

## ENLACE HTML-JAVASCRIPT

<code>&lt;script src="java.js"&gt;&lt;/script&gt;</code>	Esto lo coloco antes de cerrar el body al final y en src coloco el nombre del archivo js puede ser cualquiera con la extensión js
--	---

## CUADROS DE DIALOGO

Un modal es una pequeña ventana con la que el usuario interactúa y no desaparece hasta que presiona "OK" o "Aceptar".

<code>alert ("Esto es una alerta")</code>		<p>El alert Sirve para dar información a modo de alerta y se quita únicamente dándole aceptar.</p>
<code>let confirmar = confirm ("quiere seguir navegando");</code> <code>console.log(confirmar);</code>		<p>El confirm Sirve para dar información, pero me da la opción de dar click en aceptar o cancelar y guarda los datos booleanos true o false.</p>
<code>let nombre = prompt ("Por favor, introduzca su nombre");</code> <code>console.log(nombre);</code>		<p>El prompt Sirve para preguntarle algo al usuario y que pueda ingresar lo que desee en un texto y lo podamos almacenar.</p>
<code>let confirmar = confirm ("desea asistir a la fiesta");</code>  <code>if (confirmar) {</code> <code>    let nombre = prompt ("por favor ingrese su nombre");</code> <code>    alert (nombre + "estas inscrito en la fiesta te esperamos!")</code>  <code>}else{</code> <code>    alert ("Esperamos contar con vos en una próxima oportunidad")</code> <code>}</code>		<p>Utilizamos los 3 inicialmente con el confirm le preguntamos algo si acepta le solicitamos el nombre y con un alert le decimos que ya quedo inscrito, si no acepta le mostramos un alert diciendo que en otra oportunidad será.</p>

## PARSEAR (convertir) CADENAS DE TEXTO A NUMEROS

let edad = <code>parseInt(prompt("edad"));</code> <code>console.log(edad+10);</code>	Si ingreso edad 22 me suma 10 y me guarda en consola 32	El prompt es para guardar datos de tipo string y con <code>parseInt</code> lo convierto en numeros enteros por eso los puedo sumar ya al final, si no tuviera el <code>parseInt</code> lo que hace es que me concatena me mostraría 2210
let edad = <code>parseInt(prompt("edad"));</code> <code>if(isNaN(edad)) {</code> <code>console.log ("dato erróneo");</code> <code>}</code>	<code>isNaN(variable);</code>	El <code>isNaN</code> lo que hace es que si ingreso un dato que no sea numérico me muestre en este caso dato erróneo. Es una condicion.
let edad = <code>parseFloat(prompt("edad"));</code> <code>console.log(edad+10);</code>	Si ingreso edad 22.9 me suma 10 y me guarda en consola 32.9	Sirve para lo mismo convertir string a numeros, pero en este caso con decimales o punto

## MÉTODOS MATH

let num = <code>Math.random();</code> <code>console.log(num);</code>	0.9726416077230542	Retorna un punto flotante, un número aleatorio dentro del rango [0, 1). Cada que muestre en pantalla me muestra un resultado diferente entre 0 y 1.
let opcionComputadora = <code>parseInt (Math.random() *3+1);</code> <code>console.log(opcionComputadora);</code>	3 "numero aleatorio en el rango que yo quiera"	Aquí modificamos el <code>random</code> lo multiplicamos por el numero que queramos en este caso 3 para que sea de 1 a 3, es de cero a 3 pero como le sumamos 1 ya empieza desde 1 y le coloco un <code>parseInt</code> para convertirlo en numeros enteros, eso quiere decir que si sale 1.25 mostrario 1 o 2.89 mostraría 2 y asi sucesivamente.
let num = <code>Math.round(20.49);</code> <code>console.log(num);</code>	20	Devuelve el valor del número dado redondeado al entero más cercano. Recibe un número, idealmente con punto flotante (float). Retorna el valor del número dado redondeado al entero más cercano.
function aleatorio (minimo, maximo) { return <code>Math.round(Math.random() * (maximo - minimo) + minimo);</code> } <code>console.log (aleatorio (4,9));</code>	7 "Crea numero aleatorio redondeando"	Aca redondeo con el <code>round</code> un numero aleatorio del <code>randon</code> en este caso 2 parámetros minimo y maximo, ósea me muestra numeros aleatorios redondeados entre en 4 y el 9 puede ser cualquier valor.
let numUno = <code>Math.max (10, 20,50);</code> <code>console.log(numUno);</code>	50	Recibe un conjunto de numeros y devuelve el mayor en este caso es 50, si llega a existir un string dentro del paréntesis ej: (10, 20,50); me da como resultado un NaN

## BUCLES ESPECIFICOS

### For .in y For. On

<pre>let persona = {   nombre: "Felipe erira",   edad: 30,   profesion: "Desarrollador fronted" } for (let caracteristica in persona) {    console.log(persona[caracteristica]); };</pre>	<pre>Felipe erira 30 Desarrollador fronted</pre>	Sirve únicamente para objetos, en el for el let caracteristica es una palabra cualquiera, lo que hace es que me recorre el objeto, pero para que me muestre los valores de colocar el objeto que es persona y dentro de corchetes simular como si fuera un array y colocar adentro el let ósea características.
<pre>let series = ["Betty la fea ", "pedro el escamoso", "pasión de gavilanes"];  for (let novela of series) {   console.log(novela); }</pre>	<pre>Betty la fea pedro el escamoso pasión de gavilanes</pre>	Sirve únicamente para arrays y string y lo que hace es que me lo recorre todo el array, el let novela es cualquier nombre en este caso novela por que debe ser algo acorde a lo que se muestra.
<pre>const texto = "hola" for (letra of texto) {   console.log(letra); }</pre>	<pre>h o l a</pre>	En el caso del of con el string lo que hace es que me muestra letra por letra o si quisiera saber la inicial despues del of seria texto [0] en la posición cero y asi me muestra la primera si quiera la segunda en la posición [1] y asi sucesivamente.

## OBJETO WINDOW Y DOCUMENT

El objeto Window, como bien lo indica su nombre, se refiere a toda la ventana que vemos en el navegador. El objeto Document se utiliza para leer y modificar, si así lo deseamos, el contenido de la ventana. Por ejemplo, lo podríamos usar para modificar nuestro HTML a través de sus clases y estilos.

<code>window.location</code>	Retorna un objeto con los detalles de la URL, la ruta, el href, etc.
<code>window.height</code> <code>window.width</code>	Retorna la altura y/o el ancho de la pantalla actual de la ventana.
<code>document.title</code>	Retorna el título del documento sobre el que estamos navegando.

## DOM SELECTORES

DOM (La representación en objetos que hace JavaScript del documento HTML.)

<code>querySelector ()</code>	<code>let titulo = document.querySelector('.title');</code>	Nos va a retornar el primer elemento del HTML que contenga la clase "title". El <code>querySelector</code> me selección el <b>PRIMER</b> elemento que encuentra si fueran párrafos solo me mostraría el primero, las clases igual que en css con punto y los id con #.
<code>querySelectorAll ()</code>	<code>let nombres= document.querySelectorAll('.name');</code>	Nos va a retornar un listado de elementos que coincidan con la búsqueda especificada. A diferencia del anterior nos va a mostrar por consola <b>TODOS</b> los <code>.name</code> y si fueran párrafos no mostraría todos los párrafos.
<code>getElementById ()</code>	<code>let marca= document.getElementById('marca');</code>	Nos va a retornar el elemento cuyo id coincida con el deseado. Solo se utiliza para los <b>ID</b> , no necesita el # solo el nombre
<code>getElementsByClassName ()</code>	<code>let nav-bar= document.getElementsByClassName("nav-bar")</code>	Nos va a retornar el elemento cuya <b>CLASE</b> coincida con el deseado. Solo se utiliza para las clases, no necesita el. solo el nombre

## MODIFICAR DOM (textos de html)

<code>document.querySelector("h3").innerText = "usando código"</code>	Aca lo que hago es que selecciono el primer h3 y le cambio el texto ejemplo si tuviera "hola mundo" como utilice innerText (es para agregar texto) "usando código" me elimina el hola mundo y me agrega este por que lo tengo =
<code>document.querySelector("h3").innerText += "usando código"</code>	Aca a diferencia del anterior le agrego un string más += y lo que hago es que le estoy agregando texto o sea quedaría "hola mundo usando código"
<code>document.querySelector("body").innerHTML += " &lt;mark&gt;Javascript ROCKS&lt;/mark&gt;"</code>	Aca igual como tengo += le estoy agregando texto, pero ya utilizo innerHTML que lo que significa es que voy a agregar texto, pero con formato html, o sea le puedo colocar etiquetas como <i> para que me aparezca cursiva y así sucesivamente

## PLANTILLAS DE TEXTO

<pre>const nombre= 'Mauro'; const miTemplate = `Mi nombre es \${nombre}`; console.log(miTemplate);</pre>	Es lo mismo que concatenar el resultado sería mi nombre es Mauricio, pero es con comillas cruzadas `` si las pone normal las tomaría como un string y no concatenaría. y en lugar del + para concatenar se coloca <code>\${variable}</code>
<pre>function suma (var1, var2) {   return var1 + var2 } const temp = `El resultado es \${suma (2,2)} `; console.log(temp);</pre>	Lo mismo, pero en este caso para una función.

## MODIFICAR ESTILOS

<code>document.querySelector("h2").style.textAlign = 'center'</code>	Aca seleccionamos el primer h2 que encontramos y se le aplica el style despues seguido de lo que quiera hacer en este caso textAling para alinear el texto a la mitad, la única diferencia es que en css es text-aling y en js se desaparece el guion y se comienza con mayúscula la otra palabra, y el color va como un string entre comillas.
<code>document.querySelector("body").style.backgroundColor = 'blue'</code>	Aca seleccione el body se le aplica el estilo con style y seguido de background color = "blue",
<pre>let libros = document.querySelectorAll("li")  for (let libro of libros) {   libro.style.color = "Orange" }</pre>	En este ejemplo se crea una variable libros que es igual a todos lo <li> que hay en la página porque es un querySelectorAll que significa que los muestra todos, y hago un bucle for para recorrer todos los <li>, cuando utilizo un querySelectorAll es como si creara un array y todos los <li> fueran los elementos del mismo por eso lo recorro y le aplico color a cada uno.
<pre>let items = document.querySelectorAll("li"); items [0].style.fontWeight = "bold"</pre>	Aca llamo items a todas las listas y en este caso como se comportar como un array llamo la primera lista que es en la posición [0] y le coloco style para ponerle negrilla

## MODIFICAR CLASES

<code>document.querySelector("div").classList.add('container')</code>	Aquí con classList.add lo que hago es que al primer div que me encuentre le voy a aplicar una clase llamada container, que ya debe existir en el css
<code>document.querySelector("h1").classList.remove('titulo')</code>	Aquí con classList.remove lo que hago es que el h1 que tiene una clase "titulo" se la voy a eliminar
<code>document.querySelector("h1").classList.toggle('titulo-tuneado')</code>	Aquí con classList.toggle lo que hago es que si en el h1 no hay una clase llamada asi se la agrego como el add, pero si llega a existir no la crea.
<pre>items.forEach(item=&gt;{   if (item.classList.contains("destacado")) {     item.classList.add("sombra");   } })</pre>	Con el forEach recorro el array, creo el elemento item y pongo una condicion que si item.classList.contains, el contains es para ver si existe una clase, y si existe le agrego una clase llamada sombra.

## ATRIBUTOS DINAMICOS

<b>hasAttribute ()</b>	let elemento = document.querySelector("#portada"); elemento.hasAttribute("src");	Es para <b>consultar</b> si existe un atributo src, quiere decir que si en el id portada existe una imagen devuelve true.
<b>getAttribute ()</b>	let elemento = document.querySelector("#portada"); elemento.getAttribute("src");	Lo que hace el get es <b>mostrar</b> , en este caso me va a mostrar el src del id portada ósea me va a mostrar esto: imagen_portada.jpg
<b>removeAttribute ()</b>	let elemento = document.querySelector("#portada"); elemento.removeAttribute("src");	En este caso me <b>elimina</b> el src del id portada, ósea me elimina la imagen
<b>setAttribute ()</b>	let elemento = document.querySelector("#portada"); elemento.setAttribute("src", "imagen_portada.jpg");	El set es para <b>agregar</b> en este caso el id portada se le va a agregar un src que es imagen y enseguida recibe la imagen

## NODOS

let texto2= prompt ("ingrese respuesta");  let element = document.createElement("p");  let textElement = document.createTextNode(texto2);  element.appendChild(textElement);  document.body.appendChild(element);	<b>createElement:</b> Lo que hace es crearme etiquetas en este caso un párrafo  <b>createTextNode:</b> Lo que hace es crearme el texto que va dentro del párrafo que en este caso lo declare arriba y lo va a escribir el usuario.  <b>appendChild:</b> Lo que hago es que elemento padre va a contener al hijo en este caso element que es el párrafo para a contener al texto que el text element, y despues digo que el body va a contener el elemento el párrafo en este caso.
---	--

## EVENTOS

<b>load</b>	<pre>window.addEventListener("load", function () { })</pre>	Permite que los eventos se ejecuten despues de que cargue toda la página html, todos los eventos deben ir dentro del load, ósea que comienzan a funcionar cuando cargue la página web
<b>This</b>	<pre>window.addEventListener("load", function () { let contacto = document.querySelector('.contacto') contacto.addEventListener("click", function(){ alert("tocaste el botón") console.log(this); }) })</pre>	Cuando yo imprimo el this, lo que hace es que me dice sobre que elemento estoy parado, este se utiliza mucho cuando voy a hacer algo sobre varias clases a la vez para que identifique en cual lo está haciendo. Se coloca al final del evento que vaya a realizar
<b>preventDefault</b>	<pre>window.addEventListener("load", function () { let contacto = document.querySelector('.contacto') contacto.addEventListener("click", function(e){ e.preventDefault(); }) })</pre>	Este lo que hace es que creo un parámetro llamado e de eventos y colocando e. <b>preventDefault ();</b> lo que hago es que cancelo el comportamiento normal de las cosas, por ejemplo, en menú que lo dirección a uno a una página diferente de contactos, con este evento ya no lo hace no pasa nada cuando le doy click en un menú no me lleva a ningún lado
<b>This (en un array)</b>	<pre>window.addEventListener("load", function () { let contacto = document.querySelectorAll('.contacto') for (let i = 0; i &lt; contacto.length; i++) { contacto[i].addEventListener("click", function(){ this.style.color= "red" }) })</pre>	Aca lo que hago es que recorro todas las clases contacto con un for desde 0 hasta contacto.length para que me cuente cuantos son, y con this y el evento que quiero realiza que en este caso en color red, lo que hace es que lo hace cada que doy click en uno, ósea lo hace uno por uno no todos a la vez.

## EVENTOS DE MOUSE

Evento	Ejemplo	Observación
<b>Click</b>	<pre>window.addEventListener("load", function () { let contacto = document.querySelector('.contacto') contacto.addEventListener("click", function(){ contacto.style.backgroundColor = prompt("elige su color preferido"); }) })</pre>	El click es para que haga lo que yo quiera cuando le de click en este caso cambia el fondo del menú, y pregunta que color quiere y en la ventana que sale se debe colocar el color que quiera en inglés y se pinta del mismo. Ejemplo si copia red se pintará el fondo del menú del mismo.
<b>mouseover</b>	<pre>window.addEventListener("load", function () { let contacto = document.querySelector('.contacto') contacto.addEventListener("mouseover", function(){</pre>	Con el mouseover se ejecuta el evento cuando pase el mouse o cursor por encima del menú en este caso-



	<pre> contacto.style.backgroundColor = prompt("elija su color preferido"); }) }) </pre>	
<b>Mouseover + contador</b>	<pre> Contador = document.querySelector(".contador") let num = 0; contador.innerHTML = num; Contador.addEventListener("mouseover", function () {   let timer = setInterval(function () {     if (num &gt; 824) {       clearInterval(timer)     } else       num++     contador.innerHTML = num;   }, 0.6) }) </pre>	Realizo el evento del mouse y activo el setInterval que es para que los numeros vayan de uno en uno, con el if le dijo que pare en 864 y el clearInterval lo detiene, Entonces se va a ver en pantalla un numero desde el cero de uno en uno hasta 864, el 0.6 de abajo significa que cada 0.6 milisengudo se va a cambiar de un numero a otro.
<b>mouseout</b>	<pre> window.addEventListener("load", function () {   let contacto = document.querySelector('.contacto')   contacto.addEventListener("mouseout", function(){     contacto.style.backgroundColor = prompt("elija su color preferido");   }) }) </pre>	Con el mouseout se ejecuta el evento cuando dejo de pasar el mouse por encima, ósea yo estoy con el curso para en él y no pasa nada, pero cuando lo quito de ahí se activa el evento.

## EVENTOS DEL TECLADO

Evento	Ejemplo	Observación
--------	---------	-------------

### El Key Me Dice Que Letra Toco

<b>keypress</b>	<pre> window.addEventListener("keypress", function(e){   if (e.key == "#") {     alert ("no se reciben #");   } }) </pre>	El keypress es un evento que se ejecuta cuando yo presiono cualquier tecla. En este caso le puse una condicion que llama al e (evento en parámetro de la funcion) punto <b>key</b> que significa que letra en específico todo y que si es igual a # me muestre un mensaje.
<b>keydown</b>	<pre> window.addEventListener("keydown", function(e){   alert ("se presionó la tecla: " +e.key); }) </pre>	El keydown cuando presiono una tecla me muestra cual fue, me muestra de a un carácter ejemplo: se presionó la tecla f
<b>keyup</b>	<pre> window.addEventListener("keyup", function(e){   alert ("se presionó la tecla: " +e.key); }) </pre>	El keyup cuando dejo de presionar la tecla, el suelto me muestra cual presione por lo que le estoy diciendo e.key, ósea que me muestre un solo carácter