

¿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).

JavaScript no es un lenguaje de programación propiamente dicho como C, C++, Delphi, etc. Es un lenguaje script u orientado a documento.

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.

Ejemplo:

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

Ya veremos que los nombres de funciones llevan letras en mayúsculas.

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.

Ejemplos:

```
var dia, mes, anio;
```

```
var edad=20;
```

```
var edad;
```

```
edad=20;
```

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:

```
<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>
```

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.

```
<html>
  <head>
  </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>);

```

Estructuras secuenciales de programación

Realizar la carga de dos números por teclado e imprimir su suma y su producto:

```

<html>
<head>
<script type="text/javascript">
    var valor1;
    var valor2;
    valor1=prompt('Ingrese primer número:', '');
    valor2=prompt('Ingrese segundo número:', '');
    var suma=parseInt(valor1)+parseInt(valor2);
    var producto=parseInt(valor1)*parseInt(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 simples.

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

```

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

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

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

Estructuras condicionales compuestas

```
<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>
```

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.

Estructuras condicionales anidadas.

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".

Solución:

```
<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>

```

Estructura repetitiva (do/while)

La sentencia do/while es otra estructura repetitiva, la cual ejecuta al menos una vez su bloque repetitivo, a diferencia del while que puede no ejecutar el bloque.

```

<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 repetitiva (for)

Esta estructura se emplea en aquellas situaciones en las cuales CONOCEMOS la cantidad de veces que queremos que se ejecute el bloque de instrucciones.

```
<html>
<head>
</head>
<body>

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

</body>
</html>
```

Funciones

En programación es muy frecuente que un determinado procedimiento de cálculo definido por un grupo de sentencias tenga que repetirse varias veces, ya sea en un mismo programa o en otros programas, lo cual implica que se tenga que escribir tantos grupos de aquellas sentencias como veces aparezca dicho proceso.

```
<html>
<head>
</head>
<body>

<script type="text/javascript">
    document.write("Cuidado<br>");
    document.write("Ingrese su documento correctamente<br>");
    document.write("Cuidado<br>");
    document.write("Ingrese su documento correctamente<br>");
    document.write("Cuidado<br>");
    document.write("Ingrese su documento correctamente<br>");
</script>

</body>
</html>
```

Empleando función:

```
<html>
<head>
</head>
<body>

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

```

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

```

```

</body>
</html>

```

Funciones con parámetros.

```

<html>
<head>
</head>
<body>
<script type="text/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>
</html>

```

Funciones que retornan un valor.

```

<html>
<head>
</head>
<body>
<script type="text/javascript">

    function convertirCastellano(x)
    {
        if (x==1)
            return "uno";
        else
            if (x==2)
                return "dos";
            else
                if (x==3)
                    return "tres";
                else
                    if (x==4)
                        return "cuatro";
                    else

```

```

        if (x==5)
            return "cinco";
        else
            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>
</html>

```

Ejemplo 2: Confeccionar una función que reciba una fecha con el formato de día, mes y año y retorne un string con un formato similar a: "Hoy es 10 de junio de 2013".

```

<html>
<head>
</head>
<body>
<script type="text/javascript">

    function formatearFecha(dia,mes,año)
    {
        var s='Hoy es '+dia+' de ';
        switch (mes) {
            case 1:s=s+'enero ';
                break;
            case 2:s=s+'febrero ';
                break;
            case 3:s=s+'marzo ';
                break;
            case 4:s=s+'abril ';
                break;
            case 5:s=s+'mayo ';
                break;
            case 6:s=s+'junio ';
                break;
            case 7:s=s+'julio ';
                break;
            case 8:s=s+'agosto ';
                break;
            case 9:s=s+'septiembre ';
                break;
            case 10:s=s+'octubre ';
                break;
            case 11:s=s+'noviembre ';
                break;
            case 12:s=s+'diciembre ';
                break;
        } //fin del switch
        s=s+'de '+año;
        return s;
    }
    document.write(formatearFecha(11,6,2013));

</script>
</body>
</html>

```


Programación orientada a objetos.

Un objeto es una estructura que contiene tanto las variables (llamadas propiedades) como las funciones que manipulan dichas variables (llamadas métodos). A partir de esta estructura se ha creado un nuevo modelo de programación (la programación orientada a objetos) que atribuye a los mismos propiedades como herencia o polimorfismo. Como veremos, JavaScript simplifica en algo este modelo y hace una programación híbrida entre la programación estructurada y la programación orientada a objetos.

El modelo de la programación orientada a objetos normal y corriente separa los mismos en dos: clases e instancias (objetos). Las primeras son entes más abstractos que definen un conjunto determinado de objetos. Las segundas son miembros de una clase, poseyendo las mismas propiedades que la clase a la cual pertenecen.

Para acceder a los métodos y propiedades de un objeto debemos utilizar la siguiente sintaxis:

`objeto.propiedad`

`objeto.metodo(parametros)`

Ejemplos de objetos del mundo real:

- Casa:
 - atributos: tamaño, precio, cantidad de habitaciones, etc.;
 - responsabilidades: comodidad, seguridad, etc.
- Mesa:
 - atributos: altura, largo, ancho, etc.;
 - responsabilidades: contener elementos.
- Ventana:
 - atributos: tamaño, color, etc.;
 - responsabilidades: abrirse, cerrarse, etc.

Ejemplos de objetos del mundo de la programación:

- Ventana:
 - atributos: tamaño, color, etc.;
 - responsabilidades: mostrar título, achicarse etc.

Clases

Una clase es un molde para objetos que poseen las mismas características (que pueden recibir los mismos mensajes y responden de la misma manera).

La clase a la cual pertenece un objeto determina el comportamiento del objeto.

Una clase tiene encomendadas actividades que ejecutan los métodos.

Las clases están definidas por:

- Atributos (Propiedades),
- Comportamiento (operaciones o métodos) y
- Relaciones con otros objetos.

Una aplicación es un conjunto de objetos de determinadas clases.

Clase Date

JavaScript dispone de varias clases predefinidas para acceder a muchas de las funciones normales de cualquier lenguaje, como puede ser el manejo de vectores o el de fechas.

El objeto Date dispone, entre otros, de los siguientes métodos:

```

getYear()
setYear(año)
    Obtiene y coloca, respectivamente, el año de la fecha.
    Éste se devuelve como número de 4 dígitos excepto en el
    caso en que esté entre 1900 y 1999, en cuyo caso
    devolverá las dos últimas cifras.
getFullYear()
setFullYear(año)
    Realizan la misma función que los anteriores, pero sin
    tanta complicación, ya que siempre devuelven números
    con todos sus dígitos.
getMonth()
setMonth(mes)
getDate()
setDate(día)
getHours()
setHours(horas)
getMinutes()
setMinutes(minutos)
getSeconds()
setSeconds(segundos)
    Obtienen y colocan, respectivamente, el mes, día, hora,
    minuto y segundo de la fecha.
getDay()
    Devuelve el día de la semana de la fecha en forma de
    número que va del 0 (domingo) al 6 (sábado)

```

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

```

<HTML>
<HEAD>

<SCRIPT type="text/javascript">
    function mostrarFechaHora()
    {
        var fecha
        fecha=new Date();
        document.write('Hoy es ');
        document.write(fecha.getDate()+'/');
        document.write((fecha.getMonth()+1)+'/');
        document.write(fecha.getFullYear());
        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>

</BODY>

</HTML>

```

Clase Array

Un vector es una estructura de datos que permite almacenar un CONJUNTO de datos.

Ejemplo 1: Crear un vector para almacenar los cinco sueldos de operarios y luego mostrar el total de gastos en sueldos (cada actividad en una función).

```
<html>
<head>
</head>
<body>

<script type="text/javascript">
    function cargar(sueldos)
    {
        var f;
        for(f=0;f<sueldos.length;f++)
        {
            var v;
            v=prompt('Ingrese sueldo:', '');
            sueldos[f]=parseInt(v);
        }
    }

    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>
```

Ejemplo 2: Crear un vector con elementos de tipo string. Almacenar los meses de año. En otra función solicitar el ingreso de un número entre 1 y 12. Mostrar a qué mes corresponde y cuántos días tiene dicho mes.

```
<html>
<head>
</head>
<body>

<script type="text/javascript">
    function mostrarFecha(meses,dias)
    {
        var num;
```

```

    num=prompt('Ingrese número de mes:', '');
    num=parseInt(num);
    document.write('Corresponde al mes:'+meses[num-1]);
    document.write('<br>');
    document.write('Tiene '+dias[num-1]+' días');
}

var meses;
meses=new Array(12);
meses[0]='Enero';
meses[1]='Febrero';
meses[2]='Marzo';
meses[3]='Abril';
meses[4]='Mayo';
meses[5]='Junio';
meses[6]='Julio';
meses[7]='Agosto';
meses[8]='Septiembre';
meses[9]='Octubre';
meses[10]='Noviembre';
meses[11]='Diciembre';

var dias;
dias=new Array(12);
dias[0]=31;
dias[1]=28;
dias[2]=31;
dias[3]=30;
dias[4]=31;
dias[5]=30;
dias[6]=31;
dias[7]=31;
dias[8]=30;
dias[9]=31;
dias[10]=30;
dias[11]=31;
mostrarFecha(meses,dias);
</script>

</body>
</html>

```

Clase Math

Esta clase es 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
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

Ejemplo: Confeccionar un programa que permita 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:

```
<html>
<head>
</head>
<body>

<script type="text/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>
</html>
```

Clase String

Ejemplo: Cargar un string por teclado y luego llamar a los distintos métodos de la clase String y la propiedad length.

```
<html>
<head>
</head>
<body>

<script type="text/javascript">
    var cadena=prompt('Ingrese una cadena:', '');
    document.write('La cadena ingresada es: '+cadena);
    document.write('<br>');
    document.write('La cantidad de caracteres son: '+cadena.length);
```

```
document.write('<br>');
document.write('El primer carácter es:'+cadena.charAt(0));
document.write('<br>');
document.write('Los primeros 3 caracteres
son:'+cadena.substring(0,3));
document.write('<br>');
if (cadena.indexOf('hola')!=-1)
    document.write('Se ingresó la subcadena hola');
else
    document.write('No se ingresó la subcadena hola');
document.write('<br>');
document.write('La cadena convertida a mayúsculas
es:'+cadena.toUpperCase());
document.write('<br>');
document.write('La cadena convertida a minúsculas
es:'+cadena.toLowerCase());
document.write('<br>');
</script>

</body>
</html>
```