

# DESARROLLO DE APLICACIONES EN INTERNET

## LABORATORIO N° 3

### JAVASCRIPT

CODIGO DEL CURSO:



<i>Alumno(s)</i>		<i>Nota</i>
<i>Malque Zurita Willians Leonardo</i>		
<i>Grupo</i>	<i>A</i>	
<i>Ciclo</i>	<i>III</i>	
<i>Fecha de entrega</i>		

## I.- OBJETIVOS:

- Conocer los aspectos básicos para el uso de JavaScript.
- Utilizar JavaScript en una página HTML.
- Entender la estructura de Javascript.

## II.- SEGURIDAD:



### **Advertencia:**

**En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red; así como la ingestión de alimentos o bebidas.**

## III.- FUNDAMENTO TEÓRICO:

Revise sus diapositivas del tema antes del desarrollo del laboratorio.

## IV.- NORMAS EMPLEADAS:

No aplica

## V.- RECURSOS:

- En este laboratorio cada alumno trabajará con un equipo con Windows
- Visual Studio Code
- Node JS
- Extensiones:
  - o Prettier
  - o Quokka.js
  - o LiveServer

## VI.- METODOLOGÍA PARA EL DESARROLLO DE LA TAREA:

- El desarrollo del laboratorio es individual.

## VII.- PROCEDIMIENTO:

## MARCO TEORICO

JavaScript es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor. Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

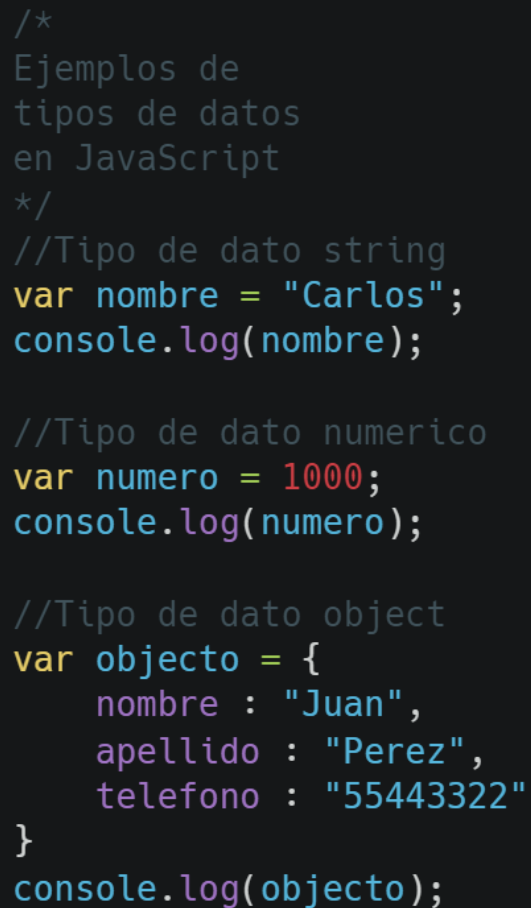
Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

## ACTIVIDADES

## USO EN NODE JS

Seguir las instrucciones del docente e iniciar con el desarrollo del laboratorio.

### 1. Tipos de Datos



```
/*
Ejemplos de
tipos de datos
en JavaScript
*/
//Tipo de dato string
var nombre = "Carlos";
console.log(nombre);

//Tipo de dato numerico
var numero = 1000;
console.log(numero);

//Tipo de dato object
var objecto = {
    nombre : "Juan",
    apellido : "Perez",
    telefono : "55443322"
}
console.log(objecto);
```

### 2. Tipos de Datos(parte 2)



```
//Tipo de dato boolean (true, false)
var bandera = false;
console.log(typeof bandera);

//Tipo de dato function
function miFuncion(){}
console.log(typeof miFuncion);

//Tipo de dato Symbol
var simbolo = Symbol("mi simbolo");
console.log(typeof simbolo);


//Tipo clase es una function
class Persona{
    constructor(nombre, apellido){
        this.nombre = nombre;
        this.apellido = apellido;
    }
}
console.log(typeof Persona);

//Tipo undefined
var x;
console.log(typeof x);

x = undefined;
console.log(typeof x);

//null = ausencia de valor
var y = null;
console.log(typeof y);
```

### 3. Arreglos y Datos Vacios



```
var y = null;
console.log(typeof y);

//arreglo en JavaScript
var autos = ['BMW', 'Audi', 'Volvo'];
console.log(autos);
console.log(typeof autos);

//Cadena vacia (empty string)
var z = '';
console.log(z);
console.log(typeof z);
```

### 4. Concatenar Caracteres



```
var nombre = 'Juan';
var apellido = 'Perez';

var nombreCompleto = nombre + ' ' + apellido;
console.log(nombreCompleto);

var nombreCompleto2 = 'Carlos' + ' ' + 'Lara';
console.log(nombreCompleto2);

var x = nombre + 2 + 4;
console.log(x);

x = nombre + (2 + 4);
console.log(x);

x = 2 + 4 + nombre;
console.log(x);
```

## 5. Concatenar Strings

```
let nombre;  
nombre = "Juan";  
console.log( nombre );  
  
const apellido = "Perez";  
//apellido = "Lara"; Error no se puede cambiar el valor de una constante
```

## 6. Reglas de declaración de Variables.

```
let nombreCompleto = "Juan Perez";  
let nombreCompleto = "Carlos Lara";  
console.log( nombreCompleto );  
console.log( nombreCompleto );  
  
let aNombreCompleto;  
let _nombreCompleto;  
let $nombreCompleto;  
//let 1nombreCompleto; no está permitido iniciar el nombre de una variable con numeros  
  
let ruptura = 10;
```

## 7. Operadores Aritmeticos

```
let a = 3, b = 2;  
let z = a + b;  
console.log("Resultado de la suma: " + z );  
  
z = a - b;  
console.log("Resultado de la resta: " + z);  
  
z = a * b;  
console.log( "Resultado de la mult:" + z);  
  
z = a / b;  
console.log( "Resultado de la division:" + z);  
  
z = a % b; //residuo de la division  
console.log( "Resultado de operacion modulo (residuo):" + z);  
z = a ** b;  
console.log( "Resultado de operador exponente:" + z);
```

## 8. Incremento y Decremento de variables

```
let a = 3, b = 2;
let z = a + b;

//Incremento
//Pre-incremento (el operador ++ antes de la variable)
z = ++a;
console.log(a);
console.log(z);

//Post-incremento (el operador ++ despues de la variable)
z = b++;
console.log(b);
console.log(z);

//Decremento
//Predecremento
z = --a;
console.log(a);
console.log(z);

//Postdecremento
z = b--;
console.log(b);
console.log(z);
```

## 9. Precedencia de Operadores

```
let a = 3, b = 2, c = 1, d = 4;

let z = a * b + c / d;
console.log(z);

z = c + a * b / d;
console.log( z );

z = (c + a) * b / c;
console.log(z);
```

## 10. Operadores de Comparación



```
let a = 3, b = 2, c = "3";

let z = a == c; // se revisa el valor sin importar el tipo
console.log(z);

z = a === c; // revisa los valores pero tambien los tipos
console.log(z);
```

## 11. Operadores de Diferencia




```
let a = 3, b = 2, c = 3;

let z = a != c; // se revisa el valor sin importar el tipo
console.log(z);

z = a !== c; // revisa los valores pero tambien los tipos
console.log(z);
```

## 12. Operador AND






```
let a = 15;
let valMin = 0, valMax = 10;


if( a >= valMin && a <= valMax ){
    console.log("Dentro de rango");
}
else{
    console.log("Fuera de rango");
}
```

### 13. Operador OR



```
//Ejemplo OR (||), regresa true si cualquier operando es true
let vacaciones = false, diaDescanso = true;
if( vacaciones || diaDescanso ){
    console.log("Padre puede asistir al juego del hijo");
}
else{
    console.log("El padre está ocupado");
}
```

### 14. Función NaN




```
let miNumero = "17";

let edad = Number(miNumero);
console.log( edad );

if( isNaN(edad)){
    console.log("No es un número");
}
else{
    if(edad >= 18){
        console.log("Puede votar");
    }
    else{
        console.log("Muy joven para votar");
    }
}

if( isNaN(edad)){
    console.log("No es un número");
}
else{
    let resultado = (edad >= 18)? "Puede votar" : "Muy joven para votar";
    console.log( resultado );
}
```


## 15. Condicional IF



```
let numero = 2;

if( numero == 1 ){
    console.log("Número uno");
}
else if( numero == 2 ){
    console.log("Número dos");
}
else if( numero == 3 ){
    console.log("Número tres");
}
else if( numero == 4 ){
    console.log("Número cuatro");
}
else{
    console.log("Número desconocido");
}
```

## 16. Condiciona Switch



```
let numero = 1;

let numeroTexto = 'Valor desconocido';

switch( numero ){
    case 1:
        numeroTexto = 'Número uno';
        break;
    case 2:
        numeroTexto = 'Número dos';
        break;
    case 3:
        numeroTexto = 'Número tres';
        break;
    case 4:
        numeroTexto = 'Número cuatro';
        break;
    default:
        numeroTexto = 'Caso no encontrado';
}

console.log(numeroTexto);
```

## 17. Ciclo While

```
let contador = 0;

while( contador < 3 ){
    console.log(contador);
    contador++;
}

console.log("Fin ciclo while");
```

## 18. Ciclo For

```
for(let contador = 0; contador < 3 ; contador++ ){
    console.log(contador);
}

console.log("Fin ciclo for");
```

## 19. Arreglos en Javascript

```
//let autos = new Array('BMW','Mercedes Benz','Volvo');
const autos = ['BMW','Mercedes Benz','Volvo'];
console.log(autos);
```

## 20. Accediendo a elemento de un Array

```
//let autos = new Array('BMW','Mercedes Benz','Volvo');
const autos = ['BMW','Mercedes Benz','Volvo'];
console.log(autos);

console.log(autos[0]);
console.log(autos[2]);

for(let i = 0; i < autos.length; i++){
    console.log(i + ' : ' + autos[i] );
}
```

## 21. Modificando Valores de un Array

```

//let autos = new Array('BMW','Mercedes Benz','Volvo');
const autos = ['BMW','Mercedes Benz','Volvo'];
console.log(autos);

console.log(autos[0]);
console.log(autos[2]);

for(let i = 0; i < autos.length; i++){
    console.log(i + ' : ' + autos[i] );
}

autos[1] = 'MerecedesBenz';
console.log(autos[1]);

autos.push('Audi');
console.log(autos);
```

## 22. Agregando Elementos a un array

```

//let autos = new Array('BMW','Mercedes Benz','Volvo');
const autos = ['BMW','Mercedes Benz','Volvo'];
console.log(autos);

console.log(autos[0]);
console.log(autos[2]);

for(let i = 0; i < autos.length; i++){
    console.log(i + ' : ' + autos[i] );
}

autos[1] = 'MerecedesBenz';
console.log(autos[1]);

autos.push('Audi');
console.log(autos);

console.log(autos.length);
autos[autos.length] = 'Cadillac';

console.log(autos);

autos[6] = 'Porsche';
console.log(autos);
```

## 23. Preguntar si es un arreglo



```
//let autos = new Array('BMW','Mercedes Benz','Volvo');
const autos = ['BMW','Mercedes Benz','Volvo'];
console.log(autos);

console.log(autos[0]);
console.log(autos[2]);

for(let i = 0; i < autos.length; i++){
    console.log(i + ' : ' + autos[i] );
}

autos[1] = 'MerecedesBenz';
console.log(autos[1]);

autos.push('Audi');
console.log(autos);

console.log(autos.length);
autos[autos.length] = 'Cadillac';

console.log(autos);

autos[6] = 'Porsche';
console.log(autos);

console.log(typeof autos);

console.log( Array.isArray(autos) );

console.log( autos instanceof Array);
```

## 24. Funciones en Javascript

```
miFuncion(4, 2);

//Declaración de la función
function miFuncion(a, b){
    console.log("Suma: " + (a + b));
}

//Llamando a la función
miFuncion(2, 3);
```

## USO EN HTML

1. En el editor de su preferencia, escriba en el archivo Hola.html el siguiente código y comente:

```
1  <!DOCTYPE html>
2  <html>
3
4      <head>
5          <meta charset="utf-8" />
6          <title>Page Title</title>
7          <script>
8              document.write('¡Hola mundo!');
9          </script>
10     </head>
11
12     <body>
13     </body>
14
15 </html>
```

2. Crea una función que se cargue al inicio y que muestre la suma de dos números.

```
1  <!DOCTYPE html>
2  <html>
3
4      <head>
5          <meta charset="utf-8" />
6          <title>Page Title</title>
7          <script>
8              function sumaNumeros() {
9                  let num1 = 5;
10                 let num2 = 10;
11
12                 document.write(num1 + num2);
13             }
14         </script>
15     </head>
16
17     <body onload="sumaNumeros()">
18     </body>
19
20 </html>
```

3. Crea dos radiobuttons, cuando uno de ellos se seleccione que muestre un mensaje diciendo que opción se seleccionó.

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="utf-8" />
6    <title>Page Title</title>
7    <script>
8      function muestraOpcion() {
9        var opciones = document.getElementsByName("eleccion");
10       console.log(opciones);
11       for (let i = 0; i < opciones.length; i++) {
12         if (opciones[i].checked) {
13           alert(opciones[i].value);
14         }
15       }
16     }
17   </script>
18 </head>
19
20 <body>
21   <input type="radio" name="eleccion" value="opcion1" checked onclick="muestraOpcion()">Opcion1<br/>
22   <input type="radio" name="eleccion" value="opcion2" onclick="muestraOpcion()">Opcion2<br/>
23   <input type="radio" name="eleccion" value="opcion3" onclick="muestraOpcion()">Opcion3<br/>
24 </body>
25
26 </html>
27
```

4. Crear dos input de números y un botón, al pulsar el botón, mostrar en un alert quien es el mayor de los dos:

```
1  <!DOCTYPE html>
2  <html>
3
4  <head>
5    <meta charset="utf-8" />
6    <title>Page Title</title>
7    <script>
8      function mayorNum() {
9        var num1 = parseInt(document.getElementById("num1").value);
10       var num2 = parseInt(document.getElementById("num2").value);
11       if (num1 && num2) {
12         if (num1 >= num2) {
13           if (num1 == num2) {
14             alert("Los numeros son iguales")
15           } else {
16             alert("El num1 es mayor")
17           }
18         } else {
19           alert("El num2 es mayor")
20         }
21       } else {
22         alert("Uno de los numeros no se relleno")
23       }
24     }
25   </script>
26 </head>
27 <body>
28   <form>
29     <input type="text" id="num1" />
30     <input type="text" id="num2" />
31     <button type="button" onclick="mayorNum()">¿Quien es el mayor?</button>
32   </form>
33 </body>
34 </html>
35
```



5. Crear un formulario para una casa de cambio que me permita seleccionar el tipo de moneda (dólar, soles, euros) a cambiar a (dólar, soles, euros), acompañado de un botón que al ser seleccionado muestre una alerta con el total a recibir.

**CASA DE CAMBIO**

Seleccionar moneda a enviar:

☒ Dólar      ☐ Euro

Ingresa monto:

#### TRABAJO PROPUESTO

Aplicar lo aprendido el día de hoy , a su presentación personal realizada en la semana 1 y 2.

<https://github.com/WilliansMalque/Desarrollo-de-Aplicaciones-en-Internet->

#### OBSERVACIONES:

- La práctica constante es clave para fortalecer nuestras habilidades en JavaScript
- Debemos dedicar tiempo a estudiar en profundidad la estructura y los conceptos fundamentales de JavaScript
- Debemos considerar realizar pruebas y validaciones exhaustivas para garantizar que nuestro código funcione de manera coherente y libre de errores

#### CONCLUSIONES:

- Se logro tener conocimiento inicial de JavaScript al implementar funciones simples para mostrar mensajes de alerta y realizar operaciones matemáticas.
- se logro incorporar JavaScript de manera efectiva en nuestra página de HTML, lo que nos permite agregar interactividad y dinamismo a nuestro sitio web.
- JavaScript nos permite cambiar y actualizar elementos HTML y CSS en tiempo real sin necesidad de recargar la página web.