno de ellos:

//*Operadores logicos*

//*Ejemplo de &&*

varpais='Argentina';

varprovincia='Buenos Aires';

if(pais=='Argentina' && provincia == 'Buenos Aires'){

    console*.*log('El pais es Argentina y la provincia es Buenos Aires')

}

//*Ejemplo de ||*

varpais='Argentina';

varprovincia='Buenos Aires';

if(pais=='Argentina' || provincia == 'Buenos Aires'){

    console*.*log('El pais es Argentina o la provincia es Buenos Aires')

}

//*Ejemplo de !=*

varpais='Argentina';

if(pais!='Argentina'){

    console*.*log('El pais no es Argentina')

}

**Bucles**

Los tipos de búcles que se usan en Javascript son:

* FOR
* WHILE
* DO WHILE

Ejemplo de cada uno de ellos:

//*Ejemplo de for*

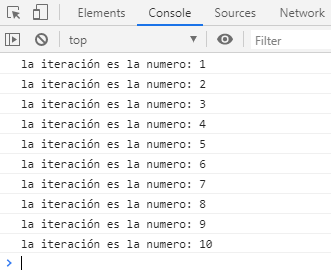
varnumero=10;

*for*(vari=1;numero >=i ;i++){

    console*.*log('la iteración es la numero: '+i)

};

Resultado en la consola:



Ejemplo de while:

//*Ejemplo de while*

vari=1;

varnum=10;

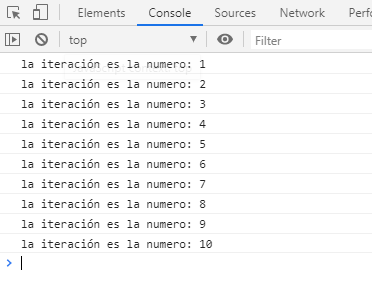
*while*(num >= i){

    console*.*log('la iteración es la numero: '+i);

    i++;

}

Resultado en consola:



Ejemplo de Do while:

//*Ejemplo de do while*

varnumero1=1;

varnumero2=10;

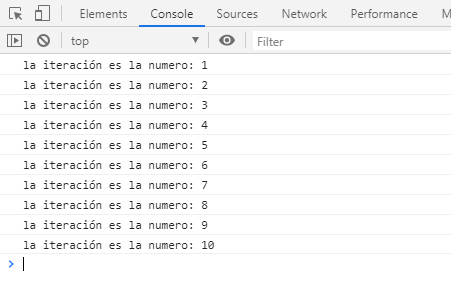
*do*{

    console*.*log('la iteración es la numero: '+numero1);

    numero1++;

} *while*(numero1 < 11)

Resultado en consola:



# **Alertas e ingresos de datos**

La función alert() sirve para mostrarle al usuario en el navegador un mensaje como pop-up. Ejemplo:

Alert(‘Bienvenido a mi página web’);

La función prompt() tiene un formato parecido a alert() pero con la diferencia de que tiene un cuadro para insertar datos y recibe de segundo parametro un valor por defecto.

Ejemplo:

prompt(‘Qué edad tenés?’, 18);

# 

# **Ejercicios de programación**

01 – Se pide hacer un programa que pida dos números al usuario y diga cual es el mayor, el menor y si son iguales.

02 – Hacer un programa que muestre todos los números que existen entre dos ingresados por el usuario.

# **Funciones**

Una función es un procedimiento en Javascript, un conjunto de sentencias que realizan una tarea o calculan un valor.

La sintaxis es:

***function*** *nombreDeLaFuncion*(parametrosOpcionales){  
 Código a ejecutar

}

Ejemplo:

varresultado=*Number*;

function *obtenerValor*(numero1,numero2){

    resultado=numero1+numero2;

    console*.*log(resultado);

}

Consola=



# **Funciones flecha**

Otra forma de declarar funciones es usando una flecha =>, ejemplo:

//*Declarar funcion*

function *imprimirMensaje*() {

    console*.*log('Hola Mundo');

};

//*Declarar funcion flecha*

var*imprimirMensaje*=()=>{

*console.log*('Hola Mundo');

}

# **Arrays**

La sintaxis de los Arrays o arreglos es:

var nombreArreglo= [contenido];

**Javascript en profundidad**



# **DOM (Document Object Model)**

La finalidad de Javascript es interactuar con la página web, es decir, interactuar con el código HTML y que mediante Javascript se pueda manipular de manera dinámica el Document Object Model, que es el arbol de etiquetas y elementos que componen una web, por ejemplo el body, el h1 y demas.

Para practicar con el DOM, se deberá seguir los siguientes pasos:

Paso 1: Crear una carpeta en la carpeta “javascript” y nombrarla “practicando-js”.

Paso 2: Crear una carpeta dentro de la carpeta “practicando-js” y nombrarla “js”.

Paso 3: Crear un archivo llamado index.html dentro de la carpeta llamada “practicando-js”.

Paso 4: Crear un archivo llamado 01-dom.js dentro de la carpeta “js”.

Paso 5: Abrir con el editor de texto el archivo index.html de la carpeta “practicando-js”.

Paso 6: Escribir una estructura html básica e importar el script 01-dom.js

<!DOCTYPE *HTML*>

<html *lang*='es'>

    <meta *charset*='utf-8'>

<head>

    <title>Practicando JS</title>

    <script *type*="text/javascript" *src*="js/01-dom.js"></script>

</head>

<body>

</body>

</html>

Paso 7: Abrir con el editor de texto el archivo 01-dom.js

Paso 8: Dentro escribir un comentario que diga DOM – Document Object Model

Paso 9: En el index.html, dentro del body, crear una etiqueta <div> con el atributo id=”caja” con su cierre </div> y que el contenido del div sea un texto que diga “¡Hola, soy una caja!”

<!DOCTYPE *HTML*>

<html *lang*='es'>

    <meta *charset*='utf-8'>

<head>

    <title>Practicando JS</title>

    <script *type*="text/javascript" *src*="js/01-dom.js"></script>

</head>

<body>

    <div *id*="caja">¡Hola, soy una caja!</div>

</body>

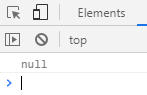
</html>

Paso 10: En el script 01-dom.js debajo del comentario escribir var micaja = document.getElementById(“caja”);

console.log(micaja)

Paso 11: Guardar los cambios y abrir el archivo index.html

Al ver la consola del navegador luego de abrir index.html se puede apreciar que el valor de micaja que muestra la consola es null.



Esto se debe a que el script se ejecuta antes de que se cargue el contenido del div, entonces para solucionar ese inconveniente se carga el script debajo del body, de esta manera primero se carga el div y luego se ejecuta el script. Quedaría así:

<!DOCTYPE *HTML*>

<html *lang*='es'>

    <meta *charset*='utf-8'>

<head>

    <title>Practicando JS</title>

</head>

<body>

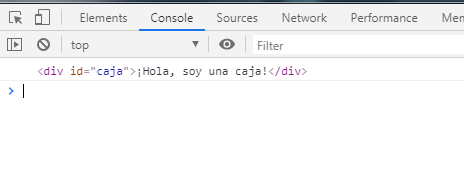
    <div *id*="caja">¡Hola, soy una caja!</div>

</body>

<script *type*="text/javascript" *src*="js/01-dom.js"></script>

</html>

De esta manera al recargar la página y ver la consola se verá el elemento con id=”caja”, en este caso el div:



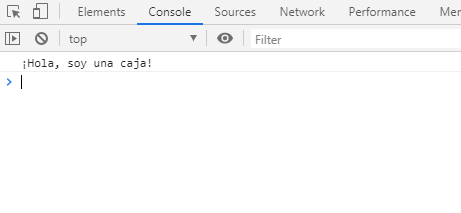
Para poder obtener solo el contenido del div con el id “caja”, se puede hacer de la siguiente manera utilizando la propiedad innerHTML.

//*DOM - Document Object Model*

varmicaja=*document.getElementById*('caja')*.innerHTML*;

console*.*log(micaja);

Entonces en la consola del navegador se mostraría así:



Para poder modificar el contenido del div con el id “caja”, se puede hacer de la siguiente manera utilizando la propiedad innerHTML:

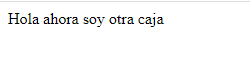
//*DOM - Document Object Model*

varmicaja=*document.getElementById*('caja');

console*.*log(micaja);

*micaja.*innerHTML="Hola ahora soy otra caja"

entonces en la página se verá así:



Incluso se puede cambiar los estilos de un objeto del documento de la siguiente manera:

//*DOM - Document Object Model*

varmicaja=*document.getElementById*('caja');

console*.*log(micaja);

*micaja.*innerHTML="Hola ahora soy otra caja"

*micaja.style.*background='red';

Entonces al refrescar la página se verá el fondo del contenido del div de color rojo.

Utilizando la propiedad style se puede cambiar el estilo de un objeto eligiendo la propiedad a cambiar.

*micaja.style.*background='red';

*micaja.style.*color='white';

*micaja.style.*padding='20px';

De esta manera la página se verá así:



Hay muchas más propiedades css que se pueden cambiar a cualquier elemento, esto lo podemos consultar en el siguiente link:

<https://www.quackit.com/css/properties/>

# **BOM (Browser Object Model)**

BOM permite acceder y modificar las propiedades de las ventanas del propio navegador.

Mediante BOM, es posible redimensionar y mover la ventana del navegador, modificar el texto que se muestra en la barra de estado y realizar muchas otras manipulaciones no relacionadas con el contenido de la página HTML.

Para practicar con BOM se debe crear un archivo 02-bom.js dentro de la carpeta js creada anteriormente y seguir los siguientes pasos:

En el index.html importar el script 02-bom.js:

<!DOCTYPE *HTML*>

<html *lang*='es'>

    <meta *charset*='utf-8'>

<head>

    <title>Practicando JS</title>

</head>

<body>

    <div *id*="caja">¡Hola, soy una caja!</div>

</body>

<!--

*<script type="text/javascript" src="js/01-dom.js"></script>*

-->

<script *type*="text/javascript" *src*="js/02-bom.js"></script>

</html>

Paso 2: Abrir con el editor de texto el archivo bom.js creado anteriormente

Paso 3: Comentar //BOM - Browser Object Model

Paso 4: Escribir console.log(window.innerHeight)

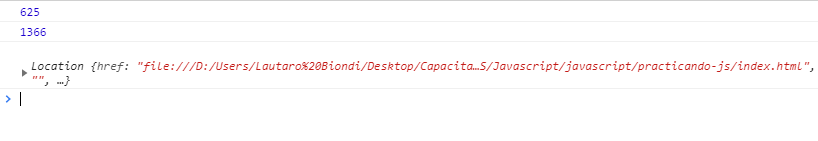
Paso 5: Guardar y abrir con el navegador el archivo index.html

La consola mostrará la altura interior de la ventana.

Paso 6: Escribir console.log(window.innerWidth)

La consola además de mostrar la altura interior de la ventana tambien mostrará la anchura interior de la ventana.

Paso 7: También se puede obtener la url de la página utilizando el comando window.location, entonces, escribir console.log(window.location)



# **Eventos**

Los eventos se ejecutan cuando sucede algo, por ejemplo, si el usuario le hace click a un boton se ejecuta el evento de click, y con el, una función.

Para practicar con Eventos seguir estos pasos:

Paso 1: Crear un archivo 03-Eventos.js en la carpeta js

Paso 2: Importar el archivo 03-Eventos.js en index.html

Paso 3: