	UNIVERSIDAD DON BOSCO **FACULTAD DE INGENERIA ESCUELA DE COMPUTACION**
CICLO 02-2019	**GUIA DE LABORATORIO Nº 6** **Nombre de la practica:** Introducción y conceptos básicos de JavaScript **Lugar de ejecución:** Laboratorio de Informática **Tiempo estimado:** 2 horas **Materia:** Lenguajes interpretados en el cliente

I. Objetivos

Que el estudiante sea capaz de:

1. Tener claridad en el proceso a seguir para implementar JavaScript en las páginas web.
2. Identificar la estructura básica JavaScript.

II. Introducción Teórica

¿Qué es JavaScript?

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 C, C++ 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. Hasta hace poco no se podía desarrollar programas con JavaScript que se ejecutaran 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.

Nuestro primer programa será el famoso "Hola Mundo", es decir un programa que muestre en el documento HTML el mensaje "Hola Mundo".

```html
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  document.write('Hola Mundo');
</script>

</body>
</html>
```

El programa en JavaScript debe ir encerrado entre las marcar 'script':

<script>

</script>

En versiones anteriores a HTML5 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. Ya veremos que los nombres de funciones llevan letras en mayúsculas.

Ejemplo #01

Realizar un programa que muestre su nombre y su edad en una página HTML. Emplear el comando write del objeto document para imprimir. Tener en cuenta que si queremos que cada dato quede en una fila distinta de la página debemos insertar la etiqueta HTML
 (salto de linea en HTML), es decir debemos disponer: document.write('
')

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  document.write('Diego Martinez');
  document.write('<br>');
  document.write('44');
</script>

</body>
</html>
```

Variables.

Una variable es un depósito donde hay un valor. Consta de un nombre y pertenece a un tipo (numérico, cadena de caracteres, etc.)

Tipos de variable:

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 que veremos más adelante.

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, anio;

a una variable se la puede definir e inmediatamente inicializarla con un valor:

var edad=20;

o en su defecto en dos pasos:

var edad;

edad=20;

Elección del nombre de una variable:

Debemos elegir nombres de variables representativos. En el ejemplo anterior los nombres dia, mes, anio son lo suficientemente claros para darnos una idea acabada sobre su contenido, una mala elección de nombres hubiera sido llamarlas a, b y c. Podemos darle otros buenos nombres. Otros no son tan representativos, por ejemplo d, m, a. Posiblemente cuando estemos resolviendo un problema dicho nombre nos recuerde que almacenamos el dia, pero pasado un tiempo lo olvidaríamos.

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.

En el siguiente ejemplo definimos una serie de variables y las mostramos en la página:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>
<script>
  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.

Recordemos que el lenguaje JavaScript es sensible a mayúsculas y minúsculas y no será lo mismo si tipeamos:

```
Document.Write(nombre);
```

Esto porque no existe el objeto 'Document' sino el objeto 'document' (con d minúscula), lo mismo no existe la función 'Write' sino 'write', este es un error muy común cuando comenzamos a programar en JavaScript

III. MATERIALES Y EQUIPO

Para la realización de la guía de práctica se requerirá lo siguiente:

No.	Requerimiento	Cantidad
1	Guía de práctica #06: Introducción y conceptos básicos de JavaScript	1
2	Computadora con navegadores y editor de texto instalado	1
3	Memoria USB o disco flexible	1

IV. PROCEDIMIENTO

Indicaciones: Para todos los ejemplos de esta guía de práctica utilizaremos el Notepad++ o Sublime Text 3. Se pide que cuando guarde el archivo lo haga con extensión .html, En todo caso, la única extensión diferente que se utilizará en esta práctica será la extensión .css, para cuando se requiera crear un script con los estilos de manera independiente.

Ejemplo # 02

Confeccionar un programa en JavaScript que defina e inicialice una variable de tipo cadena de caracteres donde almacenemos el nombre de un empleado y otra variable de tipo entera donde almacenar el sueldo. Imprimir cada variable en una línea distinta en pantalla.

Solución :

```html
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var nombre='Juan';
  var sueldo=6500;
  document.write(nombre);
  document.write('<br>');
  document.write(sueldo);
</script>

</body>
</html>
```

Entrada de datos por teclado.

Para la entrada de datos por teclado tenemos la función **prompt**. Cada vez que necesitamos ingresar un dato con ésta 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.

Para ver su funcionamiento analicemos este ejemplo:

```html
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  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>);

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

Ejemplo # 03

Confeccionar un programa que permita cargar el nombre de un usuario y su mail por teclado. Mostrar posteriormente los datos en la página HTML.

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var usuario;
  var mail;
  usuario=prompt('Ingrese el nombre de usuario:','');
  mail=prompt('Ingrese el mail:','');
  document.write('Nombre de usuario ingresado:');
  document.write(usuario);
  document.write('<br>');
  document.write('Mail ingresado:');
  document.write(mail);
</script>

</body>
</html>
```

Estructuras secuenciales de programación.

Cuando en un problema sólo participan operaciones, entradas y salidas se la denomina estructura secuencial.

El problema anterior, donde se ingresa el nombre de una persona y su edad se trata de una estructura secuencial.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var valor1;
  var valor2;
  valor1=prompt('Ingrese primer número:','');
  valor2=prompt('Ingrese segundo número','');
  var suma=parseInt(valor1)+parseInt(valor2);
  var producto=valor1*valor2;
  document.write('La suma es ');
  document.write(suma);
  document.write('<br>');
  document.write('El producto es ');
  document.write(producto);
</script>

</body>
</html>
```

Ejemplo de otro algoritmo con estructura secuencial: Realizar la carga de dos números por teclado e imprimir su suma y su producto:

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** y pasar como parámetro las variables valor1 y valor2 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.

Ejemplo # 04

Realizar la carga del lado de un cuadrado, mostrar por pantalla el perímetro del mismo (El perímetro de un cuadrado se calcula multiplicando el valor del lado por cuatro)

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var lado;
  lado=prompt('Ingrese medida del lado:','');
  var perimetro;
  perimetro=lado*4;
  document.write('Su perímetro es:');
  document.write(perimetro);
</script>

</body>
</html>
```

Ejemplo # 05

Escribir un programa en el cual se ingresen cuatro números, calcular e informar la suma de los dos primeros y el producto del tercero y el cuarto.

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var num1;
  var num2;
  var num3;
  var num4;
  num1=prompt('Ingrese primer valor:','');
  num2=prompt('Ingrese segundo valor:','');
  num3=prompt('Ingrese tercer valor:','');
  num4=prompt('Ingrese cuarto valor:','');
  var suma;
  suma=parseInt(num1)+parseInt(num2);
  var producto;
  producto=parseInt(num3)*parseInt(num4);
  document.write('La suma de los dos primeros valores es:');
  document.write(suma);
  document.write('<br>');
  document.write('El producto del tercer y cuarto valor es:');
  document.write(producto);
</script>

</body>
</html>
```

Ejemplo # 07

Se debe desarrollar un programa que pida el ingreso del precio de un artículo y la cantidad que lleva el cliente. Mostrar lo que debe abonar el comprador (Ingresar por teclado un precio sin decimales, es decir un entero: 2, 7, 90 etc.)

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var precio;
  var cantidad;
  precio=prompt('Ingrese precio del artículo','');
  cantidad=prompt('Ingrese la cantidad de artículos a llevar:','');
  var importe;
  importe=parseInt(precio)*parseInt(cantidad);
  document.write('Debe abonar:');
  document.write(importe);
</script>

</body>
</html>
```

Estructuras condicionales simples.

No todos los problemas pueden resolverse empleando estructuras secuenciales. Cuando hay que tomar una decisión aparecen las estructuras condicionales.

En nuestra vida diaria se nos presentan situaciones donde debemos decidir.

¿Elijo la carrera A o la carrera B ?
¿Me pongo este pantalón ?
¿Entro al sitio A o al sitio B ?
Para ir al trabajo, ¿elijo el camino A o el camino B ?
Al cursar una carrera, ¿elijo el turno mañana, tarde o noche ?

Por supuesto que en un problema se combinan estructuras secuenciales y condicionales.

Cuando se presenta la elección tenemos la opción de realizar una actividad o no realizarla.

En una estructura CONDICIONAL SIMPLE por el camino del verdadero hay actividades y por el camino del falso no hay actividades. Por el camino del verdadero pueden existir varias operaciones, entradas y salidas, inclusive ya veremos que puede haber otras estructuras condicionales.

Ejemplo # 08

Realizar la carga de una nota de un alumno. Mostrar un mensaje que aprobó si tiene una nota mayor o igual a 4:

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var nombre;
  var nota;
  nombre=prompt('Ingrese nombre:','');
  nota=parseInt(prompt('Ingrese su nota:',''));
  if (nota>=4)
  {
    document.write(nombre+' esta aprobado con un '+nota);
  }
</script>

</body>
</html>
```

Aparece la instrucción **if** en el lenguaje JavaScript. La condición debe ir entre paréntesis. Si la condición se verifica verdadera se ejecuta todas las instrucciones que se encuentran encerradas entre las llaves de apertura y cerrado seguidas al if.

Para disponer condiciones en un if podemos utilizar alguno de los siguientes operadores relacionales:

> mayor
>= mayor o igual
< menor
<= menor o igual
!= distinto
== igual

Siempre debemos tener en cuenta que en la condición del if deben intervenir una variable un operador relacional y otra variable o valor fijo.

Como queremos que en la variable 'nota' se guarde como entero lo convertimos llamando a **parseInt**:

```
nota=parseInt(prompt('Ingrese su nota:',''));
```

Otra cosa que hemos incorporado es el operador **+** para cadenas de caracteres:

```
document.write(nombre+' esta aprobado con un '+nota);
```

Con esto hacemos más corto la cantidad de líneas de nuestro programa, recordemos que veníamos haciéndolo de la siguiente forma:

```
document.write(nombre);
document.write(' esta aprobado con un ');
document.write(nota);
```

Ejemplo # 09

Se ingresan tres notas de un alumno, si el promedio es mayor o igual a siete mostrar el mensaje 'Promocionado'. Tener en cuenta que para obtener el promedio debemos operar suma=nota1+nota2+nota3; y luego hacer promedio=suma/3; Cuando cargamos una nota y queremos convertir inmediatamente el valor

ingresado a entero podemos hacer:

```
nota1=prompt('Ingrese primer nota:','');
nota1=parseInt(nota1);
```

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>
<script>
  var nota1;
  var nota2;
  var nota3;
  nota1=prompt('Ingrese primer nota','');
  nota1=parseInt(nota1);
  nota2=prompt('Ingrese segunda nota','');
  nota2=parseInt(nota2);
  nota3=prompt('Ingrese tercer nota','');
  nota3=parseInt(nota3);
  var suma;
  suma=nota1+nota2+nota3;
  var promedio;
  promedio=suma/3;
  if (promedio>=7)
  {
    document.write('Promocionado');
  }
</script>
</body>
</html>
```

Ejemplo # 10

Solicitar que se ingrese dos veces una clave. Mostrar un mensaje si son iguales (tener en cuenta que para ver si dos variables tienen el mismo valor almacenado debemos utilizar el operador ==)

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var clave1;
  var clave2;
  clave1=prompt('Ingrese una clave:','');
  clave2=prompt('Repita el ingreso de la clave:','');
  if (clave1==clave2)
  {
    document.write('Ingresó las dos claves iguales');
  }
</script>

</body>
</html>
```

Estructuras condicionales compuestas.

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 # 11

Realizar un programa que solicite dos números distintos y muestre el mayor de ellos:

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  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>
```

La función prompt retorna un string por lo que debemos convertirlo a entero cuando queremos saber cual de los dos valores es mayor numéricamente. En el lenguaje JavaScript una variable puede ir cambiando el tipo de dato que almacena a lo largo de la ejecución del programa.

Más adelante veremos qué sucede cuando preguntamos cuál de dos string es mayor.

Estamos en presencia de una ESTRUCTURA CONDICIONAL COMPUESTA ya que tenemos actividades por la rama del verdadero y del falso.
La estructura condicional compuesta tiene la siguiente codificación:

```
if (<condición>)
{
  <Instruccion(es)>
}
else
{
  <Instruccion(es)>
}
```

Es igual que la estructura condicional simple salvo que aparece la palabra clave “else” y posteriormente un bloque { } con una o varias instrucciones.

Si la condición del if es verdadera se ejecuta el bloque que aparece después de la condición, en caso que la condición resulte falsa se ejecuta la instrucción o bloque de instrucciones que indicamos después del else.

Ejemplo # 12

Realizar un programa que lea por teclado dos números, si el primero es mayor al segundo informar su suma y diferencia, en caso contrario informar el producto y la división del primero respecto al segundo.

Solución:

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>
<script>
  var num1,num2;
  num1=prompt('Ingrese el primer número:','');
  num2=prompt('Ingrese el segundo número:','');
  num1=parseInt(num1);
  num2=parseInt(num2);
  if (num1>num2)
  {
    var suma,diferencia;
    suma=num1+num2;
    diferencia=num1-num2;
    document.write('La suma es:'+suma);
    document.write('<br>');
    document.write('La diferencia es:'+diferencia);
  }
  else
  {
    var producto,division;
    producto=num1*num2;
    division=num1/num2;
    document.write('El producto es '+producto);
    document.write('<br>');
    document.write('La división del primero respecto al segundo es:'+division);
  }
</script>
</body>
</html>
```

Estructuras condicionales anidadas.

Decimos que una estructura condicional es anidada cuando por la rama del verdadero o el falso de una estructura condicional hay otra estructura condicional.

Ejemplo: Confeccionar un programa que pida por teclado tres notas de un alumno, calcule el promedio e imprima alguno de estos mensajes:

Si el promedio es >=7 mostrar "Promocionado".
Si el promedio es >=4 y <7 mostrar "Regular".
Si el promedio es <4 mostrar "Reprobado".

Ejemplo # 13

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>
<script>
  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>
```

Analicemos el siguiente programa. Se ingresan tres string por teclado que representan las notas de un alumno, se transformas a variables enteras y se obtiene el promedio sumando los tres valores y dividiendo por 3 dicho resultado.

Primeramente, preguntamos si el promedio es superior o igual a 7, en caso afirmativo por la rama del verdadero de la estructura condicional mostramos un mensaje que indique 'Promocionado' (con comillas indicamos un texto que debe imprimirse en pantalla).

En caso que la condición nos de falso, por la rama del falso aparece otra estructura condicional, porque todavía debemos averiguar si el promedio del alumno es superior o igual a cuatro o inferior a cuatro.

Los comentarios en JavaScript los hacemos disponiendo dos barras previas al comentario (los comentarios en tiempo de ejecución no son tenidos en cuenta y tienen por objetivos de documentar el programa para futuras modificaciones):

```
//Convertimos los 3 string en enteros
```

Si queremos disponer varias líneas de comentarios tenemos como alternativa:

```
/*
linea de comentario 1.
linea de comentario 2.
etc.
*/
```

Es decir, encerramos el bloque con los caracteres /* */

Operadores lógicos && (y) 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.

Las utilizaciones de operadores lógicos permiten en muchos casos, plantear algoritmos más cortos y comprensibles.

Veamos un ejemplo: Confeccionar un programa que lea por teclado tres números distintos y nos muestre el mayor de ellos.

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  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>
```

Podemos leerla de la siguiente forma:

Si el contenido de la variable num1 es mayor al contenido de la variable num2 Y si el contenido de la variable num1 es mayor al contenido de la variable num3 entonces la CONDICION COMPUESTA resulta Verdadera.

Si una de las condiciones simples da falsa, la CONDICION COMPUESTA da Falsa y continúa por la rama del falso.

Es decir que se mostrará el contenido de num1 si y sólo si num1>num2 y num1>num3.

En caso de ser Falsa la condición analizamos el contenido de num2 y num3 para ver cual tiene un valor mayor.

En esta segunda estructura condicional, al haber una condición simple, no se requieren operadores lógicos.

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.

Ejemplo # 14

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  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.');
  }
</script>

</body>
</html>
```

La carga de una fecha se hace por partes, ingresamos las variables dia, mes y año.

Si alguna de las condiciones simples del if da verdadero luego se muestra el mensaje:

```
if (mes==1 || mes==2 || mes==3)

{
  document.write('corresponde al primer trimestre del año.');
}
```

Estructuras switch.

La instrucción switch es una alternativa para remplazar en algunas situaciones 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.

Ejemplo # 15

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  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>
```

Debemos tener en cuenta que la variable que analizamos debe ir después de la instrucción switch entre paréntesis. Cada valor que se analiza debe ir luego de la palabra clave 'case' y seguido a los dos puntos, las instrucciones a ejecutar, en caso de verificar dicho valor la variable que analiza el switch.

Es importante disponer la palabra clave 'break' al finalizar cada caso. La instrucciones que hay después de la palabra clave 'default' se ejecutan en caso que la variable no se verifique en algún case. De todos modos el default es opcional en esta instrucción.

Plantearemos un 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 mostraremos un mensaje indicando el color ingresado:

Estructura repetitiva (while)

Hasta ahora hemos empleado estructuras SECUENCIALES y CONDICIONALES. Existe otro tipo de estructuras tan importantes como las anteriores que son las estructuras REPETITIVAS.

Una estructura repetitiva permite ejecutar una instrucción o un conjunto de instrucciones varias veces.

Una ejecución repetitiva de sentencias se caracteriza por:

- La o las sentencias que se repiten.

- El test o prueba de condición antes de cada repetición, que motivará que se repitan o no las sentencias.

Funcionamiento del while: En primer lugar se verifica la condición, si la misma resulta verdadera se ejecutan las operaciones que indicamos entre las llaves que le siguen al while.
En caso que la condición sea Falsa continúa con la instrucción siguiente al bloque de llaves.

El bloque se repite MIENTRAS la condición sea Verdadera.

Importante: Si la condición siempre retorna verdadero estamos en presencia de un ciclo repetitivo infinito. Dicha situación es un error de programación, nunca finalizará el programa.

Ejemplo: Realizar un programa que imprima en pantalla los números del 1 al 100.

Sin conocer las estructuras repetitivas podemos resolver el problema empleando una estructura secuencial. Inicializamos una variable con el valor 1, luego imprimimos la variable, incrementamos nuevamente la variable y así sucesivamente. Pero esta solución es muy larga.

La mejor forma de resolver este problema es emplear una estructura repetitiva:

Ejemplo # 16

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var x;
  x=1;
  while (x<=100)
  {
    document.write(x);
    document.write('<br>');
    x=x+1;
  }
</script>

</body>
</html>
```

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.

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

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

Ejemplo: Mostrar por pantalla los números del 1 al 10.

Ejemplo # 17

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo de JavaScript</title>
  <meta charset="UTF-8">
</head>
<body>

<script>
  var f;
  for(f=1;f<=10;f++)
  {
    document.write(f+" ");
  }
</script>

</body>
</html>
```

Análisis de resultados

1. Realice una página web que cargue por teclado tres números distintos. Mostrar por pantalla el mayor y menor de ellos, no debe de aceptar números negativos ni números iguales.

2. Realice una página web que permita cargar un número entero positivo de hasta tres cifras y muestre un mensaje indicando si tiene 1, 2, o 3 cifras.

3. Realice una página web que lea por teclado dos números, si el primero es mayor al segundo informar su suma y diferencia, en caso contrario informar el producto y la división del primero respecto al segundo.

VI. Ejercicios complementarios

1. Realice una página web que solicite el ingreso alguna de estas palabras (casa, mesa, perro, gato) luego mostrar la palabra traducida en inglés. Es decir, si se ingresa 'casa' debemos mostrar el texto 'house' en la página.

2. Realice una página web que lea los lados de 4 triángulos, e informar:
   a) De cada uno de ellos, qué tipo de triángulo es: equilátero (tres lados iguales), isósceles (dos lados iguales), o escaleno (ningún lado igual)
   b) Cantidad de triángulos de cada tipo.
   c) Tipo de triángulo del que hay menor cantidad.

Fecha de entrega: Próxima semana