

JavaScript

M.C. Rafael A. García Rosas.

rafaelgr@umad.edu.mx

¿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.
- **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.**
- No es un lenguaje de programación propiamente dicho como 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.

Hola mundo ;)

```
<html>
<head>
</head>
<body>
  <script language="javascript">
    document.write('Hola Mundo');
  </script>
</body>
</html>
```

Atención -> JavaScript es SENSIBLE A MAYUSCULAS Y MINUSCULAS.

Variables.

Fundamentalmente una variable puede almacenar:

- Valores Enteros (100, 260, etc.)
- Valores Reales (1.24, 2.90, 5.00, etc.)
- Cadenas de caracteres ("Juan", "Compras", "Listado", etc.)
- Valores lógicos (true,false)

```
<body>  
  <script language="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>
```

Entrada 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.

```
<body>
  <script language="JavaScript">
    var nombre;
    var edad;
    nombre=prompt('Ingrese su nombre:', '');
    edad=prompt('Ingrese su edad:', '');           //mensaje y valor inicial a mostrar
    document.write('Hola ');
    document.write(nombre);
    document.write(' así que tienes ');
    document.write(edad);
    document.write(' años');
  </script>
</body>
```


Estructuras secuenciales de programación.

- Cuando en un problema sólo participan operaciones, entradas y salidas se la denomina **estructura secuencial**.
- Ej. Realizar la carga de dos números por teclado e imprimir su suma y su producto:

```
<html>
<head>
<script language="JavaScript">
var valor1;
var valor2;
valor1=prompt('Ingrese primer
número:', '');
valor2=prompt('Ingrese segundo
número:', '');
var
suma=parseInt(valor1)+parseInt(val
or2);
```

```
var producto=valor1*valor2;
document.write('La suma es ');
document.write(suma);
document.write('<br>');
document.write('El producto es ');
document.write(producto);
</script>
</head>
<body>
</body>
</html>
```

Estructuras condicionales if, if-else.

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

```
<html>
<head>
</head>
<body>
<script language="javascript">
var nombre;
var nota;
nombre=prompt('Ingresa nombre:','');
nota=prompt('Ingresa su nota:','');
if (nota>=4) //posibles >, >=, <, <=, !=, ==
{
  document.write(nombre+' esta aprobado con un '+nota);
}
</script>
</body>
</html>
```

Operadores lógicos && (and) y || (or) en las estructuras condicionales.

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

```
<body>
<script language="javascript">
var num1,num2,num3;
num1=prompt('Ingresa primer
número:','');
num2=prompt('Ingresa segundo
número:','');
num3=prompt('Ingresa 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>
```


Otras estructuras

- También es posible usar:
 - Switch
 - While
 - do/while
 - For
- **Propón** un escenario donde sea útil usar una de estas estructuras.

Funciones.

- Incluso los programas más sencillos tienen la necesidad de fragmentarse.
- Las funciones son los únicos tipos de subprogramas que acepta JavaScript.
- Tienen la siguiente estructura:

```
function <nombre de función>(argumento1, argumento2, ..., argumento n)
{
  <código de la función>
}
```

Funciones.

```
<body>
<script lenguaje="javascript">
function mostrarMensaje()
{
    document.write("Cuidado<br>");
    document.write("Ingrese su documento
correctamente<br>");
}
```

```
mostrarMensaje();
mostrarMensaje();
mostrarMensaje();
</script>
</body>
```

Funciones con parámetros.

```
<body>
<script language="javascript">
function mostrarComprendidos(x1,x2)
{
  var inicio;
  for(inicio=x1;inicio<=x2;inicio++)
  {
    document.write(inicio+' ');
  }
}

var valor1,valor2;
valor1=prompt('Ingrese valor inferior:','');
valor1=parseInt(valor1);
valor2=prompt('Ingrese valor superior:','');
valor2=parseInt(valor2);
mostrarComprendidos(valor1,valor2);
</script>
</body>
```

Funciones que retornan un valor.

Confeccionar una función que reciba un valor entero comprendido entre 1 y 5. Luego retornar en castellano el valor recibido.

```
<body>
<script language="javascript">
function convertirCastellano(x)
{
  switch (x)
  {
    case 1:return "uno";
    case 2:return "dos";
    case 3:return "tres";
    case 4:return "cuatro";
    case 5:return "cinco";
    default:return "valor incorrecto";
  }
}
```

```
var valor;
valor=prompt("Ingrese un valor entre 1 y 5","");
valor=parseInt(valor);
var r;

r=convertirCastellano(valor);

document.write(r);
</script>
</body>
```


Tarea 1

Problema 1

Elaborar una función a la cual le enviemos tres enteros y muestre el menor.

Problema 2

Confeccionar una función a la cual le envíe tres enteros y los muestre ordenados de menor a mayor.

Problema 3

Elaborar una función a la cual le envíe el valor del lado de un cuadrado y me retorne su perímetro.

Tarea 1 (2)

Problema 4

Desarrollar una función que retorne la cantidad de dígitos que tiene una variable entera positiva.

Problema 5

Elaborar una función que reciba tres enteros y retorne el promedio.

Problema 6

Confeccionar una función que solicite la carga de 5 valores por teclado y retorne su suma.

➤ Para todos los casos

Implementa un **formulario html** por cada función para probar su funcionamiento.

Programación orientada a objetos.

- Una clase es una agrupación de objetos que comparten las mismas propiedades y comportamientos.
- La clase es un molde para objetos que poseen las mismas características (que pueden recibir los mismos mensajes y responden de la misma manera).
- Un objeto es una estructura que contiene tanto las variables (llamadas propiedades) como las funciones que manipulan dichas variables (llamadas métodos).
- Para acceder a los métodos y propiedades de un objeto debemos utilizar la siguiente sintaxis:
objeto.propiedad objeto.metodo(parámetros)

Clase Date

- JavaScript dispone de varias clases predefinidas.

```
fecha = new Date();           //creación de un objeto de la clase Date  
fecha = new Date(año, mes, dia);  
fecha = new Date(año, mes, dia, hora, minuto, segundo);
```

- Si no utilizamos parámetros, el objeto fecha contendrá la fecha y hora actuales, obtenidas del reloj de nuestra computadora.
- En caso contrario hay que tener en cuenta que los meses comienzan por cero. Así, por ejemplo:

```
navidad06 = new Date(2006, 11, 25)
```


Clase Date

Ejemplo: Mostrar en una página la fecha y la hora actual.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function mostrarFechaHora()
{
    var fecha
    fecha=new Date();
    document.write('Hoy es ');
    document.write(fecha.getDate()+'/');

    document.write((fecha.getMonth()+1)
+'/');
```

```
document.write(fecha.getYear());
    document.write('<br>');
    document.write('Es la hora ');
    document.write(fecha.getHours()+' ');
    document.write(fecha.getMinutes()+' ');
    document.write(fecha.getSeconds());
}
//Llamada a la función
mostrarFechaHora();
</SCRIPT>
</HEAD>
<BODY>
```


Clase Array

- Un vector es una estructura de datos que permite almacenar un conjunto de datos.
- Con un único nombre se define un vector y por medio de un subíndice hacemos referencia a cada elemento del mismo (componente).

sueldos=new Array(5);

Clase Array

Crear un vector para almacenar los **cinco sueldos** de operarios y luego mostrar el total de gastos en sueldos (cada actividad en una función independiente).

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
function cargar(sueldos)
{
  var f;
  for(f=0;f<sueldos.length;f++)
  {
    var v;
    v=prompt('Ingrese sueldo:', '');
    sueldos[f]=parseInt(v);
  }
}
```

Clase Array (2)

```
function calcularGastos(sueldos)
{
    var total=0;
    var f;
    for(f=0;f<sueldos.length;f++)
    {
        total=total+sueldos[f];
    }
    document.write('Listado de
sueldos<br>');
    for(f=0;f<sueldos.length;f++)
    {
        document.write(sueldos[f]+'<br>');
    }
}
```

```
document.write('Total de gastos en
sueldos:'+total);
}

var sueldos;
sueldos=new Array(5);
cargar(sueldos);
calcularGastos(sueldos);

</SCRIPT>
</BODY>
</HTML>
```

Clase Math

Esta clase no está construido para que tengamos nuestras variables Math, sino como un contenedor que tiene diversas constantes (como Math.E y Math.PI) y los siguientes métodos matemáticos.

Método	Descripción	Expresión de ejemplo	Resultado del ejemplo
abs	Valor absoluto	Math.abs(-2)	2
sin, cos, tan	Funciones trigonométricas, reciben el argumento en radianes	Math.cos(Math.PI)	-1
asin, acos, atan	Funciones trigonométricas inversas	Math.asin(1)	1.57
exp, log	Exponenciación y logaritmo, base E	Math.log(Math.E)	1
ceil	Devuelve el entero más pequeño mayor o igual al argumento	Math.ceil(-2.7)	-2

Clase Math (2)

Método	Descripción	Expresión de ejemplo	Resultado del ejemplo
floor	Devuelve el entero más grande menor o igual al argumento	Math.floor(-2.7)	-3
round	Devuelve el entero más cercano o igual al argumento	Math.round(-2.7)	-3
min, max	Devuelve el menor (o mayor) de sus dos argumentos	Math.min(2,4)	2
pow	Exponenciación, siendo el primer argumento la base y el segundo el exponente	Math.pow(2,3)	8
sqrt	Raíz cuadrada	Math.sqrt(25)	5
random	Genera un valor aleatorio comprendido entre 0 y 1.	Math.random()	Ej. 0.7345

Clase Math (3)

Ej: Programa que permite cargar un valor comprendido entre 1 y 10. Luego generar un valor aleatorio entre 1 y 10, mostrar un mensaje con el número sorteado e indicar si ganó o perdió:

```
<body>
<script language="JavaScript">
  var selec=prompt('Ingrese un valor entre 1 y 10,');
  selec=parseInt(selec);

  var num=parseInt(Math.random()*10)+1;
  if (num==selec)
    document.write('Ganó el número que se sorteó es el '+ num);
  else
    document.write('Lo siento se sorteó el valor '+num+' y usted eligió el '+selec);
</script>
</body>
```

Formularios y Eventos.

- El uso de JavaScript en los formularios HTML se hace fundamentalmente con el objetivo de **validar los datos ingresados**.
- Se hace esta actividad en el cliente (navegador) para desligar de esta actividad al servidor que recibirá los datos ingresados por el usuario.
- ***El navegador crea un objeto por cada control visual que aparece dentro de la página. Nosotros podemos acceder posteriormente desde JavaScript a dichos objetos.***
- El objeto principal es el FORM que contendrá todos los otros objetos: TEXT (editor de líneas), TEXTAREA (editor de varias líneas), etc.
- Nuestra actividad en JavaScript es procesar los eventos que generan estos controles (**un evento es una acción que se dispara, por ejemplo si se presiona un botón**).

Formularios y Eventos.

Ej. dispondremos un botón y cada vez que se presione, mostraremos un contador:

```
<body>
  <script language="JavaScript">
    var contador=0;
    function incrementar()
    {
      contador++;
      alert('El contador ahora vale : ' + contador);
    }
  </script>

  <form>
    <input type="button" onClick="incrementar()" value="incrementar">
  </form>
</body>
```

Controles FORM, BUTTON y TEXT.

Muestra el nombre y edad de una persona:

```
<body>

<script language="JavaScript">
function mostrar()
{
  var
nom=document.form1.nombre.value;
  var ed=document.form1.edad.value;
  alert('Ingreso el nombre:' + nom);
  alert('Y la edad:' + ed);
}
</script>
```

```
<form name="form1">
  Ingrese su nombre:
  <input type="text"
name="nombre"><br>
  Ingrese su edad:
  <input type="text"
name="edad"><br>
  <input type="button"
value="Confirmar"
onClick="mostrar()">
</form>

</body>
```


Tarea 2

Problema 1

Desarrollar un programa que permita ingresar un vector de 8 elementos, e informe:

- El valor acumulado de todos los elementos del vector.
- El valor acumulado de los elementos del vector que sean mayores a 36.
- Cantidad de valores mayores a 50.

Problema 2

Realizar un programa que pida la carga de dos vectores numéricos.

- Obtener la suma de los dos vectores, dicho resultado guardarlo en un tercer vector del mismo tamaño.
- Sumar componente a componente.
- El tamaño del vector es a elección.

➤ Para todos los casos

Implementa una **página html** por cada función para probar su funcionamiento.

Tarea 2 (2)

Problema 3

Crear un programa que permita cargar un entero en un text y al presionar un botón nos muestre dicho valor elevado al cubo (emplear la función *alert*).

Problema 4

Cargar dos números en objetos de tipo text y al presionar un botón, mostrar el mayor.

Problema 5

Cargar un nombre y un apellido en campos text. Al presionar un botón, concatenarlos y mostrarlos en un tercer text (Tener en cuenta que podemos modificar la propiedad value de un objeto TEXT cuando ocurre un evento).

Control PASSWORD.

La mayoría de las veces este dato se procesa en el servidor. Pero podemos en el cliente (es decir en el navegador) verificar si ha ingresado una cantidad correcta de caracteres.

```
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
function validar()
{
  if
  (document.form1.text1.value.length<5)
  {
    alert("Ingrese al menos 5 caracteres");
    document.form1.text1.value="";
  }
}
</SCRIPT>
</HEAD>
```

```
<BODY>
<FORM NAME="form1">
  Ingrese clave(al menos 5 caracteres): <INPUT
TYPE="password" NAME="text1">
  <INPUT TYPE="button" ONCLICK="validar()"
  VALUE="Enviar">
</FORM>
</BODY>
```

Control SELECT.

El objetivo fundamental en JavaScript es determinar qué elemento está seleccionado y qué valor tiene asociado, para dinámicamente hacer algo con él. Esto lo hacemos cuando ocurre el evento **OnChange**.

```
<html><head></head>
<body>
<script language="JavaScript">
function cambiarColor()
{
    document.form1.text1.value =
document.form1.select1.selectedIndex;
    document.form1.text2.value =

document.form1.select1.options[document
form1.select1.selectedIndex].text;
    document.form1.text3.value =
        document.form1.select1.options
[document.form1.select1.selectedIndex].v
alue;
}
</script>
```

```
<form name="form1">
<select size="1" name="select1"
ONCHANGE="cambiarColor()">
<option value="0xff0000">Rojo</option>
<option value="0x00ff00">Verde</option>
<option value="0x0000ff">Azul</option>
</select></p>
<br>
Número de índice seleccionado del objeto
SELECT:<input type="text"
name="text1"><br>
Texto seleccionado:<input type="text"
name="text2"><br>
Valor asociado:<input type="text"
name="text3"><br>
</form></body></html>
```


Control CHECKBOX.

Página que muestra 4 lenguajes de programación que el usuario puede seleccionar si los conoce. Luego muestra un mensaje indicando la cantidad de lenguajes que ha seleccionado.

```
<html><head></head>
<body>
<script language="JavaScript">
function contarSeleccionados(){
    var cant=0;
    if (document.form1.lenguaje1.checked)
        cant++;
    if (document.form1.lenguaje2.checked)
        cant++;
    if (document.form1.lenguaje3.checked)
        cant++;
    if (document.form1.lenguaje4.checked)
        cant++;
    alert('Conoce ' + cant + ' lenguajes');
}
</script>
```

```
<form name="form1">
<input type="checkbox"
name="lenguaje1">JavaScript
<br>
<input type="checkbox"
name="lenguaje2">PHP
<br>
<input type="checkbox"
name="lenguaje3">JSP
<br>
<input type="checkbox"
name="lenguaje4">VB.Net
<br>
<input type="button" value="Mostrar"
onClick="contarSeleccionados()">
</form></body></html>
```

Control RADIO / TEXTAREA.

Tarea.

PROBLEMA

Confeccionar una página que muestre dos objetos de la clase RADIO solicitando que seleccione si es mayor de 18 años o no. Al presionar un botón mostrar que radio está seleccionado.

PROBLEMA

Confeccionar una página de visitas a un sitio, solicitar ingresar el nombre de una persona, su mail y los comentarios (TEXTAREA). Mostrar luego llamando a la función **alert** los datos ingresados.

Eventos onFocus y onBlur

- El evento onFocus se dispara cuando el objeto toma foco y el evento onBlur cuando el objeto pierde el foco.
- Ej: formulario que solicita la carga del nombre y la edad de una persona. Cuando el control toma foco borrar el contenido actual, al abandonar el mismo, mostrar un mensaje de alerta si el mismo está vacío.

Eventos onFocus y onBlur

```
<html>
<head></head>
<body>
<script language="JavaScript">
function vaciar(control)
{
    control.value="";
}

function verificarEntrada(control)
{
    if (control.value=="")
        alert('Debe ingresar datos');
}
</script>
```

```
<form name="form1">
Ingrese su nombre:
<input type="text" name="nombre"
onFocus="vaciar(this)"
onBlur="verificarEntrada(thi
s)"><br>
Ingrese su edad:
<input type="text" name="edad"
onFocus="vaciar(this)"
onBlur="verificarEntrada(this)"><
br>
<input type="button"
value="Confirmar">
</form>
</body></html>
```

Eventos onMouseOver y onMouseOut

- El evento onMouseOver se ejecuta cuando pasamos la flecha del mouse sobre un hipervínculo y el evento onMouseOut cuando la flecha abandona el mismo.
- Ej: Página que cambie el color de fondo del documento cuando el mouse pase sobre un link.

Eventos onMouseOver y onMouseOut

```
<html>
<head></head>
<body>
<script language="JavaScript">
function pintar(col)
{
    document.bgColor=col;
}
</script>

<a href="pagina1.html"
onMouseOver="pintar('#ff0000
')"
onMouseOut="pintar('#ffffff')
">Rojo</a>
-
```

```
<a href="pagina1.html"
onMouseOver="pintar('#00ff00')
"
onMouseOut="pintar('#ffffff')
">V
erde</a>
-
<a href="pagina1.html"
onMouseOver="pintar('#0000ff')
"
onMouseOut="pintar('#ffffff')
">A
zul</a>
<br>
<br>
<br>
<a href="pagina2.html">ver segundo
problema</a>
</body>
```

Evento onLoad

- El evento **onLoad** se ejecuta cuando cargamos una página en el navegador.
- Uno de los usos más frecuentes es para fijar el foco en algún control de un formulario, para que el operador no tenga que activar con el mouse dicho control.
- Este evento está asociado a la marca body.

Evento onLoad

```
<html><head></head>
<body onLoad="activarPrimerControl()">
<script language="JavaScript">
function activarPrimerControl()
{
  document.form1.nombre.focus();
}
</script>
<form name="form1">
Ingrese su nombre:
<input type="text" name="nombre"><br>
Ingrese su edad:
<input type="text" name="edad"><br>
<input type="button" value="Confirmar">
</form>
</body></html>
```

Archivo JavaScript externo (*.js)

- El lenguaje JavaScript permite agrupar funciones y disponerlas en un archivo **separado** a la página HTML.
- Esto trae muchos beneficios:
 - Reutilización de funciones en muchos archivos. No tenemos que copiar y pegar sucesivamente las funciones en las páginas en las que necesitamos.
 - Facilita el mantenimiento de las funciones al encontrarse en archivos separados.
 - Nos obliga a ser más ordenados.

Archivo JavaScript externo (*.js)

La mecánica para implementar estos archivos externos en JavaScript es:

1. Crear un archivo con extensión ***.js** ,
2. Escribir las funciones en la misma:

```
function retornarFecha()  
{  
  var fecha  
  fecha=new Date();  
  var cadena=fecha.getDate()+ '/' +(fecha.getMonth()+1)+ '/' +fecha.getFullYear();  
  return cadena;  
}
```

```
function retornarHora()  
{  
  var fecha  
  fecha=new Date();  
  var cadena=fecha.getHours()+ ':' +fecha.getMinutes()+ ':' +fecha.getSeconds();  
  return cadena;  
}
```

Archivo JavaScript externo (*.js)

3. Creamos un archivo html que utilizará las funciones contenidas en el archivo *.js:

```
<html>
<head>
<title>Problema</title>
  <script language="javascript" type="text/javascript"
    src="pagina2.js"></script>
</head>
<body>
<script language="javascript">
document.write('La fecha de hoy es:'+retornarFecha());
document.write('<br>');
document.write('La hora es:'+retornarHora());
</script>
</body>
</html>
```