

# DESARROLLO WEB

2017-I



SEMANA 01

## FUNDAMENTOS DE JAVASCRIPT

**Eric Gustavo Coronel Castillo**

ecoronel@uch.edu.pe - gcoronelc.blogspot.com

# FUNDAMENTOS DE JAVASCRIPT

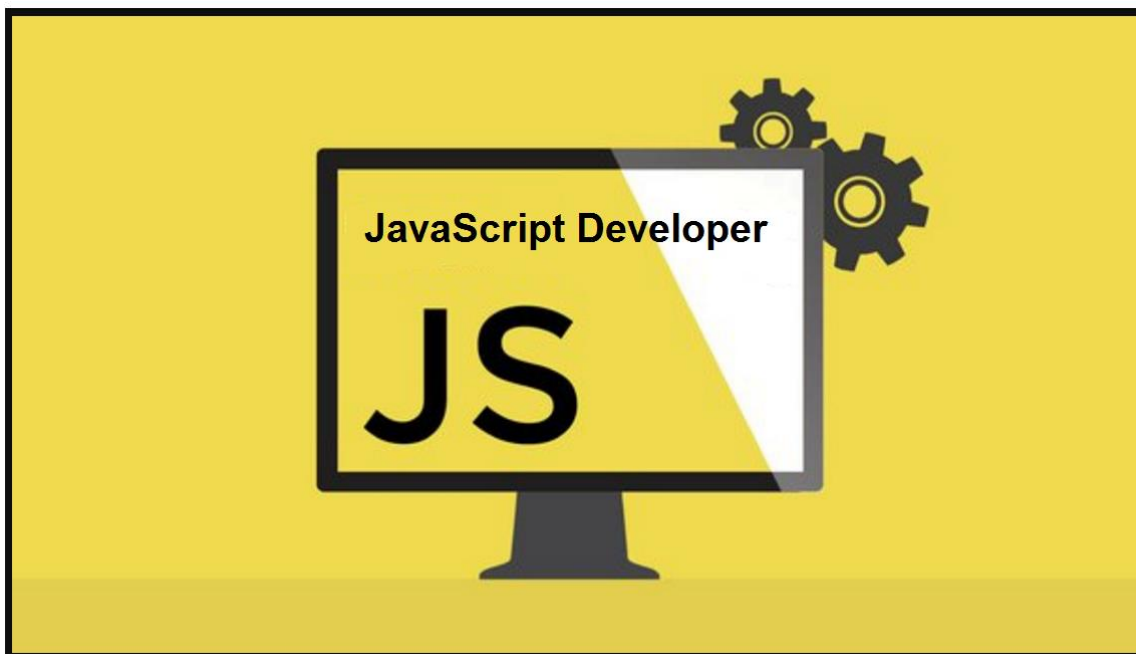
El objetivo de esta semana es que apliques JavaScript para solucionar problemas de sencillos.

## ÍNDICE

<b>OBJETIVO.....</b>	<b>5</b>
<b>INTRODUCCIÓN .....</b>	<b>6</b>
<b>¿QUÉ ES JAVASCRIPT? .....</b>	<b>7</b>
<b>JAVASCRIPT SE INTEGRA CON LAS PÁGINAS WEB .....</b>	<b>8</b>
<b>JAVASCRIPT ES UN LENGUAJE INTERPRETADO.....</b>	<b>9</b>
<b>JAVASCRIPT ES CASE SENSITIVE.....</b>	<b>10</b>
<b>COMENTARIOS .....</b>	<b>13</b>
<b>VARIABLES.....</b>	<b>14</b>
<b>OPERADORES .....</b>	<b>16</b>
OPERADORES ARITMÉTICOS.....	16
OPERADORES DE ASIGNACIÓN .....	17
OPERADORES DE COMPARACIÓN .....	17
OPERADORES LÓGICOS .....	18
OTROS OPERADORES .....	18
<i>Condicional ( ? : ) .....</i>	<i>18</i>
<i>Concatenación ( + ) .....</i>	<i>18</i>
<b>MENSAJES AL USUARIO.....</b>	<b>19</b>
<b>LECTURA DE DATOS.....</b>	<b>20</b>
<b>CONDICIONALES.....</b>	<b>21</b>
OPERADOR TERNARIO .....	21
ESTRUCTURA IF.....	21
<b>BUCLES .....</b>	<b>23</b>
BUCLE: FOR .....	23

BUCLE: WHILE .....	24
BUCLE: DO WHILE .....	25
BUCLE: FOR IN .....	26
SENTENCIAS BREAK Y CONTINUE.....	27
<i>Sentencia break</i> .....	27
<i>Sentencia continue</i> .....	27
INSTRUCCIÓN: WITH.....	28
INSTRUCCIÓN: SWITCH .....	29
<b>EJERCICIOS PROPUESTOS .....</b>	<b>31</b>

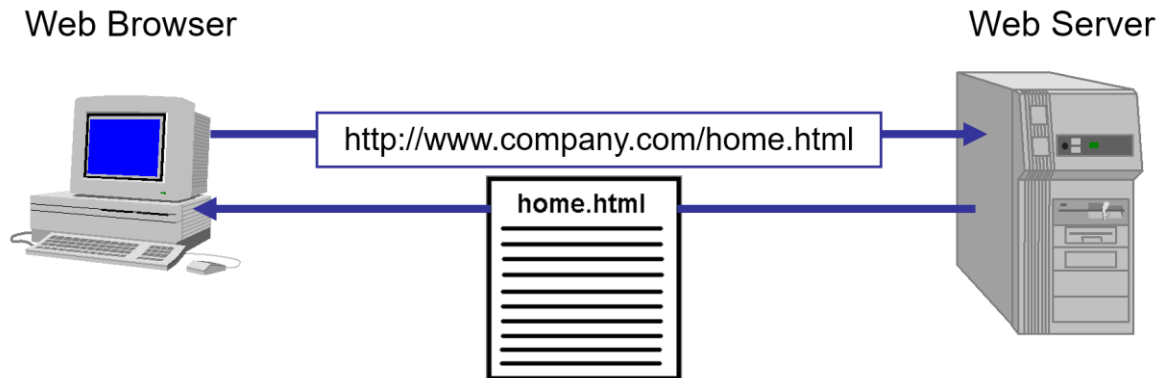
## Objetivo



El objetivo de esta semana es que apliques JavaScript (JS) para solucionar problemas de sencillos.

Siendo JS el lenguaje utilizado para desarrollar las interfaces de usuario en las aplicaciones Web, junto con HTML y CSS, es de vital importancia que tú aprendas los fundamentos de JS para que no tengas problemas a utilizar los framework de JS.

## Introducción



- Es un complemento de HTML
- Permite a las páginas realizar algunas tareas por sí misma, sin necesidad de sobrecargar el servidor del cual dependen.
- Permiten realizar cálculos simples, formatear texto para que pueda ser leído por distintas personas de manera distinta.
- Provee de medios para configurar la visualización de una página.

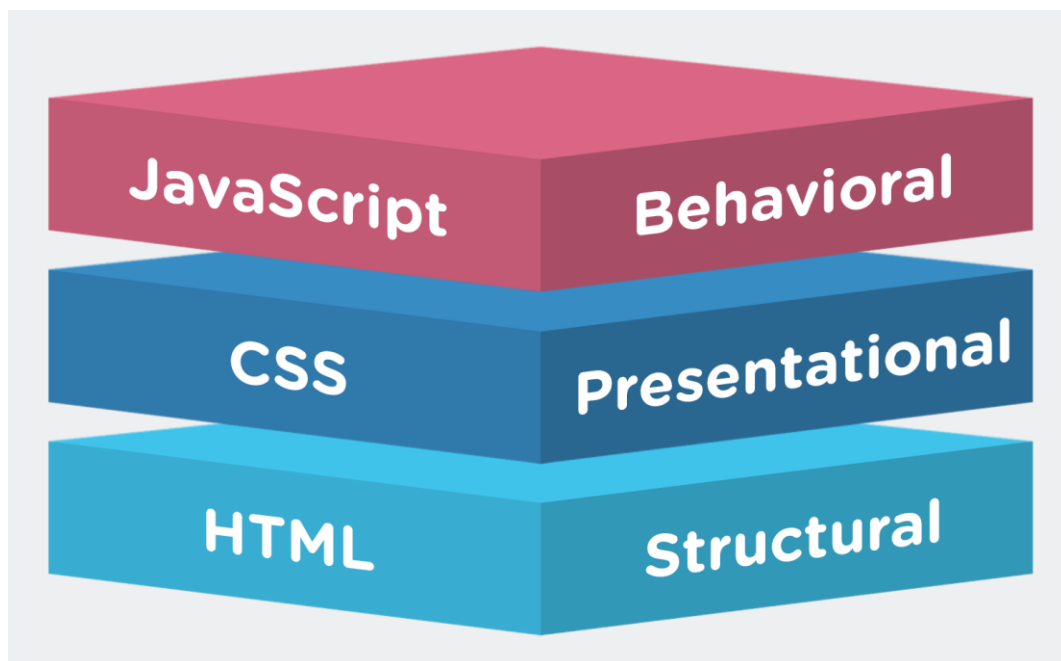
## ¿Qué es JavaScript?



- Es un Lenguaje de tipo script, basado en objetos, y guiado por eventos, para ser aplicado en las páginas Web.
- JavaScript agrega dinamismo y permite procesamiento de datos.
- Proporciona un conjunto de objetos que permiten trabajar muy bien con los elementos de un documento Web.

```
<SCRIPT LANGUAGE="JavaScript">  
  
    // Aquí escribes el código JavaScript  
  
</SCRIPT>
```

## JavaScript se Integra con las Páginas Web

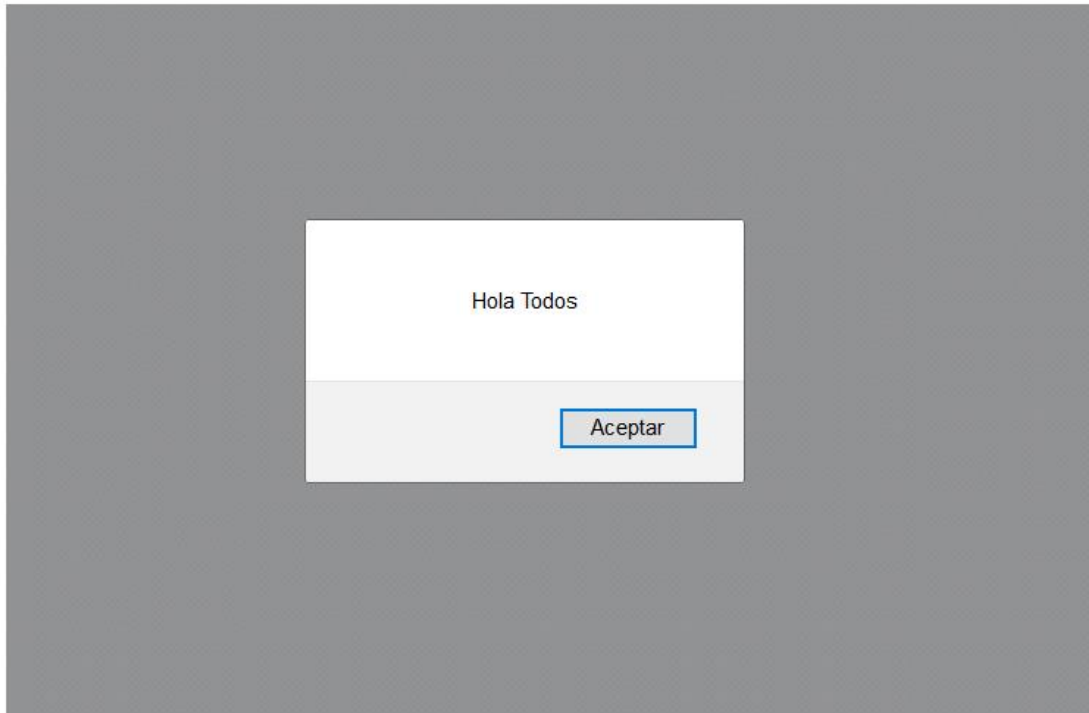


Programas en JavaScript están incluidos en la página Web donde se ejecutará.

```
<BODY>

<SCRIPT LANGUAGE="JavaScript">
// Aquí escribes el código JavaScript
</SCRIPT>
</BODY>
```

## JavaScript es un Lenguaje Interpretado



El código JS es interpretado por el Browser.

### **Ventajas**

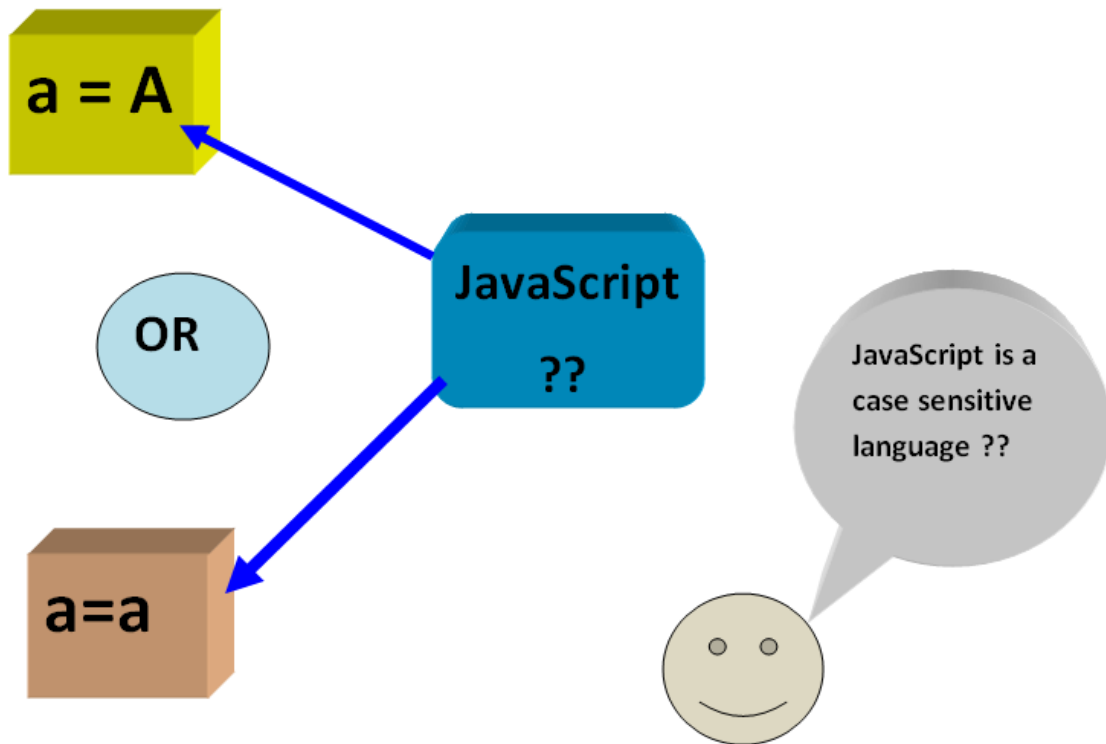
- En todo momento es posible actualizar el código fuente.
- Viaja con el documento HTML.
- Son fáciles de aprender.

### **Desventajas**

- El código fuente siempre será visible para todos.



## JavaScript es Case Sensitive



**Case Sensitive** es el término inglés para expresar que un lenguaje de programación es capaz de distinguir entre mayúsculas y minúsculas, y se comporta de manera estricta a la hora de ejecutarlas.

Las siguientes variables son reconocidas como distintas:

- Cont
- cont
- CONT

## Ejemplo 1

Archivo: ejm01.html

```
<script>

y= 5;
Y = 10;

r1= y + y;
r2 = y + Y;

document.write( "<p> y = " + y + "</p>" );
document.write( "<p> Y = " + y + "</p>" );
document.write( "<p> y + y = " + r1 + "</p>" );
document.write( "<p> y + Y = " + r2 + "</p>" );

</script>
```

El resultado es el siguiente:

```
y = 5
Y = 5
y + y = 10
y + Y = 15
```

## Ejemplo 2

**Archivo: ejm02.html**

---

```
<HTML>
<HEAD><TITLE>JavaScript</TITLE></HEAD>
<BODY>
  Mensaje de JavaScript:
  <SCRIPT LANGUAGE="JavaScript">
    <!-- // Inicia Programa
      document.write("<H1>Bienvenido a JavaScript</H1>");
    // Fin de Programa -->
  </SCRIPT>
</BODY>
</HTML>
```

## Ejemplo 3

**Archivo: prog.js**

---

```
<!-- // Inicia Programa
  document.write("<H1>Bienvenido a JavaScript</H1>");
// Fin de Programa -->
```

**Archivo: ejm03.html**

---

```
<HTML>
<HEAD>
  <TITLE>JavaScript</TITLE>
</HEAD>
<BODY>
  Mensaje de JavaScript.
  <SCRIPT LANGUAGE="JavaScript" SRC="P0101.js"></SCRIPT>
</BODY>
</HTML>
```

## Comentarios

Comentarios en una línea:

```
<SCRIPT LANGUAGE="JavaScript">

    // Esto es un Comentario
    document.getElementById("id").innerHTML = "1";

    // Esto tambien es un comentario y no es tenido en cuenta.
    document.getElementById("name").innerHTML = "Gustavo Coronel";

    var saldo = 5;           // Esta variable inicia con un valor de 5
    var res = saldo + 2;     // Esta variable guarda el valor de la variable saldo mas 2

</SCRIPT>
```

Comentarios Multi-Línea

```
<SCRIPT LANGUAGE="JavaScript">

    /*
    El siguiente código realiza:

    Asignacion de valor 1 al elemento id
    Asignacion de valor Gustavo Coronel al elemento name
    Datos del programador:
    Sitio Web: gcoronelc.blogspot.com
    Email: gcoronelc@gmail.com
    */

    document.getElementById("id").innerHTML = "1";

    document.getElementById("name").innerHTML = "Gustavo Coronel.";

</SCRIPT>
```

## Variables

- Cuando se declara una variable, no se especifica su tipo:  
`var nombreVariable;`
- El nombre de las variables tiene que empezar por una letra o guión bajo:

Variables bien definidas	Variables mal definidas
tres_mosqueteros	3mosqueteros
Nombre	*nombre
_id	(id)

El tipo de la variable está determinado por el dato almacenado en ella.

En JavaScript el nombre de una variable es sensible a las mayúsculas y minúsculas.

Los tipos de datos disponibles son:

- Numérico (numeric)
- Cadena (string)
- Booleano (boolean)
- Objeto (object)
- Función (function)
- Nulo (null)

## Ejemplo 4

En este ejemplo se ilustra cómo identificar el tipo de datos de una variable.

**Archivo: ejm03.html**

```
<HTML>
<HEAD><TITLE>Variables</TITLE></HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
<!-- // Inicia Programa

    var var1 = 20;
    var var2 = "Alianza Lima";
    var var3 = true;

    document.write( "var1 es " + typeof var1 );
    document.write( "<BR>var2 es " + typeof var2 );
    document.write( "<BR>var3 es " + typeof var3 );

// Fin de Programa -->
</SCRIPT>
</BODY>
</HTML>
```

El resultado es:

```
var1 es number
var2 es string
var3 es boolean
```

## Operadores

### Operadores Aritméticos

Operador	Operación	Ejemplo
+	Suma	$c = a + b;$
-	Resta	$c = a - b;$
*	Multiplicación	$c = a * b;$
/	División	$c = a / b;$
%	Resto	$c = a \% b;$
++	Incremento	$a++; b = a++; c = ++b;$
--	Decremento	$a--; b = a--; c = --b;$

### Ejemplo 5

Archivo: ejm05.html

```
<HTML>
<HEAD><TITLE>Operadores Aritméticos</TITLE></HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">

    var a = 40, b = 15;
    document.write("a = ", a, "<BR>");
    document.write("b = ", b, "<BR>");
    document.write(a, " + ", b, " = ", a+b, "<BR/>");
    document.write(a, " - ", b, " = ", a-b, "<BR/>");
    document.write(a, " * ", b, " = ", a*b, "<BR/>");
    document.write(a, " / ", b, " = ", a/b, "<BR/>");
    document.write(a, " % ", b, " = ", a%b, "<BR/>");

</SCRIPT>
</BODY>
```

</HTML>

## Operadores de Asignación

Operador	Operación	Ejemplo	Equivalente
=	Asignación	c = a + b;	
+=	Suma-Asignación	c += a;	c = c + a;
-=	Resta-Asignación	c -= a;	c = c - a;
*=	Multiplicación-Asignación	c *= a;	c = c * a;
/=	División-Asignación	c /= a;	c = c / a;
%=	Resto-Asignación	c %= a;	c = c % a;

## Operadores de Comparación

Operador	Operación	Ejemplo
==	Igual que	c = a == b;
!=	Diferente que	c = a != b;
>	Mayor que	c = a > b;
>=	Mayor o igual que	c = a >= b;
<	Menor que	c = a < b;
<=	Menor o igual que	c = a <= b;

Dependiendo del valor de a y b, c puede tomar valor **true** o **false**.

Se utilizan principalmente para la toma de decisiones.



## Operadores Lógicos

Operador	Operación	Ejemplo
&&	Y Lógico (And)	c = a && b;
	O Lógico (Or)	c = a    b;
!	Negación (Not)	c = !a;

Dependiendo del valor de a y b, c puede tomar valor **true** o **false**.

Se utilizan principalmente para la toma de decisiones.

## Otros Operadores

### Condicional ( ? : )

Formato:

```
Condición?Expresión1:Sxpresión2
```

Ejemplo:

```
var a = 20;  
var b = 15;  
c = (a>b)?a:b;
```

### Concatenación ( + )

Formato:

```
Cadena1 + Cadena2
```

Ejemplo:

```
ElMejor = "Alianza" + " " + "Campeón";
```

## Mensajes al Usuario

Formato 1:

```
document.write("mensaje")
```

Formato 2:

```
alert("mensaje")
```

### Ejemplo 6

Archivo: ejm06.html

```
<HTML>
<HEAD>
  <meta charset="utf-8">
  <TITLE>Mensajes JavaScript</TITLE>
  <SCRIPT LANGUAGE=javascript TYPE="text/javascript">
    alert("¡Bienvenidos a JavaScript!")
  </SCRIPT>
</HEAD>
<BODY BGCOLOR="blue" TEXT="white">
<H1>
  <SCRIPT LANGUAGE=javascript TYPE="text/javascript">
    document.write("¡Hola Mundo!")
  </SCRIPT>
</H1>
</BODY>
</HTML>
```

## Lectura de Datos

Sintaxis:

```
prompt( mensaje [, EntradaPredeterminada ] )
```

Muestra un cuadro de diálogo para que el usuario pueda ingresar un dato.

Retorna el dato ingresado.

### Ejemplo 7

Verificar ingreso de datos.

**Archivo: ejm07.html**

```
<SCRIPT LANGUAGE="JavaScript">

var nombre

nombre = prompt("Ingrese su nombre","Digite aquí su nombre");

if (typeof nombre == "string"){
    document.write("<H3>Hola ", nombre, ", bienvenido a JavaScript</H3>" );
} else{
    document.write("<H3>No has ingresado tú nombre</H3>");
}

</SCRIPT>
```

## Condicionales

### Operador ternario

Sintaxis

```
(condición) ? sentenciaV : sentenciaF
```

### Estructura if

```
if (condición) {  
    sentenciasV  
} else {  
    sentenciasF  
}
```

### Ejemplo 8

Determinar si un número es par.

**Archivo: ejm08.html**

```
<SCRIPT LANGUAGE="JavaScript">  
    var n = prompt("ingrese un numero entero");  
    var r = (n%2==0) ? "Es Par" : "Es Impar";  
    document.write( n + " " + r);  
</SCRIPT>
```

## Ejemplo 9

Mayor de tres números.

**Archivo: ejm09.html**

---

```
<SCRIPT LANGUAGE = "JavaScript">
// Variables
var n1, n2, n3, mayor;

// Datos
n1 = prompt( "N1", "" );
n2 = prompt( "N2", "" );
n3 = prompt( "N3", "" );

// Proceso
mayor = n1;
if (n2 > mayor){
    mayor = n2
}
if (n3 > mayor){
    mayor = n3
}

// Reporte
document.write("<H3> El mayor valor de: ");
document.write(n1, " , ", n2, " y ", n3 + " es:</H3>");
document.write("<H1>" + mayor + "</H1>");
</SCRIPT>
```

## Bucles

### Bucle: For

Sintaxis:

```
for(exp_inicialización; exp_condición; exp_bucle){  
    instrucciones  
}
```

Ejemplo

```
for( k=1; k <= 100; k++ ) {  
    document.write( k ) ;  
}
```

### Ejemplo 10

Calcular el factorial de un número.

Archivo: **ejm10.html**

```
<SCRIPT LANGUAGE ="JavaScript">  
    // Variables  
    var n, f=1, i;  
  
    // Dato  
    n = prompt( "Factorial de ?", "" );  
  
    // Proceso  
    for( i=2; i <= n; i++ ){  
        f *= i;  
    }  
  
    // Reporte  
    document.write( "factorial de " + n + " es " + f );</SCRIPT>
```

## Bucle: while

Sintaxis:

```
while ( condición ) {  
    Instrucciones  
}
```

### Ejemplo 11

Calcular el MCD de dos números.

Archivo: **ejm11.html**

```
<SCRIPT LANGUAGE ="JavaScript">  
  // Variables  
  var n1, n2, mcd, n1a, n2a;  
  // Datos  
  n1 = prompt( "Un numero entero:", "" );  
  n2 = prompt( "Otro número entero;", "" );  
  // Proceso  
  n1a = n1;  
  n2a = n2;  
  while (n1 != n2) {  
    if (n1 > n2) {  
      n1 -= n2;  
    }  
    else {  
      n2 -= n1;  
    }  
  }  
  mcd = n1;  
  // Reporte  
  document.write( "<p>N1=" + n1a + "</p>" );  
  document.write( "<p>N2=" + n2a + "</p>" );  
  document.write( "<p>MCD=" + mcd + "</p>" );  
</SCRIPT>
```

## Bucle: Do While

Sintaxis:

```
do {  
    Instrucciones  
} while ( condición );
```

### Ejemplo 12

Mostrar los números pares.

**Archivo: ejm12.html**

```
<SCRIPT LANGUAGE ="JavaScript">  
var k=0;  
document.write("<H3>Números pares entre 1 y ...</H3>");  
do {  
    k=k+50;  
    for (i=k+1-50 ; i<=k; i++){  
        if (i%2==0) { document.write(i + ","); }  
        if (i%10==0) { document.write("<BR>"); }  
    }  
} while (confirm("¿Deseas ver los 50 próximos pares?"));  
document.write("<H3>Fin</H3>");  
</SCRIPT>
```



## Bucle: For In

Sintaxis:

```
for( var propiedad in objeto ){  
    Instrucciones  
}
```

### Ejemplo 13

Mostrar las propiedades del Objeto **document**.

Archivo: **ejm13.html**

---

```
<SCRIPT LANGUAGE ="JavaScript">  
    var k = 0;  
    document.write("<H3>Propiedades del objeto document ...</H3>");  
    for ( propiedad in document ){  
        k++;  
        document.write(k,".- document." + propiedad + " = "  
            + document[propiedad] + "<BR>");  
    }  
    document.write("<BR>Fin")  
</SCRIPT>
```

## Sentencias break y continue

### Sentencia break

Finaliza la ejecución de un bucle.

### Sentencia continue

Lleva el control a la expresión del bucle para ser nuevamente evaluada, ignorando las instrucciones que se encuentran posteriores a la instrucción continue.

### Ejemplo 14

Números Pares.

**Archivo:** ejm14.html

```
<SCRIPT LANGUAGE = "JavaScript">
  var n, k = 1, cont = 0;
  n = prompt( "Cuantos numeros pares?", "" );
  while (true){
    if ( (k % 2) == 0 ) {
      document.write( k + "," );
      cont++;
      if ( (k % 5) == 0 ) { document.write( "<BR>" ); }
    }
    k++;
    if (cont < n) { continue; }
    if (cont == n) { break; }
    document.write( "Nunca se ejecuta" );
  }
  document.write("<H3>Fin</H3>");
</SCRIPT>
```

## Instrucción: WITH

Se utiliza para evitar escribir la referencia a un objeto cuando se está accediendo a las propiedades o métodos de dicho objeto.

### Sintaxis:

```
with ( objeto ) {  
    Instrucciones  
}
```

### Ejemplo 15

Propiedades del objeto **document**.

**Archivo: ejm15.html**

```
<HTML>  
<HEAD>  
    <TITLE>Estructuras de Control</TITLE>  
</HEAD>  
<BODY>  
<SCRIPT LANGUAGE = "JavaScript">  
    with ( document ) {  
        write("<H3>Utilizando la sentencia \"WITH\"</H3>");  
        write("título: \"\" + title + \"\"<BR><BR>");  
        write("color de fondo: " + bgColor + "<BR>");  
        write("color de primer plano: " + fgColor + "<BR>");  
        write("color de los enlaces visitados: " + vlinkColor + "<BR>");  
        write("color de los enlaces no visitados: " + alinkColor + "<BR>");  
        write("<H3>Fin</H3>");  
    }  
</SCRIPT>  
</BODY>  
</HTML>
```

## Instrucción: Switch

Se utiliza para comparar un dato entre un conjunto de posibles valores.

### Sintaxis:

```
switch ( DatoComparar ) {  
  case valor1:  
    Instrucciones;  
    break;  
  case valor2:  
    Instrucciones;  
    break;  
  -----  
  -----  
  case valorn:  
    Instrucciones;  
    break;  
  default:  
    Instrucciones  
}
```

## Ejemplo 16

Identificación de usuario.

**Archivo: ejm16.html**

---

```
<SCRIPT LANGUAGE=javascript TYPE="text/javascript">
  document.write( "Ingreso al Sistema<BR>");
  usuario = prompt("Por favor, identifique: ", "invitado");
  switch (usuario.toUpperCase()) {
    case "ADMINISTRADOR":
      document.write("Bienvenido Administrador...<BR>");
      break;
    case "OPERADOR":
      document.write("Bienvenido Operador...<BR>");
      break;
    case "INVITADO":
      document.write("Bienvenido Invitado...<BR>");
      break;
    default:
      document.write("No estás autorizado...<BR>");
      break;
  }
  document.write("Fin del proceso.")
</SCRIPT>
```

---

## Ejercicios Propuestos

---

1. Desarrollar un programa para mostrar la tabla de multiplicar de N, el programa debe solicitar el valor de N.
2. Realizar un programa para calcular el factorial de un número.
3. Realizar un programa para imprimir N términos de la serie de Fibonacci, el programa debe solicitar el valor de N, y debe validar a que sea mayor que 2.
4. Desarrollar un programa que solicite un número entero N, y luego determine si se trata de un número primo.
5. Realizar un programa para calcular el importe de una venta.