

 JavaScript	<b>MEDIA TÉCNICA DESARROLLO DE SOFTWARE</b> <b>GUIA JAVASCRIPT No.2</b> <b>MODULO: Elementos de software web</b> <b>TEMA: JavaScript</b>	
---	---	--

**DOCENTE:** Juan Carlos Pérez Pérez

**ESTUDIANTE :** \_\_\_\_\_ **FECHA:** \_\_/\_\_/\_\_

**JUSTIFICACIÓN:** Se pretende con esta guía hacer conozca y aplique javaScripts y algunas funcionalidades en html al lado del cliente.

**OBJETIVOS :** Reconocer la estructura básica.  
 Definir tipos de datos.  
 Capturar datos.  
 Realizar Operaciones básicas.  
 Estructuras básicas.

**CONTENIDO:**

“JavaScript, al igual que Flash, Visual Basic Script, es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML (lenguaje para el diseño de páginas de Internet). Al ser la más sencilla, es por el momento la más extendida. JavaScript no es un lenguaje de programación propiamente dicho como java, C, C++, Delphi, etc.

Es un lenguaje script u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto y planillas de cálculo. No se puede desarrollar un programa con JavaScript que se ejecute fuera de un Navegador, aunque en este momento comienza a expandirse a otras áreas como la programación en el servidor con Node.js, JavaScript es un lenguaje interpretado que se embebe en una página web HTML. Un lenguaje interpretado significa que a las instrucciones las analiza y procesa el navegador en el momento que deben ser ejecutadas.”

Ahora la mayoría de los frameworks orientados a dispositivos móviles están basados en JavaScript

Tomado de: [http://www.javascriptya.com.ar/temarios/descripcion.php?cod=2\[01-02-2011](http://www.javascriptya.com.ar/temarios/descripcion.php?cod=2[01-02-2011) 09:45:a.m]

**Ejemplo1**

```
<html><head></head>
<body>
<script type="text/javascript">
document.write('Hola Juan');
</script>
</body> </html>
```

El programa en JavaScript debe ir encerrado entre la marca script e inicializada la propiedad type con la cadena text/javascript: `<script type="text/javascript">`

`</script>`

“Para imprimir caracteres sobre la página debemos llamar al comando 'write' del objeto document. La información a imprimirse debe ir entre comillas y encerrada entre paréntesis. Todo lo que indicamos entre comillas aparecerá tal cual dentro de la página HTML.

Es decir, si pedimos al navegador que ejecute esta página mostrará el texto 'Hola Mundo'. Cada vez que escribimos una instrucción finalizamos con el carácter punto y coma.

ES IMPORTANTISIMO TENER EN CUENTA QUE JavaScript es SENSIBLE A MAYUSCULAS Y MINUSCULAS. NO ES LO MISMO ESCRIBIR:  
document.write que DOCUMENT.WRITE (la primera forma es la correcta, la segunda forma provoca un error de sintaxis).

Nos acostumbraremos a prestar atención cada vez que escribamos en minúsculas o mayúsculas para no cometer errores sintácticos.“

---

Tomado de : [http://www.javascriptya.com.ar/temarios/descripcion.php?cod=2\[01-02-2011 09:45:a.m\]](http://www.javascriptya.com.ar/temarios/descripcion.php?cod=2[01-02-2011 09:45:a.m]) Cabe anotar que los java script no muestran en la mayoría errores , simplemente el explorador web, no lo interpreta, por lo tanto no ejecuta ó realiza la acción.

### **Definición de variables:**

Una variable es un espacio de memoria ram en el pc donde hay un valor. Consta de un nombre y pertenece a un tipo de datos (numérico, cadena de caracteres, etc.)

Tipos de variables más comunes: Una

variable puede almacenar:

Valores Enteros (100, 260, etc.)

Valores Reales (1.24, 2.90, 5.01, etc.)

Cadenas de caracteres ('Juan', 'Compras', 'Listado', etc.)

Valores lógicos (true,false) Existen otros tipos de variables.

Las variables son nombres que ponemos a los lugares donde almacenamos la información. En JavaScript, deben comenzar por una letra o un subrayado (\_), pudiendo haber además dígitos entre los demás caracteres. Una variable no puede tener el mismo nombre de una palabra clave del lenguaje.

Una variable se define anteponiéndole la palabra clave var: var dia; se pueden declarar varias variables en una misma línea: var dia, mes, año; a una variable se la puede definir e inmediatamente inicializarla con un valor: var edad=20; ó en su defecto en dos pasos: var edad; edad=20; Debemos tener en cuenta las reglas al definir en nombre de una variable:

### **Impresión de variables en una página HTML:**

Para mostrar el contenido de una variable en una página utilizamos el objeto document y llamamos a la función write.

Ejemplo 2:

```
<html>
  <head>
  </head>
  <body>
  <script type="text/javascript">
var nombre='Juan'; var
edad=10; var altura=1.92;
var casado=false;
document.write(nombre);
document.write('<br>');
```

```

document.write(edad);
document.write('<br>');
document.write(altura);
document.write('<br>');
document.write(casado);
</script>
</body>
</html>

```

Cuando imprimimos una variable, no la debemos disponer entre simples comillas (en caso de hacer esto, aparecerá el nombre de la variable y no su contenido)

Los valores de las variables que almacenan nombres (es decir, son cadenas de caracteres) deben ir encerradas entre comillas simples o dobles. Los valores de las variables enteras (en este ejemplo la variable edad) y reales no deben ir encerradas entre comillas. Cada instrucción finaliza con un punto y coma.

Las variables de tipo boolean pueden almacenar solo dos valores: true o false. El resultado al visualizar la página debe ser 4 líneas similares a éstas:

```

Juan
10 1.92
false

```

Es decir que se muestran los contenidos de las 4 variables. Una variable es de un tipo determinado cuando le asignamos un valor: var edad=10;

Es de tipo entera ya que le asignamos un valor entero.

var nombre='juan'; Es de tipo cadena.

Para mostrar el contenido de una variable en una página debemos utilizar la función 'write' que pertenece al objeto document.

### Captura de datos por teclado:

Para la entrada de datos por teclado tenemos la función prompt. Cada vez que necesitamos ingresar un dato con esta función, aparece una ventana donde cargamos el valor. Hay otras formas más sofisticadas para la entrada de datos en una página HTML, pero para el aprendizaje de los conceptos básicos de JavaScript nos resultará más práctica esta función.

### Ejemplo 3:

```

<html>
<head> <meta http-equiv="Content-type" content="text/html;charset=UTF-8"></head>
<body>
<script type="text/javascript">
var nombre; var edad;
nombre=prompt('Ingrese su nombre:');
edad=prompt('Ingrese su edad:');
document.write('Hola ');
document.write(nombre); document.write('
asi que tienes ');
document.write(edad);
document.write(' años');
</script>
</body>
</html>

```

### La sintaxis de la función prompt es:

<variable que recibe el dato>=prompt(<mensaje a mostrar en la ventana>,<valor inicial a mostrar en la ventana>); Como se vió en el ejemplo.

**nombre=prompt('Ingrese su nombre:');**

La función prompt tiene dos parámetros: uno es el mensaje y el otro el valor inicial a mostrar.

### Operaciones básicas secuenciales:

Son idénticas como se realizan en Java, pero sin la exigencia del mismo veamos el siguiente ejemplo:

```
<html>
<head>
<meta http-equiv="Content-type" content="text/html;charset=UTF-8"> </head>
<body>
<script type="text/javascript">
var a=0; var b=0; var
resultado=0.0;
a=prompt('Valor a:');
b=prompt('Valor b:');
/*Multiplicación*/
resultado=parseInt(a)*parseInt(b);
document.write('<br>');
document.write('La Multiplicación de ',a : ',a,' * ', b : ',b,' Es : ', resultado);
/*Suma*/
resultado= parseInt(a)+parseInt(b);
document.write('<br>');
document.write('La suma de ',a : ',a,' + ', b : ',b,' Es : ', resultado);
/*Resta*/
resultado= parseInt(a)-parseInt(b);
document.write('<br>');
document.write('La resta de ',a : ',a,' - ', b : ',b,' Es : ', resultado);
/*Division*/
resultado= parseInt(a)/parseInt(b);
document.write('<br>');
document.write('La División*/ de ',a : ',a,' / ', b : ',b,' Es : ', resultado); </script>
</body>
</html>
```

Lo primero que debemos tener en cuenta es que si queremos que el operador + sume los contenidos de los valores numéricos ingresados por teclado, debemos llamar a la función parseInt ó parseFloat y pasar como parámetro las variables **a** y **b** sucesivamente.

Con esto logramos que el operador más, sume las variables como enteros y no como cadenas de caracteres. Si por ejemplo sumamos 1 + 1 sin utilizar la función parseInt el resultado será 11 en lugar de 2, ya que el operador + concatena las dos cadenas.

En JavaScript, como no podemos indicarle de qué tipo es la variable, requiere mucho más cuidado cuando operamos con sus contenidos.

Este problema es secuencial ya que ingresamos dos valores por teclado, luego hacemos dos operaciones y por último mostramos los resultados.

### Estructuras condicionales

Cuando se presenta la elección tenemos la opción de realizar una actividad u otra. Es decir tenemos actividades por el verdadero y por el falso de la condición. Lo más importante que hay que tener en cuenta es que se realizan las actividades de la rama del verdadero o las del falso, NUNCA se realizan las actividades de las dos ramas.

En una estructura condicional compuesta tenemos entradas, salidas, operaciones, tanto por la rama del verdadero como por la rama del falso.

Ejemplo:

Realizar un programa que lea dos números distintos y muestre el mayor de ellos: <html>

```
<head>
</head>
<body>
<script type="text/javascript"> var
num1,num2; num1=prompt('Ingrese el primer
número:'); num2=prompt('Ingrese el
segundo número:'); num1=parseInt(num1);
num2=parseInt(num2);
if (num1>num2)
{
  document.write('el mayor es '+num1);
} else {
  document.write('el mayor es '+num2);
}
</script>
</body>
</html>
```

Como se nota este sino no difiere en nada de java excepto **document.write('el mayor es '+num1);**

### Estructura anidada

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
var nota1,nota2,nota3;
  nota1=prompt('Ingrese 1ra. nota:', '');
  nota2=prompt('Ingrese 2da. nota:', '');
  nota3=prompt('Ingrese 3ra. nota:', '');
  //Convertimos los 3 string en enteros
  nota1=parseInt(nota1);
  nota2=parseInt(nota2);
  nota3=parseInt(nota3); var pro;
  pro=(nota1+nota2+nota3)/3;
  if (pro>=7) {
    document.write('promocionado');
  }else{ if
  (pro>=4) {
    document.write('regular');
  }else{
```

```

        document.write('reprobado');
    }
}
</script>
</body>
</html>

```

### Operadores && en las estructuras condicionales

El operador &&, traducido se lo lee como "Y". Se emplea cuando en una estructura condicional se disponen dos condiciones.

Cuando vinculamos dos o más condiciones con el operador "&&" las dos condiciones deben ser verdaderas para que el resultado de la condición compuesta de Verdadero y continúe por la rama del verdadero de la estructura condicional.

Recordemos que la condición debe ir entre paréntesis en forma obligatoria.

La utilización de operadores lógicos permiten en muchos casos, plantear algoritmos más cortos y comprensibles.

Veamos un ejemplo:

**Realizar un programa que lea por teclado tres números distintos y nos muestre el mayor de ellos.** <html>

```

<head>
</head>
<body>
<script type="text/javascript"> var
num1,num2,num3; num1=prompt('Ingrese
primer número:'); num2=prompt('Ingrese
segundo número:'); num3=prompt('Ingrese
tercer número:');
num1=parseInt(num1);
num2=parseInt(num2);
num3=parseInt(num3); if
(num1>num2 && num1>num3)
{
document.write('el mayor es el '+num1);
} else {
if (num2>num3)
{
document.write('el mayor es el '+num2);
} else {
document.write('el mayor es el '+num3);
}
}
}
</script>
</body>
</html>

```

## Operadores lógicos || (o) en las estructuras condicionales.

Traducido se lo lee como "O". Si la condición 1 es Verdadera o la condición 2 es Verdadera, luego ejecutar la rama del Verdadero.

Cuando vinculamos dos o más condiciones con el operador "O", con que una de las dos condiciones sea Verdadera alcanza para que el resultado de la condición compuesta sea Verdadero.

Ejemplo: Se carga una fecha (día, mes y año) por teclado. Mostrar un mensaje si corresponde al primer trimestre del año (enero, febrero o marzo).

Cargar por teclado el valor numérico del día, mes y año por separado.

```
<html>
<head>
</head>
<body>
<script type="text/javascript">
var dia,mes,año;
dia=prompt('Ingrese día:', '');
mes=prompt('Ingrese mes:', '');
año=prompt('Ingrese año:', '');
dia=parseInt(dia);
mes=parseInt(mes);
año=parseInt(año);
  if (mes==1 || mes==2 || mes==3)
  {
    document.write('corresponde al primer trimestre del año.');
```

## Estructura repetitiva for

Cualquier problema que requiera una estructura repetitiva se puede resolver empleando la estructura while.

Pero hay otra estructura repetitiva cuyo planteo es más sencillo en ciertas situaciones.

Esta estructura se emplea en aquellas situaciones en las cuales CONOCEMOS la cantidad de veces que queremos que se ejecute el bloque de instrucciones. Ejemplo: cargar 10 números, ingresar 5 notas de alumnos, etc. Conocemos de antemano la cantidad de veces que queremos que el bloque se repita.

Por último, hay que decir que la ejecución de la sentencia break dentro de cualquier parte del bucle provoca la salida inmediata del mismo.

Sintaxis:

```
for (<Inicialización>; <Condición>; <Incremento o Decremento>)
{
  <Instrucciones>
}
```

Esta estructura repetitiva tiene tres argumentos: variable de inicialización, condición y variable de incremento o decremento.

Funcionamiento:

- Primero se ejecuta por única vez el primer argumento .  
Por lo general se inicializa una variable.
- El segundo paso es evaluar la (Condición), en caso de ser verdadera se ejecuta el bloque, en caso contrario continúa el programa.
- El tercer paso es la ejecución de las instrucciones.

- El cuarto paso es ejecutar el tercer argumento (Incremento o Decremento). - Luego se repiten sucesivamente del Segundo al Cuarto Paso.

Este tipo de estructura repetitiva se utiliza generalmente cuando sabemos la cantidad de veces que deseamos que se repita el bloque.

Ejemplo: Cree un programa que permita generar una tabla de acuerdo al número de filas ingresa por teclado

Ir pidiendo datos para llenar cada columna(1), y en la columna(llevar la multiplicación del dato ingresado \* la variable controladora del ciclo), así: Si ingresa 3 Se ingresan 4,5,6.

Dato Ingresado	N	Dato Ingresado* N
4	0	0
5	1	5
6	2	12

```

<html>
<head>
</head>
<body>
<script type="text/javascript">
  var nf=0;
  var dato=0;
  var n=0;
  nf=prompt('Ingrese filas:');
  nf=parseInt(nf);

  document.write('<table border="1" bgcolor="#CEF6EC" >');
  document.write('<tr><td>Dato Ingresado</td><td>N</td><td>Dato Ingresado* N</td> </tr>');
  for(n=0;n<nf;n++)
  {
    dato=prompt('Ingrese dato:');
    document.write('<tr>');
    document.write('<td>',dato,' </td>');
    document.write('<td>',n,'</td>');
    document.write('<td>',dato*n,'</td>');
    document.write('</tr>');
  }
  document.write('</table>');
</script>
</body>
</html>

```

**Nota:** Desde JavaScript como hemos visto se pueden generar etiquetas personalizadas de html de acuerdo a las circunstancias ó programa a implementar, aunque estas funcionalidades son implementadas de manera más sofisticadas desde JQUERY, AJAX, JASON, DOM , pero esta es su esencia.



### Ciclo while

Este ciclo al igual que un lenguaje de programación, el programa se ejecute n veces siempre y cuando cumpla con la condición

```
<html><head></head>
<body>
<script type="text/javascript">
var x; x=1; while (x<=100)
{
  document.write(x);
document.write('<br>');
  x=x+1;
}
</script>
</body>
</html>
```

### Ciclo do while

Este ciclo al igual que un lenguaje de programación, garantiza que el programa se ejecute una sola vez

```
<html>
<head></head>
<body>
<script type="text/javascript">
  var valor;
do {
  valor=prompt('Ingrese un valor entre 0 y 999:');
valor=parseInt(valor);
  document.write('El valor '+valor+' tiene ');
if (valor<10)
  {
  document.write('Tiene 1 digitos');
  }
  else
  {
  if (valor<100)
  {
  document.write('Tiene 2 digitos');
  }
  else
  {
  document.write('Tiene 3 digitos');
  }
  }
  document.write('<br>');
```

```
    } while(valor!=0);  
</script>  
</body>  
</html>
```

## Estructura switch

La instrucción switch es una alternativa para reemplazar los if/else if.

De todos modos se puede aplicar en ciertas situaciones donde la condición se verifica si es igual a cierto valor. No podemos preguntar por mayor o menor.

Con un ejemplo sencillo veremos cuál es su sintaxis. Confeccionar un programa que solicite que ingrese un valor entre 1 y 5. Luego mostrar en castellano el valor ingresado. Mostrar un mensaje de error en caso de haber ingresado un valor que no se encuentre en dicho rango.

```
<html>  
<head>  
</head>  
<body>  
<script type="text/javascript">  
    var valor;  
    valor=prompt("Ingrese un valor comprendido entre 1 y 5:");  
    //Convertimos a entero  
    valor=parseInt(valor); switch  
    (valor) {  
        case 1: document.write('uno');  
            break;  
        case 2: document.write('dos');  
            break;  
        case 3: document.write('tres');  
            break;  
        case 4: document.write('cuatro');  
            break;  
        case 5: document.write('cinco');  
            break;  
        default:document.write('debe ingresar un valor comprendido entre 1 y 5.');  
    }  
</script>  
</body>  
</html>
```

Segundo problema para ver que podemos utilizar variables de tipo cadena con la instrucción switch.

Ingresar por teclado el nombre de un color (rojo, verde o azul), luego pintar el fondo de la página

```
<html><head></head>  
<body><script type="text/javascript">  
var col;
```

```

col=prompt('Ingrese el color con que se quiere pintar el fondo de la ventana (rojo, verde, azul)');
switch (col) {
  case 'rojo': document.bgColor='#ff0000';
break;
  case 'verde': document.bgColor='#00ff00';
break;
  case 'azul': document.bgColor='#0000ff';

      break;
}
</script></body></html>

```

**Actividad evaluativa:**

Realice un programa en javascript que permita generar en una página html con las operaciones básicas de multiplicar, sumar, dividir, restar como lo muestra la siguiente gráfico Un dato ingresado por pantalla por el consecutivo del ciclo como lo muestra el gráfico

Dato Ingresado	consecutivo	*	+	-	/	dato > consecutivo
4	1	4	5	3	4	Sí
1	2	2	3	-1	0.5	No
6	3	18	9	3	2	Sí

Para este caso :

$4 * 1 = 4.$

$4 + 1 = 5. 4 - 1$

$= 3.$

$4 / 4 = 1.$

4 es mayor que 1 = Sí

**Bibliografía:**

<http://html-color-codes.info/codigos-de-colores-hexadecimales> <http://www.javascriptya.com.ar/>

<http://www.w3schools.com/>