JavaScript

(Manual)

Antes de empezar:

- **Manual F.V.**
  Significa "manual **práctico** de informática", pero realmente realmente PRÁCTICO.

- En el texto me refiero continuamente a *TuCarpeta*, esto es un subdirectorio de tu ordenador donde es conveniente que grabes todos los programas de este "tu manual".

- Todos los programas y ejercicios del manual han sido probados sin problema en el **Microsoft Internet Explorer 5.0**

- Mi consejo, es que escribas de nuevo tal como te indico en el texto, cada uno de los programas utilizando el **Bloc de Notas** del Windows. Sólo en el caso de que el programa no te funcione y no descubras cuál es el error, cópialo a partir del Word en el Bloc de Notas y grábalo con otro nombre; para poder compararlo con el tuyo y así descubrir el error.
  Piensa que se aprende más descubriendo los propios errores, que avanzar sin hacer ningún error.

ÍNDICE

I.- Introducción a la programación en JavaScript

1.- Introducción

- De todos los servicios que ofrece **INTERNET**, no cabe duda de que el más popular es la WWW (World Wide Web).

  La WWW no es más que millones de páginas en formato electrónico, con los contenidos y temáticas más diversas a las que podemos acceder gracias a un ordenador + modem + browser (navegador).

- Una página **WEB** no es más que un fichero de texto (ASCII), escrito en formato HTML (Hyper Text Markup Language = lenguaje etiquetado de hipertexto).

- El HTML es un lenguaje basado en pares de **tags** (etiquetas). Un **tag** es un código entre <>, si es de apertura o un código entre </ > si es de cierre.

  Los **browsers** (navegadores de Internet), son capaces de traducir estas etiquetas (tags) de forma que presentan visualmente la página.

- **Estructura de un fichero HTML**

  Básicamente consta de cuatro pares de etiquetas:

```
<HTML>
<HEAD>
<TITLE>Nuestra primera página</TITLE>
</HEAD>
<BODY>
.....
.....                    Lo que visualizará nuestra página
.....
</BODY>
</HTML>
```

  - El par <HTML> y </HTML>
    Determina que un fichero sea HTML

  - El par <HEAD> y </HEAD>
    Determina la cabecera del fichero HTML, que puede contener un título.

  - El par <TITLE> y </TITLE>
    Encierra el "título": frase de texto que aparecerá en el marco del navegador (primera línea), al ejecutar el fichero HTML

  - El par <BODY> y </BODY>
    Encierra el **contenido** de la página **html**, es decir lo que se visualizará en el navegador.

Los navegadores (browsers) más conocidos son:

- Netscape Comunicator
- Microsoft Internet Explorer

- Ejecuta el **"Bloc de Notas"**, es decir:
  [Inicio]
  Programas
  Accesorios
  Bloc de Notas

- Escribe:

```
<HTML>
<HEAD>
<TITLE>PROG000.HTM</TITLE>
</HEAD>
<BODY>
<P>Esto aparecerá en el navegador porque es un párrafo</P>
<P>Esto es otro párrafo</P>
<P>Observa lo que aparece en la línea de título</P>
<P>La línea de título es la frase que aparece en el marco
  del navegador, línea superior</P>
<P>¿Está claro para qué sirven los tags P, /P?</P>
</BODY>
</HTML>
```

- Graba el fichero anterior con el nombre **PROG000.HTM** en *TuCarpeta*

- Ejecuta el "Explorador de Windows". Sitúate en *TuCarpeta* y haz un doble click en **PROG000.HTM** De esta forma, se ejecuta el navegador de tu ordenador (supondré que es el **Internet Explorer**) y se "carga" la página **PROG000.HTM**

  El navegador nos muestra la página HTML "visualmente", para ver su código debes hacer:
  Menú Ver
  Código fuente

- La programación en HTML no tiene ninguna utilidad para un usuario normal, ya que en el mercado existen herramientas que evitan la necesidad de tener que introducir manualmente los "tags": HotMetal, FontPage, Word, etc.
  Lo que sí tienen sentido es el estudio de la programación en **JavaScript**.

- **JavaScript** es un lenguaje de programación creado por la empresa **Netscape** (creadora de uno de los navegadores más conocido)
  Es el lenguaje de programación más utilizado en Internet para añadir interactividad a las páginas Web

- No confundir el **JavaScript** con el **Java**. El **Java** es un lenguaje de programación de propósito general como lo son el **C++** o el **Visual Basic**.

- Un programa en **JavaScript** se integra en una página Web (entre el código HTML) y es el navegador el que lo interpreta (ejecuta). Es decir el JavaScript es un lenguaje **interpretado**, no **compilado** (no se genera ningún tipo de fichero objeto o exe).

- Para programar en **JavaScript** sólo necesitamos un editor de texto (utilizaremos el **Bloc de Notas** del Windows) y un navegador (utilizaremos el **Microsoft Internet Explorer**) para ejecutarlo.

- ¿Porqué el **JavaScript** y no otro lenguaje de programación?
  Porque:
  - Es moderno (tiene pocos años)
  - Es sencillo (su hermano mayor: el **Java**, es bastante más complejo)
  - Es útil (el desarrollo de Internet, se prevé muy rápido en los próximos años)
  - Es potente: permite la moderna **POO** (programación orientada a objetos)
  - Es barato: sólo necesitamos un editor de textos (el "Bloc de Notas" está incluido en el Windows) y un navegador (es gratuito, ya sea el "Internet Explorer" o el "Netscape").

- Es visual: permite la moderna "programación visual" (ventanas, botones, colores, formularios, etc.).

En definitiva: **es ideal para un primer curso de introducción a la programación.**
Y has de tener en cuenta que hay un "dicho" en informática, que afirma: **"Si dominas un lenguaje de programación, los conoces todos"**.

2.- Sintaxis básica

- Escribe, utilizando el "Bloc de Notas" del Windows, el siguiente **programa:**

```
<HTML>
<HEAD>
<TITLE>PROG001.HTM</TITLE>
<SCRIPT LANGUAGE="JavaScript">

alert("¡Hola Mundo!");

</SCRIPT>
</HEAD>
<BODY>
<P>
Programa 1 en JavaScript
</P>
</BODY>
</HTML>
```

- Graba el fichero anterior en *TuCarpeta* con el nombre **Prog001.htm**

- Ejecuta el programa **Prog001.htm**, es decir:
  - Ejecuta el "Explorador de Windows"
  - Sitúate en *TuCarpeta*
  - Clic-Click en **Prog001.htm**

- Estudio del **Prog001.htm**:
  - Un programa "JavaScript" se escribe **integrado** en una página HTML, por lo tanto no es más que un fichero de texto que contiene una serie de **pares de tags** correspondientes a la página Web (como mínimo el par: <HTML>, </HTML>), además del par de tags característico de un programa **JavaScript**. Dicho fichero se ha de grabar necesariamente con la extensión **HTM** (característica de una página HTML).
  - Un programa "JavaScript" no es más que una secuencia de ordenes, que deben **terminar en punto y coma**, entre los tags:
    <SCRIPT LANGUAGE="JavaScript">
    y
    </SCRIPT>
  - En nuestro **Prog001.htm**, hemos incluido el programa en la cabecera (HEAD) de la página, pero podríamos colocarlo en cualquier parte del fichero **htm**
  - Nuestro primer programa JavaScript contiene una única sentencia: **alert(""¡Hola Mundo!");**
    Que "abre" una ventana con el mensaje que tenemos entre comillas.
    Al hacer clic en el **[Aceptar]** de la ventana **"alert"**, se acaba el programa JavaScript (se encuentra el tag </SCRIPT>) y continua ejecutándose la página HTML.

- Utilizando el "Bloc de Notas" escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG002.HTM
var nom;
nom=prompt("Escribe tu nombre ","Paco");
alert("Mucho gusto "+ nom);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog002.htm**

- Ejecútalo varias veces, observando detenidamente lo que sucede.

- Estudio del **Prog002.htm**

  1º) Primera y última línea: <HTML> y </HTML>
  Es decir: página html mínima que necesitamos para incluir un programa JavaScript.

  2º) Segunda y penúltima líneas: <SCRIPT LANGUAGE=.....> y </SCRIPT>, es decir programa en JavaScript

  3º) Primera sentencia del programa: **// PROG002.HTM**
  Única línea del programa que no es necesario acabarla con punto y coma.
  Todas las líneas que empiezan por // son comentarios para el programador, es decir no forman parte del programa, dicho de otra forma: el navegador si encuentra una línea que empieza por //, la salta.

  4º) **var nom;**
  Definimos una **variable** de nombre **nom**

  5º) **nom = prompt("Escribe tu nombre","Paco");**
  Aparece un recuadro con un mensaje y un campo donde podemos escribir algo; el mensaje corresponde a lo que escribimos en el primer argumento de la función **prompt**, encerrado entre comillas. El segundo argumento del **prompt** contiene el valor que aparece por defecto en el campo del cuadro de diálogo.
  El valor del **prompt** es **nom**, es decir lo que nosotros escribamos en el cuadro será el valor que tomará la variable **nom**.
  Si no escribimos nada y hacemos click en [Aceptar], el **prompt**, es decir la variable **nom** tomará el valor de **Paco**, porque es el valor que aparece por defecto.

  6º) **alert("Mucho gusto "+nom);**
  Aparece un cuadro con el mensaje "Mucho gusto" y a continuación el valor de la variable "nom", que será lo que hemos escrito en el primer cuadro que nos ha aparecido.

En definitiva:
- La función **prompt** nos permite introducir "valores", dichos valores se han de guardar en **variables**, que previamente hemos de declarar con la palabra reservada **"var"**
- La función **"alert"** muestra mensajes y/o valores de variables.

- Utilizando el "Bloc de Notas del Windows" escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG003.HTM
/*Programa que sirve para calcular
  el área de un rectángulo */
```

```
var base,altura;
base=prompt("Escribe la base del Rectángulo","");
altura=prompt("Escribe la altura del Rectángulo","");
alert("El área del Rectángulo es = "+(base*altura));
</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog003.htm**

- Ejecútalo varias veces. Sería conveniente utilizar números enteros y también decimales (5.72, 0.531: observa que has de utilizar el punto decimal inglés).

- Si has ejecutado el programa una vez, para volverlo a ejecutar, no es necesario que "cierres" el navegador, basta que hagas:

  Menú Ver
  Actualizar
  o si quieres ir más rápido, pulsa la tecla **[F5]**

- Es importante que tengas claro este programa:
  - Declaramos dos variables (**var**), que necesitamos para introducir la base y la altura del rectángulo, a través de dos "prompts":
    base= prompt.....
    altura= prompt.....
  - Por último necesitamos dos "alerts" que nos **muestre** el resultado del programa, que es simplemente el producto **base * altura**

- El único elemento nuevo que aparece en el **Prog003.htm** es:

  ```
  /*Programa que sirve para calcular
    el área de un rectángulo */
  ```

  Todo lo que aparece escrito entre **/* y */** no es más que un comentario para el programador, igual que pasaba con las líneas que empezaban por //
  La diferencia entre // y /* */ está en que esta última forma de incluir un comentario, nos permite colocarlo de forma que ocupe más de una línea.

3.- Variables

- **Declaración de variables**

Una variable se puede declarar en **JavaScript**, de dos formas:

- Forma Explícita: **var** nombre Variable;
- Forma Implícita: **var** nombre Variable= valor;

En el último caso no es imprescindible escribir **var**, pero es conveniente, ya que de esta forma localizamos rápidamente todas las variables del programa.

El "JavaScript" es un lenguaje de programación **"Case Sensitive"**, esto es: no es lo mismo las mayúsculas que las minúsculas. Es decir, para el JavaScript: **pepe** es distinto de **Pepe** y distinto de **pEpe**.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE=''JavaScript''>
```

```
// PROG004.HTM
/* Programa que utiliza una variable explícita
   y dos implícitas */

var Expli;
var pi=3.141592;
var radio=7;
Expli=pi*radio*radio;
alert("Área del Círculo = "+Expli);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog004.htm**

- Ejecútalo

4.- Tipos de Datos

Cuando declaramos una variable, ésta no pertenece a ningún tipo de dato en concreto, se dice que es **Undefined**. Es al asignarle un valor cuando pasa a ser de uno u otro tipo, según el dato que albergue.

Existen 6 tipos de datos:

String: cadenas de texto	**Object**: objetos
Number: valores numéricos	**Null**: nulo
Boolean: true o false	**Undefined**: no definido.

Podemos averiguar el tipo de dato que contiene una variable si utilizamos la función incorporada **typeof**

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG005.HTM
var Pepe;
var PEPE="Hola que tal ";
var pepE=75.47;
var pEpe=" ¿Como estás?";

Pepe=PEPE+pEpe;
alert("PEPE="+PEPE);
alert("PEPE es "+typeof(PEPE));
alert("pepE="+pepE);
alert("pepE es "+typeof(pepE));
alert("pEpe="+pEpe);
alert("pEpe es "+typeof(pEpe));
alert("Pepe="+Pepe);
alert("Pepe es "+typeof(Pepe));

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog005.htm**

- Ejecútalo tantas veces como quieras.

5.- El tipo de dato "String":

En JavaScript los textos se denotan por comillas dobles o comillas simples, pero no ambas a la vez.

```
Variable1 = "Pepito"
Variable2 = 'Paquito'
```

Podemos incluir un **carácter de control** en una cadena de texto, si utilizamos el llamado **carácter de escape** que es: \
Los caracteres de control más usados son:

\n salto de línea
\t tabulador

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG006.HTM

var num;
alert("Hola que tal");
alert("Hola \nque tal");
alert("Hola\t"+"que"+"\n"+"tal");
alert("Hola\n que\n t\nal");
num=prompt("Escribe un número: ","");
alert("El \ndoble\n es: \n"+(num*2));

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog006.htm**

- Ejecútalo, observando detenidamente la acción de **\n** y **\t**

- Si no introducimos un número en el "prompt", en el "alert" correspondiente al doble del número, aparecerá **NaN**, que quiere decir que no es un número.

6.- El tipo de datos "Number"

Podemos guardar indistintamente en una variable **number** un número entero, decimal, positivo o negativo.
Ejemplos:

```
var numNatural= 1234;
var numEntero = -479;
var numDecimal = 3.141592;
var numDecimal2 = -0.123;
```

- **Bases de Numeración en JavaScript (números enteros)**

Por defecto, el sistema de numeración es el decimal.

- Base Hexadecimal (base 16): anteponemos el prefijo **0X**
- Base Octal (base 8): anteponemos **un cero**

Dígitos del sistema decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Dígitos del sistema hexadecimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A (que corresponde al 10 en base decimal), B (que corresponde al 11 en decimal), C (12), D (13), E (14), F (15)

Ejemplo:
FF32 en base 16 es $2 + 3*16 + 15*16^2 + 15*16^3 = 65330$ en sistema decimal
Es decir: **0xFF32 (número en base 16, según notación de JavaScript) = $FF32_{16} = 65330_{10}$**

$12345_8 = 5 + 4*8 + 3*8^2 + 2*8^3 + 1*8^4 = 5349_{10}$
Es decir: **012345 (número en base 8, según notación de JavaScript) = $12345_8 = 5349_{10}$**

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE=''JavaScript''>

// PROG007.HTM

var n1=57;          // número en base 10
var n2=012345;      //base 8, porque empieza por 0
var n3=0xFF32;      //base 16, porque empieza por 0x
alert("número decimal= "+n1);
alert("el 12345 en base 8 es en decimal= "+n2);
alert("el FF32 en base 16 es en decimal= "+n3);

/* Observa que al escribir una variable numérica en un "alert"
   siempre nos da el número en decimal,
   aunque sea en octal o hexadecimal  */

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog007.htm** y ejecútalo.

En el programa anterior aparecen 3 "alerts", uno para cada uno de los tres números. No hay ningún problema para **incluir** los tres "alerts" en uno solo....

- En efecto, escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE=''JavaScript''>

// PROG007b.HTM

var n1=57;          // número en base 10
var n2=012345;      //base 8, porque empieza por 0
var n3=0xFF32;      //base 16, porque empieza por 0x
alert("número decimal= "+n1+"\n"+
   "el 12345 en base 8 es en decimal= "+n2+"\n"+
   "el FF32 en base 16 es en decimal= "+n3);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog007b.htm** y ejecútalo.

- Observa que la solución de nuestro problema está:
    - Unir texto (entre comillas) y variables (sin comillas), con el signo "+"
    - Cada vez que deseemos un cambio de línea, incluimos "\n"

Veamos otra forma de incluir en un "alert" muchos datos ...

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// Prog007bb.htm
var x,y,z;
salida="";
var x1="Pepito Grillo", y1="Paquita González";
var num1=37, num2=3.752;
x=prompt("Escribe tu nombre:","");
y=prompt("Escribe tu primer apellido:","");
z=prompt("Escribe tu segundo apellido:","");
salida=salida+"Ahora un alert largo ";
salida=salida+x+y+z;
salida=salida+x1+" "+y1;
salida=salida+num1+" "+num2;
salida=salida+" ,ya me he cansado";
alert(salida);
</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog007bb.htm** y ejecútalo.

- Observa de qué forma **acumulamos** muchos datos en un solo "alert" (ésta será la forma de proceder, cuando necesitemos una "salida" con muchos valores):
    - definimos una variable "vacía": **var salida="";**
    - **acumulamos** a la variable **salida** todo lo que queramos: **salida=salida+ lo que sea**
    - "lo que sea" puede ser una cadena (un texto) o una variable (sea numérica o de texto).
      salida=salida + x1 + " " + y1;
      salida=salida + "ya me he cansado";
    - Para acabar, un solo "alert": **alert(salida);**

Variables nulas

Cuando una variable no contiene ningún valor, su contenido es **nulo**

Ejemplo:
```
miVariable = "Hola";
miVariable= null; // la vaciamos para que no ocupe memoria.
```

Valores especiales para variables numéricas

NaN: no es un número.
Infinity: infinito, por ejemplo 3/0

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG008.HTM
var var1="Pepe";
var var2="Paquito";
var var3=5/0;
alert("Pepe es "+var1);
alert("Paquito + 2 es ="+(var2+2));
alert("5/0 es ="+var3);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog008.htm** y ejecútalo.

7.- Contadores

Uno de los instrumentos más utilizados en todo lenguaje de programación es el llamado **contador**
Observa:

```
var x= 10;
x = x+1;
x = x-5;
```

Declaramos una variable "x", que es numérica y que inicialmente es 10.
La línea "x = x+1" es un contador, que hemos de leer: **El nuevo valor de "x" es igual al anterior valor de "x" más una unidad**. Es decir, que en estos momentos nuestra variable "x" es igual a 11.
La siguiente línea: "x = x-5" es otro contador que hemos de leer: **el nuevo valor de "x" es igual al anterior valor de "x" menos 5 unidades**. Es decir, que el valor de la variable "x" es ahora 11 - 5 = **6**

- Haz un programa para comprobar lo que hemos dicho sobre los contadores, es decir:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG009.HTM
var x=10;
alert("El valor inicial de x es "+x);
x=x+1;
alert("Después de x=x+1, x="+x);
x=x-5;
alert("Después de x=x-5, x="+x);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog009.htm** y ejecútalo.

8.- Conversión entre tipos de datos

- **Conversión implícita de tipos**

Observa:
```
var var1 = "75";
var var2 = 25;
var var3, var4;
var3 = var1 + var2;
var4 = var2 + var1;
```

Las varibles **var3 y var4** contienen ¿números o textos?
Cuando se suman **cadenas de texto** con cualquier otra cosa, los otros tipos de datos se convierten en cadenas de texto. Pero si restamos, multiplicamos o dividimos "cadenas de texto", ¿sucede lo mismo?. Vamos a descubrirlo en los siguientes programas.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG010.HTM

var num1="30";
var num2="15";

// Observa que definimos dos cadenas de texto

alert("30+15= "+(num1+num2));
alert("30*15= "+(num1*num2));

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog010.htm**

- Ejecútalo.

- La conclusión está clara:
  - Si **sumamos** dos cadenas (aunque contengan números) de texto se produce la concatenación (unión) de los dos textos.
  - Si **multiplicamos** (o hacemos cualquier operación aritmética que no sea la **suma**), dos cadenas de texo que en realidad contienen números, se produce **una conversión implícita** de las cadenas a números y aparece el producto aritmético de los números.

- ¿Qué sucede si introducimos dos números a través de la función **prompt**?. Vamos a verlo:

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

//PROG011.HTM

var num1,num2;
num1=prompt("Escribe un número","");
num2=prompt("Escribe otro número","");
alert("La suma es ="+(num1+num2));

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta*, con el nombre **Prog011.htm**

- Ejecútalo. No funciona como deseamos debido a que el resultado de un **prompt**, siempre es una **cadena de texto**.

El problema que tenemos pendiente, es ¿cómo sumar en JavaScript?

- **Conversión explícita de tipos**

parseFloat(cadena)

Toma la "cadena" y la transforma en un número en coma flotante, si es posible.

parseFloat ("123.456") = 123.456
parseFloat ("123ABC") = 123
parseFloat ("ABC") = NaN

parseInt(cadena, número)

Devuelve números enteros, el segundo argumento nos permite escoger la base de numeración (entre 2 y 36)

parseInt ("ABC",16) = 2748 $ABC_{16} = 2748_{10}$
Si no especificamos el segundo argumento, por defecto es 10.
Si la cadena empieza por **0x** y no existe el segundo argumento, se entiende que es 16.
Si la cadena empieza por 0 y no existe el segundo argumento, se entiende que es 8

toString(argumento)

Si argumento = número
Devuelve una cadena que contiene el número
Puede haber un argumento opcional:
(13).toString(16) =”d” siendo $13_{10} = d_{16}$
(13).toString(2) = “1101” siendo $13_{10} = 1101_2$

- Vamos a hacer un programa que sume números en JavaScript. Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

//PROG012.HTM

var num1,num2;
num1=prompt("Escribe un número","");
num1=parseFloat(num1);
num2=prompt("Escribe otro número","");
num2=parseFloat(num2);
alert("La suma es ="+(num1+num2));

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog012.htm**

- Ejecútalo, utilizando números enteros, naturales y decimales.

A partir de este momento hemos de tener claro si los “prompts” corresponden a números enteros, decimales o de texto:

- Si "x" ha de ser un número entero escribiremos:
  x = parseInt(prompt("Escribe un número entero",""));

- Si "x" ha de ser un número decimal escribiremos:
  x = parseInt(prompt("Escribe un número entero o decimal",""));

- Si "x" ha de ser una cadena de texto escribiremos:
  x = prompt("Escribe el texto correspondiente","");

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// Prog012b.htm
var x,y;
x=parseInt(prompt("Escribe un entero ",""));
y=parseInt(prompt("Escribe otro entero ",""));
alert("La suma de "+ x +" + " + y +" es =" + (x+y) +"\n"+
    "El producto de "+ x +" y "+ y +" = "+(x*y)+"\n"+
    "El promedio de "+ x +" y "+ y +" es "+ (x+y)/2);
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog012b.htm**, y ejecútalo.

En el ejercicio **Prog007.htm** habíamos visto una forma de convertir un número en base 8 o base 16 en base 10. Pero dichos números debíamos de escribirlos implícitamente, es decir en el programa. Nos gustaría hacer un programa que:
- Nos preguntara un número en base 16 (prompt)
- Nos diera como resultado el número anterior pero en base 10

Vamos a ver si lo conseguimos:

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// Prog012bb.htm
var m,n,x;
m=prompt("Escribe un número en base 16","");
n=parseInt(m,16);

alert("El número "+m+" en base 16, es igual a "+n+" en base 10");
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog012bb.htm** en *TuCarpeta* y ejecútalo para el número "FF32", a ver si es verdad que nos da "65330" como resultado.

Observa el funcionamiento del programa **Prog012bb.htm**:

- **m = prompt("Escribe un número en base 16", "")**

  Lo que escribamos (un número en base 16), se guardará en la variable "m" como texto (ya que no hemos puesto ningún "parseInt" ni "parseFloat").

- **n =parseInt(m,16)**

La variable "n" guardará el número en base 10
Es decir: **parseInt(cadena, 16)**, transforma la "cadena", en nuestro ejemplo un número escrito en base 16, en el correspondiente **número** (no cadena) pero en base decimal.

Vamos a ver si el procedimiento sirve para cualquier otra base...

Antes de todo veamos unos cuantos números en diferentes bases:

$36_7 = 6 + 3.7 = 27$ en base 10
$123_4 = 3 + 2.4 + 1.4^2 = 27$ en base 10
$5134_6 = 4 + 3.6 + 1.6^2 + 5.6^3 = 1138$ en base 10

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// Prog012bbb.htm
var m7,n7;
var m4,n4;
var m6,n6;
m7=prompt("Escribe un número en base 7","");
n7=parseInt(m7,7);
m4=prompt("Escribe un número en base 4","");
n4=parseInt(m4,4);
m6=prompt("Escribe un número en base 6","");
n6=parseInt(m6,6);

alert("El número "+m7+" en base 7, es igual a "+n7+" en base 10\n"+
    "El número "+m4+" en base 4, es igual a "+n4+" en base 10\n"+
    "El número "+m6+" en base 6, es igual a "+n6+" en base 10");

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog012bbb.htm** y ejecútalo, probándolo con los números anteriores.

Veamos el problema inverso: dado un número en base 10, nos interesa convertirlo a base 7, por ejemplo.

Matemáticamente:

Sea 593 un número en base 10

$593_{10} = 1505_7$, porque:

En JavaScript deberíamos hacerlo de la siguiente forma, escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// Prog012bbbb.htm
var m,n;
n=parseInt(prompt("Escribe un número entero en base 10",""));
m=(n).toString(7);
alert("El número "+n+" en base 10, es igual a "+m+" en base 7");
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog012bbbb.htm** y ejecútalo, probando su funcionamiento con el número **593**

Autoevaluación I

1) Haz un programa de nombre **Eval1A.htm**, que has de grabar en *TuCarpeta* que sirva para restar dos números cualesquiera, sean enteros o reales. ¿Qué sucede si en lugar de restar dos números, intentamos restar dos textos?.
   Haz un programa de nombre **Eval1Ab.htm**, que demuestre el único caso en que podemos restar dos cadenas de texto.

2) Haz un programa de nombre **Eval1B.htm**, que has de grabar en *TuCarpeta*, que sirva para dividir dos números.
   ¿Qué sucede si en lugar de dividir dos números, intentamos dividir dos textos?
   ¿Qué sucede si el divisor es el número 0?

3) Haz un programa de nombre **Eval1C.htm**, que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:
- El programa nos pregunta nuestro nombre.
- El programa nos pregunta nuestra edad.
- El programa da como resultado nuestro nombre y a continuación los días que hemos vivido hasta el momento (deberás multiplicar la edad por 365).

4) Haz un programa de nombre **Eval1D.htm**, que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:
- El programa nos pide un número.
- El programa nos muestra en una única pantalla (un único "alert"), el doble, el triple y cuadrúple del número que habíamos introducido.

5) El siguiente programa tiene errores. Escríbelo (grábalo con el nombre **Eval1E.htm** en *TuCarpeta*) y corrígelo para que funcione y explica para qué sirve:

```
<HTML>
<SCRIPT LANGUAGE="JavaScrip">
/* EVAL1E.HTM
var a,b;
a=prompt("Escribe la base:")
b=promp("Escribe la altura:")
alert("Área= "+(a*b/2);
</SCRIP>
</HTML>
```

6) Haz un programa de nombre **Eval1F.htm**, que has de grabar en *TuCarpeta*, que sirva para calcular la longitud de una circunferencia y el área del círculo correspondiente.

7) Haz un programa de nombre **Eval1G.htm**, que has de grabar en *TuCarpeta*, que sirva para calcular un determinante de 2º orden.

8) Haz un programa de nombre **Eval1H.htm**, que has de grabar en *TuCarpeta*, igual que el **Eval1G**, pero que presente los 4 elementos del determinante tabulados en 2 filas y 2 columnas.

9) Haz un programa de nombre **Eval1I.htm**, que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:

- El programa nos pide nuestro nombre.
- El programa nos pide nuestro primer apellido.
- El programa nos pide en qué población vivimos.
- El programa presenta una pantalla aproximadamente igual a la siguiente:

==========

Hola nombre Apellido

Adiós habitante de Población

==========

10) Haz un programa de nombre **Eval1J.htm**, que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:

- El programa nos pide un número.
- Utiliza tres contadores:
  - Un contador: suma 5
  - Otro contador: suma 21
  - Otro contador: resta 4
- El programa nos presenta los 4 números de la siguiente forma:
  - La primera línea: el número introducido.
  - La segunda línea: los tres números tabulados, que han resultado de los tres contadores.

De forma que si introducimos el nº 5 debería aparecer:

11) Haz un programa de nombre **Eval1K.htm**, que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:

- El programa nos pide un número entero.
- El programa nos da como resultado el mismo número pero en base 16
- Y por último nos lo escribe en base 5

Comprueba el programa para el número 52. Deberás calcular en primer lugar matemáticamente el valor de 52 en base 16 y en base 5.

12) Haz un programa de nombre **Eval1L.htm**, que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:

- El programa nos pide un número en base ocho
- El programa nos lo escribe en base decimal.
- Y por último en base 2.

Comprueba el programa para el número 6561_8. Deberás resolver en primer lugar el problema matemáticamente.

13) Haz un programa de nombre **Eval1M.htm** que has de grabar en *TuCarpeta*, que funcione de la siguiente forma:

- El programa nos pide un número entero.
- El programa nos pide la **base**
- El programa nos escribe el número introducido en la "base" deseada.

Comprueba el programa para el número 100, en base 2, 3 y 11. Deberás resolver en primer lugar el problema matemáticamente.

14) ¿Qué es la WWW?

15) Cuáles son los "browsers" más conocidos.

16) Escribe un fichero HTML (que no sea un programa JavaScript), que presente la frase: "Hola que tal" y debajo tu nombre y apellidos.

17) ¿Qué encierra el par de tags: <BODY>, </BODY>?

18) ¿Qué relación hay entre el **Java** y el **JavaScript**?

19) ¿Quién creó el lenguaje de programación JavaScript?

20) ¿Cuáles son las características del **JavaScript**?

21) El JavaScript es un lenguaje ¿compilado o interpretado?. ¿Qué programa hace de compilador o interprete?

22) ¿Porqué decimos que el JavaScript es un lenguaje de programación barato?

23) ¿Qué indican las siglas POO?

24) ¿Cómo se llaman y cómo funcionan las dos formas de definir una variable en JavaScript?

25) ¿Porqué el JavaScript es un lenguaje "Case Sensitive"?

26) Nombra todos los tipos de datos que existen en JavaScript.

27) ¿Para qué sirve la función **typeof**?. Inventa un par de ejemplos.

28) ¿Qué es el carácter de escape? ¿Qué es un caráter de control?. Escribe dos caracteres de control e indica para qué sirven.

29) ¿Qué dará por resultado el siguiente programa, y porqué?

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// EVAL1N.HTM
var num=0xA;
alert("Número= "+num);
</SCRIPT>
```

</HTML>

Grábalo con el nombre **Eval1N.htm** en *TuCarpeta*.

30) Calcula el número **9AC1** que está en base 16 a base 10, matemáticamente

31) Calcula el número 7640 que está en base 8 a base 10, matemáticamente.

32) Utiliza programas ya hechos para comprobar los dos ejercicios anteriores.

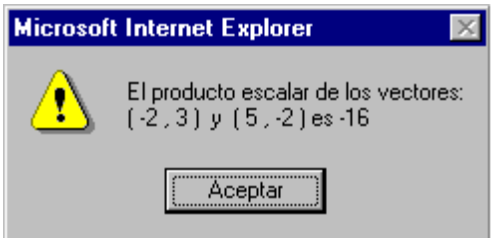
33) ¿Qué diferencia hay entre **null** y **NaN**

34) ¿Qué problema hay al sumar en JavaScript?. Explica cómo solucionarlo.

35) Indica a qué es igual los siguientes valores:
   a) parseInt("A",16)=
   b) parseFloat("31H")=
   c) toString(13)=
   d) toString(4.273,49)=

36) Haz un programa de nombre **Eval1o.htm**, que sirva para calcular el producto escalar de dos vectores del plano.
La "salida" ha de ser de la siguiente forma:

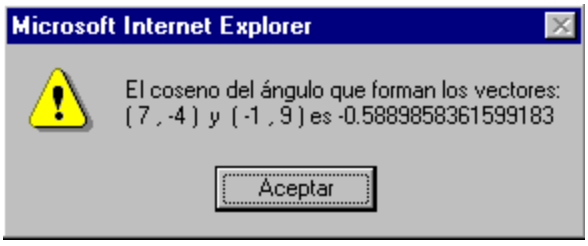
Comprueba el funcionamiento del programa, utilizando el caso concreto que aparece en la ilustración.



37) Haz un programa de nombre **Evalip.htm**, que sirva para calcular el coseno del ángulo que forman dos vectores del plano, sabiendo que **Math.sqrt(x)** calcula la raíz cuadrada de "x".

La "salida" ha de ser de la siguiente forma:

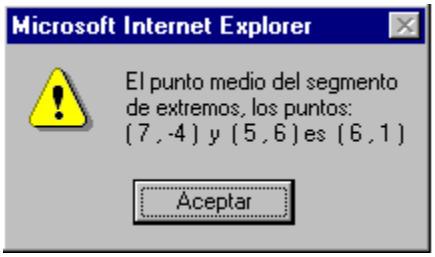
Comprueba el funcionamiento del programa, utilizando el caso concreto que aparece en la ilustración.



38) Haz un programa de nombre **Eval1q.htm** que sirva para calcular el punto medio de un segmento.

La "salida" ha de ser de la siguiente forma:

Comprueba el funcionamiento del programa, utilizando el caso concreto que aparece en la ilustración.



II.- Estructuras de Programación

1.- Operadores Lógicos y Relacionales

>, <, <=, >=	
==	**igualdad**
!=	**diferente**
&&	**y**
||	**o**
!	**No**

2.- La estructura "if-else"

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG013.HTM
var num;
num=parseFloat(prompt("Escribe un número",""));
if(num==100)
    {
    alert("El número que has escrito es 100");
    }
else
    {
    alert("El número que has escrito no es 100");
    }
if(num>0)
    {
    alert("El número que has escrito es positivo");
    }
else
    {
    alert("El número es negativo o 0");
    }
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog013.htm** en *TuCarpeta*

- Ejecútalo.

- **La Estructura de programación "if-else"**

Sintaxis:

```
if(condición)
{
sentencia1;
sentencia2;
sentencia3;
}
else
{
sentencia4;
sentencia5;
sentencia6;
}
```

Significado:
Si se cumple la condición, se ejecutan las sentencias 1, 2 y 3 y si no se cumple se ejecutan las sentencias 4, 5, 6. La opción "else" es opcional.

- Observa la diferencia entre "=" y "=="

a = 3*9 es una asignación, es decir la variable "a" es 27. En cambio **if(a==5)** es una condición: si "a" es idéntico a 5 (si el valor de "a" es el número 5)…

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG014.HTM

var a,b;
a=parseFloat(prompt("Escribe el primer número",""));
b=parseFloat(prompt("Escribe el segundo número",""));
if(a==b) alert("Los dos números son iguales");
if(a != b) alert("Los dos números son distintos");
if(a>b)
    {
    alert("El primer número es mayor que el segundo");
    }
else
    {
    alert("El primer número no es mayor que el segundo");
    }
if((a>b) && (100>a))
    {
    alert("El primero es mayor que el segundo");
    alert("Además los dos son menores de 100");
    }
else
    {
    alert("El primero no es mayor que el segundo");
    alert("O uno de los dos números es mayor o igual a 100");
    }
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog014.htm** en *TuCarpeta*.

- Ejecuta el programa para los siguientes casos, observando detenidamente lo que aparece:
    1) a=70, b=2
    2) a=50, b=30
    3) a=7, b=11
    4) a=100, b=50
    5) a=50, b=100

- Observa:
    a==b "a" es igual a "b"
    a != b "a" es diferente a "b"
    (a>b) && (100>a)
    "a" es mayor que "b" y además "100 es mayor que –a-"
    El **else** correspondiente a la condición anterior, sería equivalente a **no(a>b) || no(100>a)**, es decir la negación de la primera condición **o** la negación de la segunda (o las dos negaciones a la vez).

3.- La estructura de programación "while"

Sintaxis:

```
while(condición)
{
setencia1;
sentencia2;
sentencia3;
}
```

Significado:

"Mientras" se cumpla la condición, se irán repitiendo las sentencias 1, 2 y 3.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG015.HTM
var x=0;
while(x<6)
{
alert("El valor de x es= "+x);
x=x+1;
}

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog015.htm** en *TuCarpeta* y ejecútalo.

- **Estudio del Prog015.htm**
  - En "palabras", nuestro programa nos dice: "mientras" la x sea inferior a 6;
    escribe el valor de "x";
    incrementa en una unidad el valor de "x";
  - Al principio **x=0**
    Se cumple la condición del while: x<6
    Aparece escrito x=0
    Al pasar por el contador x=1
    Se cumple la condición x<6
    Aparece escrito x=1
    Al pasar por el contador x=2
    Aparece escrito x=2
    …
    …
    Cuando x=6 no se cumple la condición y por lo tanto se acaba el programa.

4.- Contadores en JavaScript

a=a+1	es equivalente a escribir **a++**
a=a-1	es equivalente a escribir **a—**
num=num+2	es equivalente a escribir **num += 2**
num=num*2	es equivalente a escribir **num *= 2**
num=num/2	es equivalente a escribir **num /= 2**

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG016.HTM
var i=2;
var res="";
var j=7;
while (i<j)
        {
        res=res+" "+i+" "+j+"\n";
        i++;
        j--;
        }
alert(res);

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog016.htm** en *TuCarpeta*

- Ejecuta el programa, es importante que tengas claro el funcionamiento del **Prog016.htm**: compara el listado del programa con lo que sucede al ejecutarlo. Observa cómo conseguimos escribir **toda la salida en un único "alert"** (variable "res").

Programa que repite un texto cualquiera, el número de veces que queramos, utilizando un "while"

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG017.HTM

var nom;
var indice=1;
var num;
var respuesta=" ";
nom=prompt("Escribe tu nombre","");
num=prompt("Cuántas veces quieres que lo repita","");
num=parseInt(num,10); // era una cadena y ahora es un número
while (indice <= num)
  {
  respuesta=respuesta+nom+"\n";
  indice++;
  }
alert(respuesta);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog017.htm** y ejecútalo

- Recuerda que en el "ParseInt" no es necesario escribir el 10, ya que por defecto (si no lo escribimos), es base 10.

Programa que acumula la suma y el producto de los números que queramos

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG018.HTM
var num;
var sum=0;
var pro=1;
var respuesta="";
num=prompt("Escribe un número diferente de cero=","");
num=parseFloat(num);
while(num != 0)
        {
        sum=sum+num;
        pro=pro*num;
        respuesta=respuesta+num+"\tsuma parcial:"+sum+"\tproducto parcial:"+pro+"\n";
        num=prompt("Escribe otro número (para acabar introduce cero)","");
        num=parseFloat(num);
        }
alert(respuesta);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog018.htm** y ejecútalo.

- Observa como acumulamos la suma y el producto:
  Contador que acumula la suma de los números "num": **sum=sum+num;**
  Hemos de inicializar a 0 la variable "sum".
  Contador que acumula el producto de los números "num": **pro=pro*num**
  Hemos de inicializar a 1 la variable "pro".

5.- La Estructura de programación "For"

```
for (contador = valor inicial; condición; expresión de incremento)
        {
        …;
        …;
        …;
        }
```

Ejemplo:

```
        for (i=1;i<=10;i++)
                {
                sentencia1;
                sentencia2;
                sentencia3;
                }
```

En palabras significa:
"Desde i=1, hasta i=10 de 1 en 1, repite las sentencias 1, 2 y 3"
Es decir: Repite 10 veces las sentencias 1, 2 y 3

Observa que el contador o índice del "for" (en nuestro caso la "i"), es una variable que no es necesario declararla, ya que la expresión **"i=1"** la declara e inicializa.

Programa que repite un texto cualquiera en número de veces que queramos, utilizando un "for"

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG019.HTM

var texto;
var num;
var salida="";
texto=prompt("Escribe un texto","");
num=prompt("Cuántas veces quieres que lo repita","");
num=parseInt(num,10);
for(i=1;i<=num;i++)
        {
        salida=salida+texto+"\n";
        }
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el fichero en *TuCarpeta*, con el nombre **Prog019.htm** y ejecútalo unas cuántas veces.

Programa que calcula todos los múltiplos de 11 menores de 3000 y por último nos da la suma de todos ellos.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG020.HTM

var salida="";
var sum=0;
for(multi=11;multi<3000;multi=multi+11)
      {
      salida=salida+multi+" ";
      sum=sum+multi;
      }
alert(salida+"\nLa Suma de todos ellos es= "+sum);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta*, con el nombre **Prog020.htm** y ejecútalo.

Programa que calcula el factorial de un número

Recuerda que el factorial del número "x" es: 1*2*3*4*5*…*x

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG021.HTM
var salida="";
var fact=1;
var num;
num=prompt("Cálculo del factorial del numero ","");
num=parseInt(num,10);
for(i=1;i<=num;i++) fact=fact*i;
alert("El factorial de "+num+" es "+fact);


</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog021.htm** y ejecútalo varias veces.

  Observa que nuestro "for" no necesita llaves, porque contiene una única sentencia.

Programa que calcula los 10 primeros múltiplos del número que queramos, por último nos da la suma de todos ellos.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG022.HTM
var salida="";
var num;
var mult;
var sum=0;
num=prompt("¿Múltiplos de qué número?","");
num=parseInt(num,10);
for(i=1;i<=10;i++)
  {
  mult=num*i;
  salida=salida+mult+" ";
  sum=sum+mult;
  }
alert(salida+"\nSuma= "+sum);


</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog022.htm** en *TuCarpeta* y ejecútalo varias veces.

Tabla de valores de la función $y=x^2-5x+10$

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG023.HTM

var x1,x2,paso;
var salida="";
var y;
x1=prompt("Escribe el menor valor de x","");
x1=parseFloat(x1);
x2=prompt("Escribe el mayor valor de x","");
x2=parseFloat(x2);
paso=prompt("Escribe el incremento de x:","");
paso=parseFloat(paso);
for(i=x1;i<=x2;i=i+paso)
        {
        y=i*i-5*i+10;
        salida=salida+i+" "+y+"\n";
        }
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el fichero en *TuCarpeta* con el nombre **Prog023.htm** y ejecútalo varias veces.

- Al ejecutar el programa anterior, nos podemos encontrar con una serie de problemas, por ejemplo si introducimos en el valor menor de "x" (x1), un valor que sea mayor que el introducido en la variable x2, o también puede suceder que en la variable **paso** escribamos un número negativo.

- Vamos a solucionar estos posibles problemas, es decir vamos a "mejorar" el programa anterior. Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG024.HTM

var x1,x2,paso;
var salida="";
var y;
x1=prompt("Escribe el menor valor de x de la tabla","");
x1=parseFloat(x1);
x2=prompt("Escribe el mayor valor de x de la tabla","");
x2=parseFloat(x2);
if (x1>=x2)
  {
   alert("No tiene sentido lo que intentas hacer");
  }
else
  {
```

```
paso=prompt("Escribe el incremento de x","");
paso=parseFloat(paso);
if (paso<=0)
   {
   alert("No tine sentido lo que intentas hacer");
   }
else
   {
   for (i=x1;i<=x2;i=i+paso)
       {
       y=i*i-5*i+10;
       salida=salida+i+" "+y+"\n";
       }
   alert(salida);
   }
}

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog024.htm** en *TuCarpeta* y ejecútalo, convendría probar los casos **x1>=x2 y paso<0**

- Observa de qué forma salimos del programa si introducimos datos que no tienen sentido.

6.- El Objeto "Math"

Nos permite trabajar con funciones matemáticas.
Concretamente:

Math.log(x) = ln(x)
Math.exp(x) = e^x
Math.sqrt(x) = raiz cuadrada de "x"
Math.pow(a, b) = a^b
Math.floor(): número entero más cercano y menor
Math.ceil(): número entero más cercano y mayor
Math.round(): redondea al entero más próximo.
Math.random(): número aleatorio entre 0 y 1
Math.round(y-x)*Math.random()+x: número aleatorio entre "x" e "y".
Math.sin(x)= sin(x) x en radianes
Math.cos(x)= cos(x) x en radianes
Math.tan(x)= tg(x) x en radianes
Math.atan(x)= arctg(x) resultado en radianes
Math.abs(x): valor absoluto de "x"
Math.max(a,b) : máximo valor de los dos
Math.min(a,b): mínimo valor de los dos.

Programa que calcula la hipotenusa de un triángulo rectángulo

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG025.HTM

var cat1,cat2,hipo;
```

```
cat1=prompt("Escribe el valor de un cateto","");
cat2=prompt("Escribe el valor del otro cateto","");
cat1=parseFloat(cat1);
cat2=parseFloat(cat2);
hipo=Math.sqrt(cat1*cat1+cat2*cat2);
alert("La hipotenusa del triángulo de catetos "+cat1+" y "+cat2+" es "+hipo);


</SCRIPT>
</HTML>
```

- Graba el fichero con el nombre **Prog025.htm** en *TuCarpeta* y ejecútalo unas cuantas veces.

Programa que calcula tantas hipotenusas como queramos

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG026.HTM

var opcion="S";
var cat1,cat2,hipo;
while(opcion=="S" || opcion=="s")
        {
        cat1=prompt("Escribe el valor de un cateto","");
        cat2=prompt("Escribe el valor del otro cateto","");
        cat1=parseFloat(cat1);
        cat2=parseFloat(cat2);
        hipo=Math.sqrt(cat1*cat1+cat2*cat2);
        alert("La hipotenusa del triángulo de catetos "+cat1+" y "+cat2+" es "+hipo);
        opcion=prompt("¿Quieres calcular otra hipotenusa? (S/N)","");
        }
alert("Adiós muy buenas");


</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog026.htm** y ejecútalo.

Programa que resuelve una ecuación de segundo grado

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG027.HTM

var a,b,c;
var discri;
var x1,x2;
a=prompt("Escribe el coeficiente de la x^2","");
a=parseFloat(a);
b=prompt("Escribe el coeficiente de la x","");
```

```
b=parseFloat(b);
c=prompt("Escribe el término independiente","");
c=parseFloat(c);
discri=b*b-4*a*c;
if(discri<0) alert("Soluciones Imaginarias");
if(discri==0)
      {
      x1=-b/(2*a);
      alert("Solución doble que es "+x1);
      }
if(discri>0)
      {
      x1=(-b+Math.sqrt(discri))/(2*a);
      x2=(-b-Math.sqrt(discri))/(2*a);
      alert("Las soluciones son = "+x1+" y "+x2);
      }

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog027.htm**

- Ejecútalo para los casos:
  a= 1, b= 1, c= 1
  a= 2, b=-6, c= -20
  a= 1, b= 4, c= 4

Programa que construye una tabla de senos

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG028.HTM

var gra,rad;
var salida="";
for(gra=0;gra<=360;gra=gra+10)
    {
    rad=3.141592*gra/180;
    salida=salida+gra+"\t\t"+Math.sin(rad)+"\n";
    }
alert(salida);


</SCRIPT>
</HTML>
```

- Graba el fichero con el nombre **Prog028.htm** en *TuCarpeta* y ejecútalo.

Programa que calcula el logaritmo en base cualquiera de un número dado

- Escribe:

```
<HTML>
```

```
<SCRIPT LANGUAGE="JavaScript">

// PROG029.HTM

var num;
var opc,base;
num=prompt("Escribe un número positivo","");
num=parseFloat(num);
opc=prompt("1 Logaritmo Neperiano. 2 Logaritmo Decimal. 3 Logaritmo en base a\nEscribe el
número de la opción","");
opc=parseInt(opc,10);
if(opc==1)
        {
        alert("El logaritmo Neperiano de "+num+" es "+Math.log(num));
        }
if(opc==2)
        {
        alert("El logaritmo Decimal de "+num+" es "+(Math.log(num)/Math.log(10)));
        }
if(opc==3)
        {
        base=prompt("Introduce el valor de la base a","");
        base=parseFloat(base);
        alert("El Logaritmo en base "+base+" del número "+num+" es
"+(Math.log(num)/Math.log(base)));
        }


</SCRIPT>
</HTML>
```

- Graba el fichero anterior con el nombre **Prog029.htm** en *TuCarpeta* y ejecútalo varias veces.

Programa que calcula "potencias"

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG029a.HTM

var bas,exp,resul
bas=parseFloat(prompt("base de la potencia?",""));
exp=parseFloat(prompt("escribe el exponente",""));
resul=Math.pow(bas,exp)
alert(bas+" elevado a "+exp+" es igual a "+resul)
</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog029a.htm**
- Utiliza el programa anterior para calcular las siguientes expresiones:
  2^3
  $0.32^{4.5291}$
  raíz cuadrada de 2
  raíz cúbica de 5.01527

Programa que calcula la raíz enésima de un número

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG029b.HTM

var ind,rad,resul
ind=parseFloat(prompt("índice de la raíz?",""));
rad=parseFloat(prompt("escribe el radicando",""));
resul=Math.pow(rad,1/ind)
alert("La raíz "+ind+" de "+rad+" es igual a "+resul)
</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog029b.htm**
- Utiliza el programa anterior para calcular las siguientes expresiones:
  raíz 5-ésima de 32
  raíz 7-ésima de 4.7201
  raíz 0.5-ésima de 2

Programa que calcula el número "e"

Recuerda que "e" es el límite de la expresión: $(1+1/n)^n$, cuando n tiende a infinito.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG029c.HTM

var cad;
var salida="";
for(i=1;i<=5;i=i+1)
      {
      cad="n= "+i+"\t\te= "+Math.pow(1+1/i,i)+"\n";
      salida=salida+cad;
      }
for(i=100;i<=50000000000;i=i*10)
      {
      cad="n= "+i+"\t\te= "+Math.pow(1+1/i,i)+"\n";
      salida=salida+cad;
      }
salida=salida+"Verdadero valor de e ="+Math.E
alert(salida);
</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog029c.htm** y ejecútalo.
- Observa que **Math.E** nos da el número "e" con la máxima precisión que admite el JavaScript.

Números Aleatorios

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG029d.HTM

var a;
var x,y;
var salida="";
salida=salida+"5 números aleatorios entre 0 y 1\n";
for(i=1;i<=5;i++)
        {
        salida=salida+Math.random()+"\n";
        }
salida=salida+"\n\n5 números aleatorios entre 3 y 7\n";
for(i=1;i<=5;i++)
        {
        salida=salida+(Math.round(7-3)*Math.random()+3)+"\n";
        }
salida=salida+"\n\n5 números aleatorios entre 15 y 70\n";
for(i=1;i<=5;i++)
        {
        salida=salida+(Math.round(70-15)*Math.random()+15)+"\n";
        }
alert(salida);
alert("Vamos a ver 5 números aleatorios entre los dos\nque tú quieras");
x=parseFloat(prompt("Escribe el número menor (puede ser decimal)",""));
y=parseFloat(prompt("Escribe el número mayor (puede ser decimal)",""));

salida="";
salida=salida +"5 números aleatorios entre "+x+" y "+y+"\n\n\n";
for(i=1;i<=5;i++)
        {
        a=Math.round(y-x)*Math.random()+x;
        salida=salida+a+"\n";
        }
alert(salida);
/* Si quieres números aleatorios enteros basta
   cambiar los paréntesis de la siguiente forma:
          Math.round((y-x)*Math.random()+x) */
salida="";
salida=salida+"150 números enteros aleatorios entre 2 y 17\n";
for(i=1;i<=150;i++)
        {
        salida=salida+Math.round((17-2)*Math.random()+2)+" - ";
        }
alert(salida);
</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog029d.htm** y ejecútalo.

Adivinanzas

Vamos a hacer un programa que nos pregunte un número entero del 1 al 10, y el usuario del programa tiene 5 tentativas para adivinarlo.

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG029e.HTM

var x,num;
var i=0;
var control=0;
x=Math.round(9*Math.random()+1);

while(i<5)
        {
        i++;
        num=parseInt(prompt("Escribe un entero entre 1 y 10, intento "+i,""));
        if(num==x)
                {
                alert("Lo has acertado en "+i+" tentativas");
                i=5;
                control=1;
                }
        }
if(control==0)
        {
        alert("Lo siento pero se han acabado tus 'vidas', el número era "+x);
        }
</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog029e.htm** y ejecútalo varias veces.
- Observa la utilidad de las variables **control** y **i**. Próximamente veremos una forma más elegante de salir de una estructura **while** (o **for**).

Programa que nos pregunta 5 sumas aleatoriamente y al final nos da la "nota"

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG029f.HTM

var x,y,z;
var nota=0;
for(i=1;i<=5;i++)
        {
        x=Math.round(9*Math.random()+1);
        y=Math.round(9*Math.random()+1);
        z=parseInt(prompt(x+" + "+y+" = ",""));
        if(z==x+y)
                {
                alert("Muy bien");
```

```
                nota=nota+1;
                }
        else
                {
                alert("Lo siento, pero "+x+" + "+y+" = "+(x+y));
                }
        }
alert("Tu nota es "+(2*nota));
</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog029f.htm** y ejecútalo varias veces.

7.- Las sentencias BREAK y CONTINUE

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG030.HTM

var salida="";
for (x=5;x<15;x++)
    {
     if (x==8) break;
     salida=salida+x+" \n";
    }
salida=salida+"\n\n";
for (x=5;x<15;x++)
    {
    if (x==8) continue;
    salida=salida+x+"\t";
    }
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog030.htm**

- Ejecútalo varias veces, observando detenidamente lo que sucede

- **Estudio del Prog030.htm**
  La sentencia **break** nos obliga a salir del ciclo "for". Por esta razón el primer "for" del programa sólo escribe del 5 hasta el 7
  La sentencia **continue** salta al final del ciclo "for" y continua ejecutando el ciclo. Por esta razón el segundo ciclo "for" escribe todos los números del 5 al 14, exceptuando el 8.
  Las sentencias **break** y **continue** funcionan exactamente igual en las estructuras **while** y **"Do-while"** (esta última estructura la veremos próximamente).

- Corrige el **Prog029e.htm** de la siguiente forma:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG030a.HTM

var x,num;
var i=0;
x=Math.round(9*Math.random()+1);

while(i<5)
    {
    i++;
    num=parseInt(prompt("Escribe un entero entre 1 y 10, intento "+i,""));
    if(num==x)
            {
            alert("Lo has acertado en "+i+" tentativas");
            break;
            }
    }
if(i==5)
    {
    alert("Lo siento pero se han acabado tus 'vidas', el número era "+x);
    }
</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog030a.htm** y ejecútalo varias veces para comprobar que funciona correctamente.

- Compara el **Prog029e** con el **Prog030a**. Observa el uso práctico del "break".

8.- La Estructura de programación “switch-case”

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG031.HTM

var opc="0";
var num;

while (opc != "10")
{
opc=prompt("Escribe la opción que desees: (1)El Triple-(2)El Cuadrado-(3)El Logaritmo
Neperiano(4)El Seno-(5)El Coseno-(10)SALIR","");

switch(opc)
        {
        case "1":
```

```
                num=prompt("Escribe el número","");
                num=parseFloat(num);
                alert("El triple de "+ num +" es " +(3*num));
                break;
        case "2":
                num=prompt("Escribe el número","");
                num=parseFloat(num);
                alert("El cuadrado de "+ num +" es " +(num*num));
                break;
        case "3":
                num=prompt("Escribe el número","");
                num=parseFloat(num);
                alert("El Logaritmo Neperiano de "+ num +" es " +(Math.log(num)));
                break;
        case "4":
                num=prompt("Escribe el ángulo en radianes","");
                num=parseFloat(num);
                alert("El seno de "+ num +" es " +Math.sin(num));
                break;
        case "5":
                num=prompt("Escribe el ángulo en radianes","");
                num=parseFloat(num);
                alert("El coseno de "+ num +" es " +Math.cos(num));
                break;
        }
}
</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog031.htm** y ejecútalo.

- **La estructura "switch-case"**

```
switch(x)
{
case valor 1:
        sentencia1;
        sentencia2;
        …;
        …;
        break;
case valor 2:
        sentencia3;
        …;
        …;
        break;
…
…
…
}
```

Según el valor que tome la variable "x", se ejecutarán las líneas de programa del "case" correspondientes. Observa que cada "case" termina con "break".

Programa que nos da la "nota" cualitativa a partir de la cuantitativa

- El programa nos pide el número total de preguntas y el número de respuestas acertadas. A partir de aquí y utilizando la estructura "switch-case", el programa nos da la "nota" cualitativa.

- En efecto, escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG031a.HTM
var num,bien,notanum,notacual;
num=parseInt(prompt("Escribe el número total de preguntas",""));
bien=parseInt(prompt("Escribe el número de resouestas acertadas",""));
notanum=parseInt(10*bien/num);
switch(notanum)
    {
    case 0:
            notacual="Muy Deficiente";
            break;
    case 1:
            notacual="Muy Deficiente";
            break;
    case 2:
            notacual="Deficiente";
            break;
    case 3:
            notacual="Deficiente";
            break;
    case 4:
            notacual="Insuficiente";
            break;
    case 5:
            notacual="Suficiente";
            break;
    case 6:
            notacual="Bien";
            break;
    case 7:
            notacual="Notable";
            break;
    case 8:
            notacual="Notable";
            break;
    case 9:
            notacual="Excelente";
            break;
    case 10:
            notacual="Matricula de Honor";
            break;
    }
alert("La nota cualitativa es "+notacual);
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog031a.htm** en *TuCarpeta* y ejecútalo varias veces para comprobar que funciona.

9.- La Estructura de programación "Do-while"

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG032.HTM

var x=1;
var salida="while:\n";

while (x<5)
{
x=x+1;
salida=salida+x+"\t";
}
salida=salida+"\ndo while:\n";
x=1;

do
{
x=x+1;
salida=salida+x+"\t";
} while (x<5);
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog032.htm** y ejecútalo.

¿Qué diferencia hay entre la estructura **while** y la **do-while**?

- Estructura de programación **do-while**:

```
Do
{
sentencia1;
sentencia2;
sentencia3;
} while(condición);
```

Mientras se cumpla la **condición**, se repetirá la ejecución de las sentencias 1, 2 y 3.
Como la evaluación de la **condición** se efectúa al acabarse el ciclo, el **do-while** se ejecutará siempre como mínimo una vez. Ésta es la diferencia que hay entre la estructura **do-while** y la **while**.

- Para comprobarlo escribe el siguiente programa (aprovecha el Prog032.htm, porque prácticamente es el mismo):

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG033.HTM

var x=1;
var salida="while:\n";

while (x<5)
{
x=x+1;
salida=salida+x+"\t";
```

```
}
salida=salida+"\ndo while:\n";
x=1;

do
{
x=x+1;
salida=salida+x+"\t";
} while (x>5);
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog033.htm** y ejecútalo varias veces, comparándolo con el **Prog032.htm**

10.- Cálculo en forma ITERATIVA

- La llamada **sucesión de FIBONACCI** es: 0, 1, 1, 2, 3, 5, 8, 13, …
  Es decir, cada término es igual a la suma de los dos anteriores.
  Vamos a "programar" la sucesión de Fibonacci.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG034.HTM

var anterior,ultimo,aux;
anterior=0;
ultimo=1;
var solucion;
solucion="0 - 1";

while (ultimo<=25000000000000)
{
aux=anterior+ultimo;
anterior=ultimo;
ultimo=aux;
if (ultimo>0) solucion=solucion+" - "+ultimo;
}
alert(solucion);

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog034.htm** y ejecútalo.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG035.HTM
```

```
// Cálculo de factoriales

var fin;
var num=2;
var fact;
var solucion="";
fin=prompt("Factoriales hasta el número?","");
fin=parseFloat(fin);
while (num<=fin)
{
fact=1;
for (i=1;i<=num;i++)
      {
      fact=fact*i;
      }
solucion=solucion+" - "+fact;
num++;
}
alert(solucion);

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog035.htm** y ejecútalo.

11.- Variable Auxiliar

En el programa **Prog034** ya utilizábamos una variable auxiliar, vamos a hacer otro programa que la utilize.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG036.HTM

/* Programa que lee la longitud de los 3 lados de un
triángulo y analiza qué tipo de triángulo es: no es triángulo,
equilátero, isósceles, escaleno, rectángulo */

var lado1,lado2,lado3,auxiliar;
var solucion="";
lado1=prompt("Primer lado: ","");
lado1=parseFloat(lado1);
lado2=prompt("Segundo lado: ","");
lado2=parseFloat(lado2);
lado3=prompt("Tercer lado: ","");
lado3=parseFloat(lado3);

// Toma nota del uso de la variable auxiliar
if (lado1>lado2)
        {
        auxiliar=lado1;
        lado1=lado2;
        lado2=auxiliar;
        }
if (lado3<lado1)
```

```
            {
            auxiliar=lado3;
            lado3=lado2;
            lado2=lado1;
            lado1=auxiliar;
            }
if (lado3<lado2)
            {
            auxiliar=lado2;
            lado2=lado3;
            lado3=auxiliar;
            }

/* Vamos a ver si la ordenación es la correcta */
solucion=solucion+lado1+"\t"+lado2+"\t"+lado3+"\n";

/* Clasificación del triángulo */
if (lado3>=lado1+lado2)
            {
            solucion=solucion+"Esto no es un triángulo";
            }
else
            {
            if ((lado1==lado2)&&(lado2==lado3))
                        {
                        solucion=solucion+"Triángulo Equilátero";
                        }
            else
                        {
                        if ((lado1==lado2)||(lado1==lado3)||(lado2==lado3))
                                    {
                                    solucion=solucion+"Triángulo Isósceles";
                                    }
                        else
                                    {
                                    solucion=solucion+"Triángulo Escaleno";
                                    }
                        if ((lado3<lado1+lado2)&&(lado3*lado3==lado1*lado1+lado2*lado2))
                                    {
                                    solucion=solucion+" además Rectángulo";
                                    }
                        }
}
alert(solucion);

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog036.htm** en *TuCarpeta*, y ejecútalo varias veces.

Autoevaluación II

1) El siguiente programa tiene errores. Escríbelo y corrígelo para que funcione (grábalo con el nombre **Eval2A.htm** en *TuCarpeta*):

```
<SCRIPT>
<HTML LANGUAGE="JavaScript"
// EVAL2A.HTM

var farenheit,celsius,
var s="";
for(i=-2;i<=12:i++)
      {
      celsius=10*i;
      farenheit=32+(celsius*9)/5;
      s=s+"C= "+celsius+" F= "+farenheit+"\n";
      if (celsius==0) s=s+"Punto congelación del Agua\n";
      if (celsius==100) s=s+"Punto de ebullición del Agua\n";
      }
alert(s;

<//SCRIPT>
<HTML>
```

2) Haz un programa que funcione de la siguiente forma:
- El programa nos pide que escribamos dos números positivos menores de 57
- El programa nos da como resultado el producto de los dos números
- Si los números no son positivos o son mayores de 57, el programa nos lo dice.
- El programa nos pregunta al final si queremos volver a empezar.

Graba el programa con el nombre **Eval2B.htm** en *TuCarpeta*

3) Escribe un programa que nos vaya pidiendo números. Si escribimos el número 9999 se acaba; por último el programa nos da como resultado el número de números introducidos, exceptuando el 9999.
Graba el programa con el nombre **Eval2C.htm** en *TuCarpeta*.

4) Haz un programa que haga lo mismo que el anterior, pero además nos dé la suma de todos los números introducidos, exceptuando el 9999
Graba el programa con el nombre **Eval2D.htm** en *TuCarpeta*.

5) Haz un programa que haga lo mismo que el anterior, pero además nos dé el producto de los números introducidos, exceptuando el 9999.
Graba el programa con el nombre **Eval2E.htm** en *TuCarpeta*.

6) Haz un programa que escriba todos los múltiplos de 23 inferiores a 1000 y por último nos dé la suma de todos ellos.
Graba el programa con el nombre **Eval2F.htm** en *TuCarpeta*.

7) Haz un programa que sirva para hacer una tabla de valores de la función **y=sen(7x-5)**
- El programa nos pide los dos valores de "x" (valores máximo y mínimo).
- El programa nos pide el incremento (variación) de la "x".

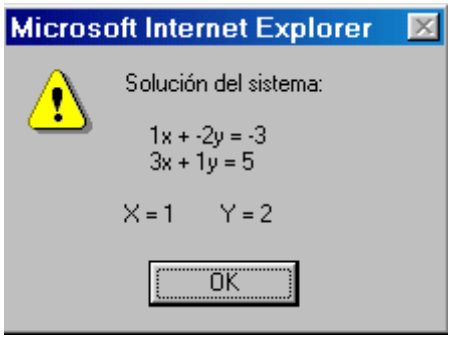
Graba el programa con el nombre **Eval2G.htm** en *TuCarpeta*.

8) Haz un programa que sirva para calcular un cateto de un triángulo rectángulo a partir del otro cateto y la hipotenusa, de la siguiente forma:
- El programa nos pide el valor de la hipotenusa.
- El programa nos pide el valor de un cateto.
- Si el cateto es mayor que la hipotenusa, el programa nos da un mensaje de error y se acaba.
- El programa nos da como resultado el valor del otro cateto y nos pregunta si queremos volver a empezar.

Graba el programa con el nombre **Eval2H.htm** en *TuCarpeta*.

9) Haz un programa que sirva para resolver ecuaciones de 2º grado del tipo $ax^2 + bx = 0$.
Graba el programa con el nombre **Eval2I.htm** en *TuCarpeta*.
(Modifica el **Prog027.htm**, que resolvia el caso general)

10) Haz un programa que sirva para resolver sistemas de ecuaciones del tipo: $ax + by = c$, $dx + ey = f$
Graba el programa con el nombre **Eval2J.htm** en *TuCarpeta*.
x=(ce-bf)/(ae-bd) ; y=(af-dc)/(ae-bd)
Prueba el funcionamiento del programa para el caso a=1; b=-2; c=-3; d=3; e=1;f=5; si todo funciona correctamente: x=1 ; y=2



11) Haz un programa con la posibilidad de hacer el **Eval2I.htm** o el **Eval2J.htm** (debes utilizar la estructura "switch")
Graba el programa con el nombre **Eval2K.htm** en *TuCarpeta*.

12) Haz un programa que escriba la tabla de valores de la función $y = ax^2 + bx + c$, el programa nos pide los valores de a, b, c y el número natural "v".
El programa construye la tabla de valores entre los valores de "-v" y "v" con un incremento de una unidad.
Graba el programa con el nombre **Eval2L.htm** en *TuCarpeta*.
La "salida" para el caso a = 1, b = -2, c=3, v=5 ha de ser de la siguiente forma:

13) Haz un programa que escriba los 15 primeros múltiplos de 7, su suma y su producto. El programa ha de tener la posibilidad de volver a empezar.
Graba el programa con el nombre **Eval2M.htm** en *TuCarpeta*.

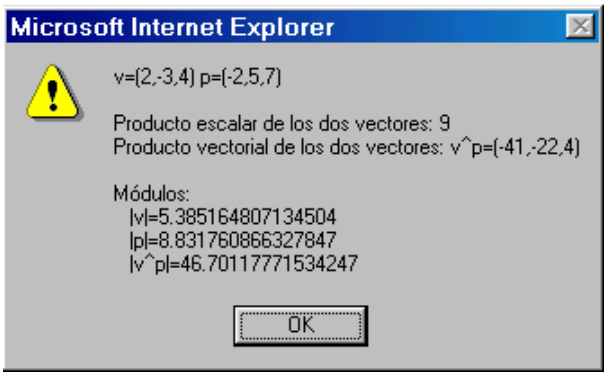
14) El siguiente programa tiene errores, indícalos y explica detalladamente lo que hace el programa:

```
<htmL>
<SCRITP>
// Eval2N.htm
var i=0;
var s="";
wile(i<5);
    {
    s=s+i
    i++;
    }
alert(s);
</SCRIPT>
<HTLM>
```

Graba el programa corregido con el nombre **Eval2N.htm** en *TuCarpeta*.

15) Haz un programa que sirva para calcular el área de un triángulo o el área de un rectángulo o el área de un círculo. El programa ha de tener la posibilidad de volver a empezar.
Graba el programa con el nombre **Eval2O.htm** en *TuCarpeta*.

16) Haz un programa tal que: dados dos vectores del espacio calcule su producto escalar, producto vectorial y además nos dé el módulo de los dos vectores y también el módulo del producto vectorial.
Graba el programa con el nombre **Eval2P.htm** en *TuCarpeta*.
v =(a , b , c) w=(d , e , f)
Producto Escalar = ad + be + cf
Producto vectorial = (bf-ec , dc-af , ae-bd)
Módulo de v = $\sqrt{a^2 + b^2 + c^2}$



17) Haz un programa que "dibuje" un rectángulo de asteriscos a partir de la base y la altura.
Graba el programa con el nombre **Eval2Q.htm** en *TuCarpeta*.

18) Haz un programa que dibuje un cuadrado, con el carácter que quieras, a partir del lado.
Graba el programa con el nombre **Eval2R.htm** en *TuCarpeta*.

19) Haz un programa que nos pida un número y dé como resultado la tabla de multiplicar del número introducido.
Graba el programa con el nombre **Eval2S.htm** en *TuCarpeta*.

20) Haz un programa que calcule el número "e" mediante el desarrollo en serie:
e = 1 + 1/(1!) + 1/(2!) + 1/(3!) + 1/(4!) + ... 1/(50!)
Graba el programa con el nombre **Eval2T.htm** en *TuCarpeta*.

21) Haz un programa que escriba 50 números aleatorios enteros entre 1 y 6.
Graba el programa con el nombre **Eval2U.htm** en *TuCarpeta*

22) En matemáticas no se puede dejar un resultado numérico sin racionalizar, ya que el resultado sin racionalizar tiene un error mayor.
Haz un programa para comprobar la afirmación anterior, concretamente para las fracciones:

$$\frac{3}{\sqrt{2}} \quad \text{y} \quad \frac{3\sqrt{2}}{2}$$

Graba el programa con el nombre **Eval2V.htm**.

23) Haz un programa que nos pregunte 10 multiplicaciones aleatoriamente y al final nos dé la nota cualitativa.
Graba el programa con el nombre **Eval2W.htm**

III.- Funciones y Arrays

1.- Funciones sin retorno de parámetro

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG039.HTM

alert("Programa que ahora llamará a una función");
mediageo();
alert("Se acabó lo que se daba");

/* ¡Atención!: a continuación tenemos la función */
function mediageo()
  {
  var a,b;
  a=prompt("Escribe un número","");
  a=parseFloat(a);
  b=prompt("Escribe otro número","");
  b=parseFloat(b);
  alert("La media geométrica de "+ a +" y "+ b +" es "+Math.sqrt(a*b));
  }

</SCRIPT>
</HTML>
```

- Graba el programa anterior con el nombre **Prog039.htm** en *TuCarpeta* y ejecútalo varias veces.

- **Estudio del Prog039.htm**

Estructura de una función:

```
function nombreFunción()
    {
    sentencia1;
    sentencia2;
    ...;
    ...;
    }
```

Se dice que la función es sin retorno de parámetros, porque no devuelve nada; para que la función devuelva un valor, debería haber la instrucción **return**.
Si una función no retorna nada se le denomina también con el nombre de **MÉTODO**.
Nuestra función **mediageo()**, calcula la media geometrica de dos números. Como no retorna nada, lo que hace en realidad la función es "agrupar" en un lugar determinado del fichero **HTML**, una serie de sentencias que por ellas mismas ya forman un programa, se dice también que es un **subprograma**.

```
<HTML>
<SCRIPT LAN....
// Programa Principal
sentencia1;

mediageo();      // llama a la función y se ejecuta en este lugar
```

```
sentencia2();
// Fin del programa Principal

function mediageo()
        {
        sentencia3;
        ...;
        ....;
        }

</SCRIPT>
</HTML>
```

Método o Subprograma = función sin retorno de parámetros

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG040.HTM

var opc="0";

while (opc != "T")
        {
        opc=prompt("Escribe la letra de la opción deseada: (S) Sumar - (R) Raíz Cuadrada - (L)
Logaritmo Neperiano - (A) Ayuda - (T) Terminar","");
        if (opc=="S") suma();
        if (opc=="R") raiz();
        if (opc=="L") logaritmo();
        if (opc=="A") ayuda();
        }

function suma()
        {
        var a,b;
        a=prompt("Escribe el primer sumando","");
        a=parseFloat(a);
        b=prompt("Escribe el segundo sumando","");
        b=parseFloat(b);
        alert("La suma de "+ a +" y "+ b +" es "+(a+b));
        }

function raiz()
        {
        var a;
        a=prompt("Escribe el radicando ","");
        a=parseFloat(a);
        alert("La raíz cuadrada de "+ a +" es "+Math.sqrt(a));
        }

function logaritmo()
        {
        var x;
        x=prompt("Escribe un número positivo","");
        x=parseFloat(x);
        alert("El logaritmo neperiano de "+ x +" es "+Math.log(x));
        }
```

```
function ayuda()
        {
        alert("Es bastante tonto que me pidas ayuda\npero aquí la tienes:\n\tPulsa S si quieres
sumar\n\tPulsa R para la raíz cuadrada\n\tPulsa L para el logaritmo neperiano\n\tPulsa A para
acceder a la ayuda\n\tPulsa T para acabar");
        }

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog040.htm** en *TuCarpeta* y ejecútalo unas cuantas veces.

2.- Funciones que retornan parámetros

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG041.HTM

/* Programa Principal */

var x,y,result;
x=prompt("Escribe el primer sumando","");
x=parseFloat(x);
y=prompt("Escribe el segundo sumando","");
y=parseFloat(y);

/* A continuación llamo a la función para calcular
el cuadrado de la suma de x e y */
result=SumaCuadrado(x,y);
alert("El cuadrado de la suma de "+ x +" y "+ y +" es "+result);
// Fin del programa principal

/* A continuación tengo la definición de la función */
function SumaCuadrado(a,b)
  {
  return (a*a+b*b+2*a*b);
  }

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog041.htm** y ejecútalo varias veces.

- Estudio del **Prog041.htm**

```
function SumaCuadrado(a,b)
  {
  return (a*a+b*b+2*a*b);
  }
```

Es una función con dos argumentos (a y b) que **retorna** un valor (parámetro), que en nuestro caso es el cuadrado de la suma de "a" y "b".
Una función que retorna parámetros, se caracteriza en que en su interior aparece la sentencia **return**, que permite devolver valores.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG042.HTM

var salida="";
salida=salida+"Enero tiene "+ diasmes(1) +" días\n";
salida=salida+"Febrero tiene "+ diasmes(2) +" días\n";
salida=salida+"Marzo tiene "+ diasmes(3) +" días\n";
salida=salida+"Abril tiene "+ diasmes(4) +" días\n";
salida=salida+"Mayo tiene "+ diasmes(5) +" días\n";
salida=salida+"Junio tiene "+ diasmes(6) +" días\n";
salida=salida+"Julio tiene "+ diasmes(7) +" días\n";
salida=salida+"Agosto tiene "+ diasmes(8) +" días\n";
salida=salida+"Septiembre tiene "+ diasmes(9) +" días\n";
salida=salida+"Octubre tiene "+ diasmes(10) +" días\n";
salida=salida+"Noviembre tiene "+ diasmes(11) +" días\n";
salida=salida+"Diciembre tiene "+ diasmes(12) +" días\n";
alert(salida);

/* Función "diasmes" */
function diasmes(mes)
    {
    var dias;
    switch(mes)
            {
            case 2:
                    dias=28;
                    break;
            case 1:
            case 3:
            case 5:
            case 7:
            case 8:
            case 10:
            case 12:
                    dias=31;
                    break;
            case 4:
            case 6:
            case 9:
            case 11:
                    dias=30;
                    break;
            }
    return dias;
    }

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog042.htm** en *TuCarpeta* y ejecútalo.
- Observa:

- El programa principal, con la sentencia **"diames(1)"** llama a la función **"diames"** y ésta devuelve el número 31.
- Observa la estructura **"switch"**: si el "case" 1, 3, 5, 7, 8, 10 no contienen nada, se ejecuta el siguiente "case" que sí contiene algo: en nuestro caso el "case 12", que da a la variable **"días"** el valor **31**.

Programa que determina si un número es primo

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG043.HTM

var num,resto;
num=prompt("Escribe un número entero","");
num=parseInt(num,10);
for (i=2;i<num-1;i++)
      {
      resto=num % i;
      if ((resto==0) && (num != 2))
            {
            alert(num+" no es primo");
            break;
            }
      }
alert("Si no ha aparecido un mensaje de que no es primo, entonces el número "+num+"
es primo");

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog043.htm** en *TuCarpeta* y ejecútalo varias veces.

- **El operador %:**
  num %i, nos da el resto de la división entre **num** y **i**.

Programa que determina si un número es primo, pero utilizando una función

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG044.HTM

var num;

num=prompt("Escribe un número entero","");
num=parseInt(num,10);
if (primo(num)=="S")
    {
    alert(num+" es primo");
    }
```

```
else
    {
    alert(num+"no es primo");
    }

function primo(x)
    {
    var resto;
    for (i=2;i<x-1;i++)
            {
            resto=x % i;
            if ((resto==0) && (x != 2))
                    {
                    return "N";
                    }
            }
    return "S";
    }

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog044.htm** en *TuCarpeta* y ejecútalo.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG045.HTM

var num,base;
num=prompt("Escribe un número positivo","");
num=parseFloat(num);
base=prompt("Escribe la base del logaritmo","");
base=parseFloat(base);
alert("El logaritmo de "+num+" en base "+base+" es "+logBase(num,base));

/* Función que calcula el logaritmo en cualquier base */
function logBase(x,a)
    {
    return Math.log(x)/Math.log(a);
    }

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog045.htm** y ejecútalo varias veces.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG046.HTM

var num,numDecimales;
num=prompt("Escribe un número decimal","");
```

```
num=parseFloat(num);
numDecimales=prompt("Escribe el número de decimales","");
numDecimales=parseInt(numDecimales,10);
alert("El número "+num+" con "+numDecimales+" decimales =
"+Aproxi(num,numDecimales));

function Aproxi(n,d)
        {
        cifra=1;
        for(i=1;i<=d;i++)
                {
                cifra *= 10;
/* recuerda que "cifra *= 10" es equivalente a "cifra=cifra*10" */
                }
        return Math.round(n*cifra)/cifra;
        }

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog046.htm** y ejecútalo varias veces.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG047.HTM

var num;
num=prompt("Escribe un número: ","");
num=parseFloat(num);
alert("El signo de "+num+" es "+Signo(num));

function Signo(x)
        {
        if(x != 0)
                {
                return x/Math.abs(x);
                }
        else
                {
                return 0;
                }
        }

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog047.htm** y ejecútalo.

Cálculo del máximo común divisor (MCD) por el método de Euclides

- Escribe el siguiente programa:

```
<HTML>
```

```
<SCRIPT LANGUAGE="JavaScript">

// PROG048.HTM

var x,y,aux,resto;

x=prompt("Escribe un número","");
x=parseInt(x,10);
y=prompt("Escribe el otro número","");
y=parseInt(y,10);

if (x<y)
      {
      aux=x;
      x=y;
      y=aux;
      }

if ((x % y)==0) resto=y;

while ((x % y) != 0)
      {
      resto=x%y;
      x=y;
      y=resto;
      }

alert("El MCD es "+resto);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog048.htm** y ejecútalo varias veces.:
  MCD (5, 25) = 5
  MCD (7, 3) = 1
  MCD (720, 300) =60

- Se trata de hacer el mismo programa, pero utilizando una función. Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG049.HTM

var x,y;
x=prompt("Escribe un número","");
x=parseInt(x,10);
y=prompt("Escribe otro número","");
y=parseInt(y,10);
alert("El MCD es: "+MCD(x,y));

function MCD(a,b)
{
var resto,aux;
if(a<b)
  {
  aux=a;
  a=b;
  b=aux;
  }
```

```
if ((a%b)==0) resto=b;
while((a%b) !=0)
  {
  resto=a%b;
  a=b;
  b=resto;
  }
return resto;
}

</SCRIPT>
</HTML>
```

- Graba el fichero en *TuCarpeta* con el nombre **Prog049.htm** y ejecútalo para comprobar que funciona.

Programa que calcula los 10 primeros múltiplos del número que queramos, utilizando una función que retorna parámetro.

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG049a.HTM
function mult10(x)
        {
        var salida="Múltiplos de "+x+"\n";
        var mult=0;
        for(i=1;i<=10;i++)
                {
                mult=mult+x;
                salida=salida+mult+" - ";
                }
        return salida;
        }
a=parseInt(prompt("Escribe un número entero",""));
alert(mult10(a));
</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog049a.htm** y ejecútalo.

- Observa:
  En primer lugar se encuentra la función y a continuación el programa, que en nuestro caso consta de 2 únicas instrucciones. En la práctica veremos que la o las funciones siempre estarán en el <HEAD> </HEAD> de la página HTML.

Programa que construye una tabla de valores de y=mx+n utilizando una función

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
```

```
// PROG049b.HTM
function recta(a,b)
        {
        salida="Tabla de valores de y = "+a+"x + "+b+"\n";
        for(i=-5;i<=5;i++)
                {
                y=a*i+b;
                salida=salida+"X = "+i+"\t"+"Y = "+y+"\n";
                }
        return salida;
        }
m=parseFloat(prompt("Tabla de valores de y = mx + n\nEscribe el valor de m",""));
n=parseFloat(prompt("Tabla de valores de y = mx + n\nEscribe el valor de n",""));
alert(recta(m,n));
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog049b.htm** y ejecútalo varias veces.

Programa que calcula la hipotenusa de un triángulo utilizando una función

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG049c.HTM
function hipot(c1,c2)
        {
        return Math.sqrt(c1*c1+c2*c2);
        }

var x,y;
x=parseFloat(prompt("Escribe el valor de un cateto",""));
y=parseFloat(prompt("Escribe el valor del otro cateto",""));
alert("La hipotenusa del triángulo de catetos "+x+" , "+y+" es = "+hipot(x,y));
</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog049c.htm** y ejecútalo varias veces para asegurarte que funciona.

Programa que resuelve una ecuación de segundo grado utilizando tres funciones

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG049d.HTM
function imagi()
        {
        return "Soluciones Imaginarias";
        }
function doble(a,b)
        {
```

```
        return -b/(2*a);
        }
function dos(a,b,disc)
        {
        var x1,x2;
        x1=(-b+Math.sqrt(disc))/(2*a);
        x2=(-b-Math.sqrt(disc))/(2*a);
        return "x1 = "+x1+" , "+"x2 = "+x2;
        }

var x,y,z,d;
var salida="";
x=parseFloat(prompt("Ecuación ax²+bx+c=0\nEscribe el valor de 'a'",""));
y=parseFloat(prompt("Ecuación ax²+bx+c=0\nEscribe el valor de 'b'",""));
z=parseFloat(prompt("Ecuación ax²+bx+c=0\nEscribe el valor de 'c'",""));
d=y*y-4*x*z;
if(d<0)
        {
        alert(imagi());
        }
if(d==0)
        {
        alert("Una solución, que es "+doble(x,y));
        }
if(d>0)
        {
        alert("Dos soluciones:\n"+dos(x,y,d));
        }

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog049d.htm** y ejecútalo varias veces (como mínimo una para cada uno de los tres casos).

Programa que calcula potencias y raices utilizando dos funciones

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG049e.HTM
function pot()
        {
        var b,e;
        var resul;
        b=parseFloat(prompt("Escribe la base de la potencia",""));
        e=parseFloat(prompt("Escribe el exponente",""));
        resul=b+" elevado a "+e+" es "+Math.pow(b,e);
        alert(resul);
        }
function raiz()
        {
        var b,e;
        var resul;
        b=parseFloat(prompt("Escribe el radicando de la raíz",""));
        e=parseFloat(prompt("Escribe el índice de la raíz",""));
        resul="La raíz "+e+"-ésima de "+b+" es "+Math.pow(b,1/e);
```

```
        alert(resul);
        }

var opc="s"
while(opc != "S")
{
opc=prompt("Escribe la opción que desees:\n(P)Calcular una potencia  (R)Calcular una
raíz  (S)Salir del programa","S");
switch(opc)
        {
        case "P":
        case "p":
                pot();
                break;
        case "R":
        case "r":
                raiz();
                break;
        }
}

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog049e.htm** y ejecútalo varias veces.

Programa que nos da la "nota" cualitativa a partir de la cuantitativa utilizando una función

- Escribe (aprovecha lo que ya has hecho en el **Prog031a.htm**):

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
// PROG049f.HTM
function cuali(notanum)
{
switch(notanum)
        {
        case 0:
        case 1:
                return "Muy Deficiente";
                break;
        case 2:
        case 3:
                return "Deficiente";
                break;
        case 4:
                return "Insuficiente";
                break;
        case 5:
                return "Suficiente";
                break;
        case 6:
                return "Bien";
                break;
        case 7:
        case 8:
```

```
                return "Notable";
                break;
        case 9:
                return "Excelente";
                break;
        case 10:
                 return "Matricula de Honor";
                break;
        }
}

var num,bien,nota,notacual;
num=parseInt(prompt("Escribe el número total de preguntas",""));
bien=parseInt(prompt("Escribe el número de resouestas acertadas",""));
nota=parseInt(10*bien/num);
alert("La nota cualitativa es "+cuali(nota));

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog049f.htm** y pruébalo varias veces para comprobar que funciona.

3.- Matrices y Arrays

Una matriz es un conjunto de elementos colocados de forma adyacente en la memoria de manera que nos podemos referir a ellos con un solo nombre común.

- Las matrices se pueden **clasificar** según su tamaño en:
  - **Matrices Estáticas:** tienen un tamaño fijo e inmutable.
  - **Matrices Dinámicas:** tienen un tamaño variable. En JavaSript las matrices siempre son dinámicas.

- Las matrices se pueden **clasificar** según sus dimensiones:
  - **Arrays:** son matrices de una dimensión.
  - **Matrices:** son matrices multidimensionales.

  En JavaScript, las matrices son siempre "Arrays", pero veremos que podemos "simular" matrices multidimensionales.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG050.HTM

var x=new Array();
var salida="";

for(i=0;i<=10;i++)
{
x[i]=2*i;
salida=salida+"x["+i+"]="+x[i]+"\t";
}
```

```
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog050.htm** en *TuCarpeta* y ejecútalo.

- **Estudio del Prog050.htm**
  - **var x=new Array();**

  Definimos la variable "x" como un "array" de un número indeterminado de elementos.
  - Los elementos de un "array" se indican: **NombreArray[índice]= valor**

  El índice= 0, 1, 2, 3, 4, ...
  En nuestro caso:
  X[0] será el primer valor del array
  X[1] será el segundo.
  Etc.
  - **x[i] = 2*i**

  Asignamos a cada elemento del array el doble de su índice, es decir: x[0]= 0; x[1]= 2; x[2]= 4; etc.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

//PROG051.HTM

var n;
var salida="";
n=prompt("Escribe el número de elementos del Array","");
n=parseInt(n);
var Vector=new Array(n);
for(i=0;i<n;i++)
    {
    Vector[i]=prompt("Introduce Valor","");
    salida=salida+Vector[i]+"\n";
    }
alert("Los valores de la matriz son:\n"+salida);

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog051.htm** en *TuCarpeta*.

- Observa de qué forma, podemos definir un array con un número variable de valores.

Programa que calcula la media aritmética de una serie indeterminada de valores

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
//PROG051a.HTM

var x=new Array();
var med;
```

```
var y=0;
var i=0;
var sum=0;
while(y != 9999)
        {
        y=parseFloat(prompt("Introduce un valor\npara acabar escribe 9999","9999"));
        x[i]=y;
        i++;
        }
x[i]=0;
for(j=0;j<i-1;j++)
        {
        sum=sum+x[j];
        }
med=sum/j;
alert("La media es "+med);

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog051a.htm** y ejecútalo varias veces para comprobar que funciona.

Programa que calcula la media aritmética de un número determinado de valores utilizando una función

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
//PROG051b.HTM
function media(n)
        {
        var x=new Array(n);
        var sum=0;
        for(i=0;i<n;i++)
                {
                num=parseFloat(prompt("Escribe un valor de la serie ("+(i+1)+"º):",""));
                x[i]=num;
                sum=sum+x[i];
                }
        return sum/n;
        }

var valores;
valores=parseInt(prompt("Escribe el número de elementos de la serie",""));
alert("La media de todos los valores es: "+media(valores));

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog051b.htm** y ejecútalo varias veces para comprobar que funciona.

- Mejora la "salida" del programa anterior de la siguiente forma:

```
<HTML>
```

```
<SCRIPT LANGUAGE="JavaScript">
//PROG051c.HTM
function media(n)
  {
  var x=new Array(n);
  var sum=0;
  var salida="";
  for(i=0;i<n;i++)
        {
        num=parseFloat(prompt("Escribe un valor de la serie ("+(i+1)+"º):",""));
        x[i]=num;
        salida=salida+x[i]+" - ";
        sum=sum+x[i];
        }
  salida=salida+"\nLa Media Aritmética de estos "+n+" números es "+(sum/n);
  return salida;
  }

var valores;
valores=parseInt(prompt("Escribe el número de elementos de la serie",""));
alert(media(valores));

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog051c.htm** y pruébalo.

Programa que calcula, dada una serie de 20 números, la media aritmética, las desviaciones respecto a la media, la desviación media, la varianza y la desviación típica.

Recuerda:
Dada la serie estadística: x1, x2, x3, ... , xn
Media aritmética: x = (x1+x2+x3+ ......)/n
Desviaciones respecto a la media: d1=|x1-x|, d2=|x2-x|, ...
Desviación media: es la media aritmética de las desviaciones
Varianza: es la media aritmética de las desviaciones al cuadrado.
Desviación típica: es la raíz cuadrada de la varianza.

- Escribe:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
//PROG051d.HTM

var x=new Array(20);
var sum=0;
var med;
var des=new Array(20);
var sum1=0;
var sum2=0;
var desmed,vari,dt;
var salida="VALORES:\n";
var salida1="DESVIACIONES RESPECTO A LA MEDIA:\n";

for(i=0;i<20;i++)
      {
```

```
        num=parseFloat(prompt("Escribe un valor de la serie ("+(i+1)+"º):",""));
        x[i]=num;
        salida=salida+x[i]+" - ";
        sum=sum+x[i];
        }
med=sum/20;

for(i=0;i<20;i++)
        {
        des[i]=Math.abs(x[i]-med);
        salida1=salida1+des[i]+" - ";
        sum1=sum1+des[i];
        sum2=sum2+(des[i]*des[i]);
        }

desmed=sum1/20;
vari=sum2/20;
dt=Math.sqrt(vari);
alert(salida+"\n"+"MEDIA ARITMÉTICA =
"+med+"\n"+salida1+"\n"+"DESVIACIÓN MEDIA ="+desmed+"\nVARIANZA
="+vari+"\nDESVIACIÓN TÍPICA = "+dt);

</SCRIPT>
</HTML>
```

- Grábalo con el nombre **Prog051d.htm** y ejecútalo varias veces para comprobar que funciona.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG052.HTM

alert(GeneraLetras());

function GeneraLetras()
{
var n;
n=parseInt(prompt("Cuántas letras quieres (1 a 26)",""));
if (isNaN(n)==true) return;
if (n<1) n=1;
if (n>28) n=26;
var Letras=new Array(n);
for (i=0;i<=Letras.length-1;i++)
        {
        Letras[i]=String.fromCharCode(65+i);
        }
return Letras;
}

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog052.htm** y ejecútalo varias veces.

- Estudio del **Prog052.htm**
  - **length**, es una propiedad de las matrices que nos indica el tamaño de éstas.

  En nuestro caso: **Letras.length** = número de elementos del array "Letras".
  - Si en una función escribimos la sentencia **return**, sin ningún parámetro, simplemente "salimos" de la función y se acaba el programa.
  - **String.fromCharCode**

  Lo estudiaremos próximamente, en detalle. Baste decir de momento que escribe un determinado carácter.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG053.HTM

var salida="";
var Datos=new Array("Pepito","Paquito","Felipe");
var Edad=new Array(57,15,26);
alert("La matriz Datos tiene "+Datos.length+" elementos");
alert("La matriz Edad tiene "+Edad.length+" elementos");
for (i=0;i<=2;i++)
        {
        salida=salida+Datos[i]+" - "+Edad[i]+"\n";
        }
alert(salida);

</SCRIPT>
</HTML>
```

- Graba el programa con el nombre **Prog053.htm** en *TuCarpeta* y ejecútalo.
- Observa que podemos inicializar un array igual que hacíamos con cualquier variable.

Matrices con varias dimensiones

JavaScript no soporta directamente matrices con varias dimensiones.
En "Java" o en "C++" definimos **matriz[5][5]** como una matriz de dos dimensiones: 5x5 = 25 elementos en total.
En **JavaScript** podemos "simular" esta matriz de la siguiente forma:

```
var Matriz2D=new Array(5);
for(i=0;i<=4;i++)
  {
  Matriz2D[i]=new Array(5);
  }
```

Con este código lo que conseguimos es crear una matriz con 5 elementos cada uno de los cuales es, a su vez, una matriz. El resultado es el deseado: una matriz con dos dimensiones.
Una vez definida una matriz multidimensional, podemos acceder a sus elementos de la siguiente forma:

Matriz2D[0][0] = primer elemento de la matriz.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">
```

```
// PROG054.HTM

var Matriz2D= new Array(5);
for (i=0;i<5;i++)
      {
      var Aux=new Array(5);
      for (j=0;j<5;j++)
            {
            Aux[j]="E"+i+j;
            }
      Aux[4] += "\n";
      Matriz2D[i]=Aux;
      }
alert(Matriz2D);

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog054.htm** y ejecútalo.

4.- Código ASCII/Unicode

De lo único que entiende un ordenador es de 0 y 1. Cuando se asigna una cadena de texto a una variable, ésta internamente, se guarda como una serie de números que identifican a los diferentes caracteres constituyentes de la misma. Esta codificación se llama **ASCII**.

El código ASCII asigna a cada carácter un número de 0 a 255, es decir 256 caracteres en total.
1 carácter = 8 bits = 1 byte = 8 ceros y unos.
0 y 1 en grupos de ocho: $2^8 = 256$

El código ASCII se ha quedado pequeño: 256 caracteres es insuficiente para representar los símbolos propios de muchos países.

La **ISO** (organización internacional para la normalización), propuso el código **UNICODE**:
1 carácter = 2 bytes = 16 ceros y unos
0 y 1 en grupos de 16: $2^{16} = 65.536$ caracteres distintos.

La mayor parte de los sistemas operativos modernos (por ejemplo el Windows), guardan internamente la representación de sus caracteres como valores **Unicode** no **Ascii**

Función **fromCharCode**: accedemos a los caracteres **Unicode**

```
var x = String.fromCharCode(75, 81,52);
alert x;
```

Aparecen los caracteres correspondientes a la codificación "unicode" de los números 75, 81 y 52.

Función **charCodeAt**, es la función inversa de la anterior, es decir a partir de una cadena devuelve el código correspondiente a la letra que se encuentre en la posición especificada:

```
var x = "JavaScript"
alert(x.charCodeAt(5));
```

Devolverá un 99, que corresponde a la "c".
Las cadenas igual que las matrices comienzan a numerarse en el cero.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG055.HTM

var x="PACO-paco";
var letras=new Array(9);
var salida="";
salida=salida+x+"\n";
for(i=0;i<9;i++)
    {
    salida=salida+x.charCodeAt(i)+"\t";
    letras[i]=x.charCodeAt(i);
    }
alert(salida);
alert(letras);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog055.htm** y ejecútalo.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG056.HTM

var x;
x=prompt("Escribe una palabra o frase","");
var letras=new Array();
var numeros=new Array();
for(i=0;i<x.length;i++)
    {
    numeros[i]=x.charCodeAt(i);
    letras[i]=String.fromCharCode(numeros[i]);
    }
alert(letras+"\n"+numeros);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog056.htm** y ejecútalo.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG057.HTM

var x;
```

```
var salida="";
var letras=new Array();
var numeros=new Array();
x=prompt("Escribe una palabra o frase","");
for(i=0;i<=x.length;i++)
        {
        numeros[i]=x.charCodeAt(i);
        letras[i]=String.fromCharCode(numeros[i]);
        salida=salida+letras[i]+" = "+numeros[i]+"\n";
        }
alert(salida);
alert(SumaASCII(x));

function SumaASCII(frase)
        {
        var aux=0;
        for(i=0;i<frase.length;i++)
                {
                aux += frase.charCodeAt(i);
                }
        return aux;
        }


</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog057.htm** y ejecútalo.
- Investiga si es verdad lo que dicen las "malas lenguas": "En el nombre del fundador y presidente de Microsoft hay el número de la bestia".

5.- Búsqueda de un carácter determinado.

El método **charAt** de **String** devuelve una cadena conteniendo el carácter situado en la posición especificada:

"cadena".charAt(4) = letra situada en el lugar 5

Ejemplo: función que determina la existencia del carácter @:

```
Function Busca(texto)
        {
        for(i=0;i<texto.length;i++)
                {
                if(texto.charAt(i)=="@") return true;
                }
        return false
        }
```

Si utilizamos la función anterior para determinar una dirección e-mail, no sería correcto ya que el símbolo @ al principio o final de la cadena, no correspondería a una dirección de correo electrónico; para este caso deberíamos modificar la función:

```
Function BuscaEmail(texto)
        {
        for(i=0;i<texto.length;i++)
                {
                if(texto.charAt(i)=="@" && i ¡= (texto.length-1) && (i ¡= 0))
return true;
```

```
                    }
                return false
                }
        alert(BuscaEmail("pepe@eso.es"));
```

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG058.HTM

/* Programa que determina el número de "a" o "A" que
   aparece en un texto */

var texto;
texto=prompt("Escribe el texto que quieras:","");
alert("Número de a y de A = "+BuscaLetra(texto));

function BuscaLetra(x)
    {
    var numero=0;
    for(i=0;i<x.length;i++)
            {
            if(x.charAt(i)=="a" || x.charAt(i)=="A") numero++;
            }
    return numero;
    }

</SCRIPT>
</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog058.htm** y ejecútalo varias veces.

6.- Temporizadores

Los temporizadores son objetos sin representación física que se encargan de ejecutar una tarea al cabo de cierto tiempo que se le debe indicar.

NombreTemp= setTimeout("nombrefunción()", tiempo);

Para detener el temporizador: **clearTimeout(NombreTemp);**

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG059.HTM
/* Texto animado */

var velocidad=200;
var letras;

function animar()
```

```
  {
  letras=new Array();
  var texto="Uso de cadenas con JavaScript para efectos dinámicos";
  for(i=0;i<texto.length;i++)
          {
          letras[i]=texto.charAt(i);
          }
  mueveLetras();
  }

var TextoAct="";
var n=-1;

function mueveLetras()
  {
  n++;
  TextoAct += letras[n];
  document.forms[0].TextoDinamico.value=TextoAct;
  if(n==letras.length-1)
          {
          n=-1;
          TextoAct="";
          }
  setTimeout("mueveLetras()",velocidad);
  }

</SCRIPT>
<BODY onload=animar();>

<FORM>
<INPUT TYPE="text" NAME="TextoDinamico" VALUE="" SIZE="120">
</FORM>

</BODY>

</HTML>
```

- Graba el programa en *TuCarpeta* con el nombre **Prog059.htm** y ejecútalo.

- Estudio del **Prog059.htm**
  - La frase "Uso de cadenas con JavaScript para efectos dinámicos", se colocará una letra cada 0,2 segundos.
  - La línea **<BODY onLoad= animar();>**, llamará a la rutina "animar()", cuando la página termine de cargarse.
  - **<FORM>**
    <INPUT TYPE="text" NAME="TextoDinamico" VALUE= "" SIZE="120">
    </FORM>
    Definimos un formulario (FORM), que contiene un cuadro de texto (INPUT TYPE="text") de nombre **TextoDinamico** que no contiene nada (VALUE="") y tamaño (SIZE="120")
    En el próximo capítulo lo estudiaremos con detalle.

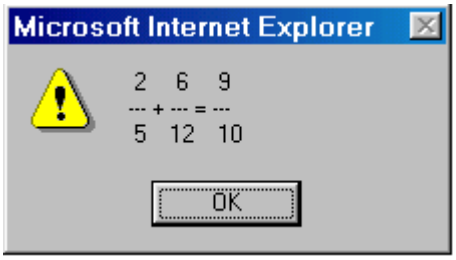
  - **document.forms[0].TextoDinamico.value= variable**;
    En el cuadro de texto de nombre "TextoDinamico", se coloca el valor de la variable.
    En el próximo capítulo lo estudiaremos con detalle.

Autoevaluación III

1) Haz un programa de nombre **Eval3A.htm**, que calcule el mínimo común múltiplo de dos números utilizando la función MCD del Prog049.htm y sabiendo que **mcm(x,y) = x*y/MCD(x,y)**

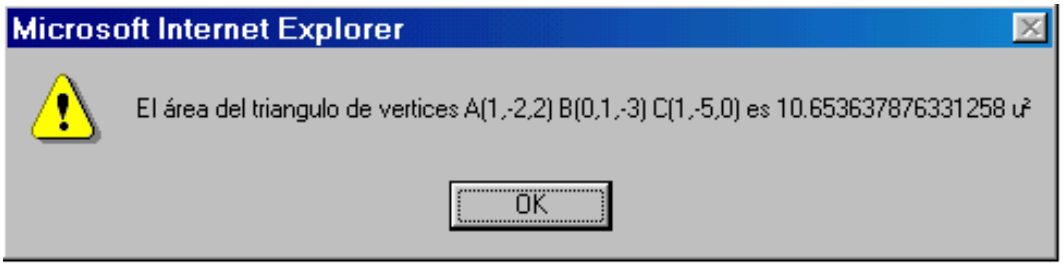
2) Haz un programa de nombre **Eval3B.htm**, que sirva para simplificar una fracción numérica, debes utilizar la función MCD del ejercicio anterior.
Observa:
$$\frac{a}{b} = \frac{a/MCD(a,b)}{b/MCD(a,b)}$$
Comprueba el funcionamiento del programa para el caso:

3) Haz un programa de nombre **Eval3C.htm**, que sirva para sumar o restar dos fracciones y después simplifique el resultado.
Observa:
$$\frac{a}{b}+\frac{c}{d}=\frac{a(mcm(b,d)/b)+c(mcm(b,d)/d)}{mcm(b,d)}$$
Compruébalo para el caso:



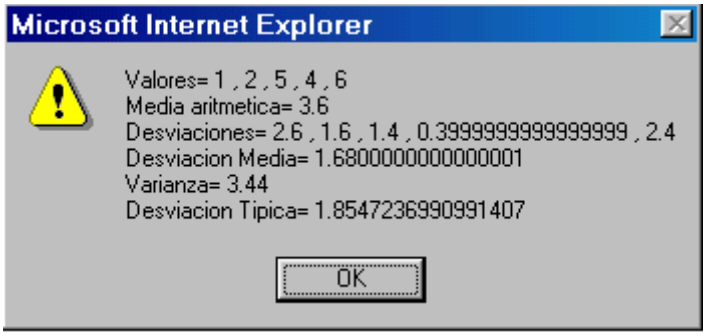
4) Haz un programa de nombre **Eval3D.htm**, que sirva para calcular el módulo de un vector en el espacio, utilizando una función.

5) Haz un programa de nombre **Eval3E.htm**, que sirva para calcular el área de un triángulo en el espacio, utilizando la función del ejercicio anterior.
Recuerda:
A=(a1,a2,a3), B=(b1,b2,b3), C=(c1,c2,c3)
AB=(b1-a1,b2-a2,b3-a3), **AC**=(c1-a1,c2-a2,c3-a3)
Área del triángulo ABC: mitad del producto vectorial de **AB** y **AC** (consulta el **Eval2P.htm**)
Compruébalo para el siguiente caso:

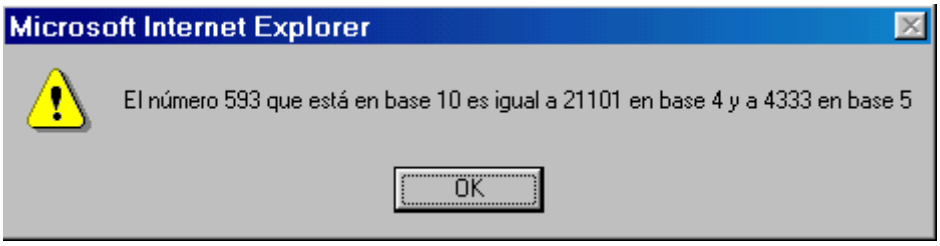


6) Haz un programa de nombre **Eval3F.htm**, que funcione de la siguiente forma:
- El programa nos pide 5 valores.
- El programa calcula la media aritmética (función con retorno de parámetro).
- El programa calcula las desviaciones respecto a la media
- El programa calcula la desviación media (llamada a la misma función de antes).
- El programa calcula la desviación típica (llamada a la misma función de antes).

Debes hacer el programa sin utilizar ningún array. Repasa el Prog051d.htm
Compruébalo para el caso:



7) Haz un programa de nombre **Eval3G.htm**, que transforma un número en base 10 a base 4 y 5. Repasa el primer capítulo.
Compruébalo para el caso:



8) Haz un programa de nombre **Eval3H.htm**, que construya el triángulo de Tartaglia o Pascal de la siguiente forma:
1º) Crea una **función** que calcule el factorial de un número (consulta el Prog021.htm)
2º) Crea otra **función** que permita calcular un número combinatorio
Recuerda:

$$\binom{m}{n} = \frac{m!}{n!(m-n)!}$$

Utiliza el **Prog021.htm** del segundo capítulo.
3º) El triángulo de **Tartaglia** no es más que los resultados de los números combinatorios:

$$\binom{0}{0}$$

$$\binom{1}{0}\binom{1}{1}$$

$$\binom{2}{0}\binom{2}{1}\binom{2}{2}$$

.....

4º) El programa nos ha de preguntar de entrada el número de filas del triángulo.

9) Haz un programa de nombre **Eval3I.htm**, tal que:
- El programa nos pregunta cuántas multiplicaciones queremos hacer.
- El programa nos las pregunta aleatoriamente.
- Al final el programa nos da la nota cualitativa (función del **Prog049f.htm**).

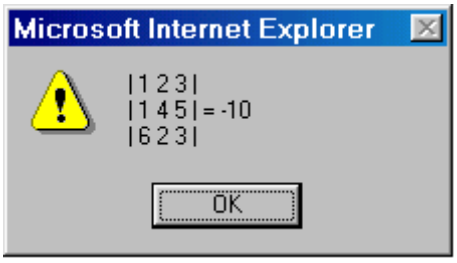
Repasa el segundo capítulo.

10) Haz un programa de nombre **Eval3J.htm** que lea un sistema de tres ecuaciones con tres incógnitas y nos escriba la matriz ampliada. Suponiendo que todos los coeficientes son positivos y constan de un único dígito. Deberás utilizar la función **texto.charAt(num)**

11) Haz un programa de nombre **Eval3K.htm**, que calcule un determinante de tercer orden.
Recuerda:

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = aei + dch + bfg - gec - hfa - dbi$$

Compruébalo para el caso:



12) Haz un programa de nombre **Eval3L.htm**, que sirva para discutir un sistema de tres ecuaciones con tres incógnitas, sólo en los casos compatible determinado e incompatible y lo resuelve en el caso compatible determinado.
De la siguiente forma:
1º) Crea una función que sirva para calcular un determinante de tercer orden (ejercicio anterior).

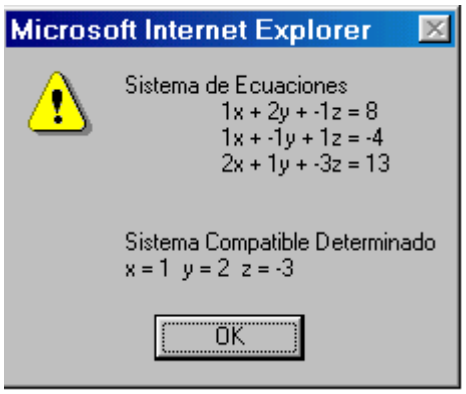
2º) Dado el sistema:
$$\left.\begin{array}{l} ax+by+cz=j \\ dx+ey+fz=k \\ gx+hy+iz=l \end{array}\right\}$$

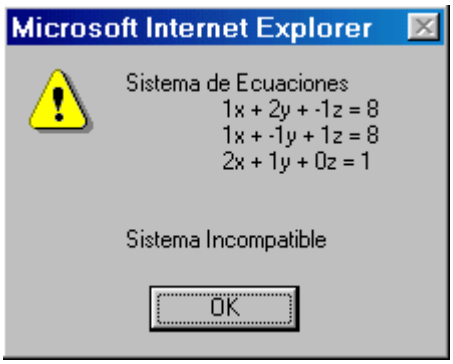
Calcula los determinantes: $det=\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix}$ $detx=\begin{vmatrix} j & b & c \\ k & e & f \\ l & h & i \end{vmatrix}$ $dety=\begin{vmatrix} a & j & c \\ d & k & f \\ g & l & i \end{vmatrix}$ $detz=\begin{vmatrix} a & b & j \\ d & e & k \\ g & h & l \end{vmatrix}$

3º) Si **det** no es 0 entonces el sistema es compatible determinado
Si **det** es 0 y (**detx** no es cero o **dety** no es 0 o **detz** no es 0) entonces el sistema es incompatible.

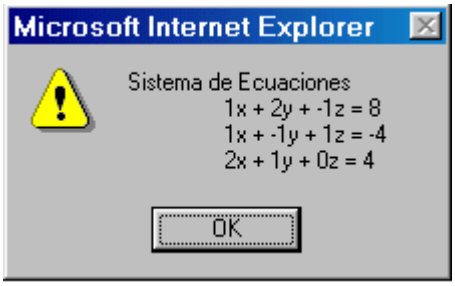
4º) Si el sistema es compatible determinado, la solución por Cramer es x=detx/det, y=dety/det, z=detz/det

Pruébalo para los casos:





El último caso corresponde a un sistema compatible indeterminado.



13) Haz un programa de nombre **Eval3M.htm**, que simule una tirada aleatoria de cinco dados de parchís.

14) Haz un programa de nombre **Eval3N.htm**, que resuelva un sistema de dos ecuaciones con dos incógnitas por el método de Cramer.
Crea una función que calcule un determinante de 2º orden

Compruébalo para el caso: $\left.\begin{matrix} 3x-y=1 \\ x+y=3 \end{matrix}\right\}$ solución: x=1, y=2

15) Haz un programa de nombre **Eval3O.htm**, que escriba un número en base 2 o 3 en base 10.
Repasa el primer capítulo

16) Haz un programa de nombre **Eval3P.htm**, que calcule los 50 primeros términos de la sucesión de termino general: (3n + 1) / (2n – 1)

17) Haz un programa de nombre **Eval3Q.htm**, que calcule la raíz cuarta de un número, utilizando una función.

18) Haz un programa de nombre **Eval3R.htm**, que cuente el número de vocales que contiene un texto cualquiera.

19) Haz un programa de nombre **Eval3S.htm**, que haga lo mismo que el anterior, pero al final el programa nos da el número de "a", número de "e", número de "i", y lo mismo con las "o" y "u".

IV.- Programación Visual

En este capítulo se trata de continuar programando en **JavaScript**, pero utilizando las posibilidades gráficas y visuales del navegador: botones, cuadros de texto, etc.

Para ello necesitamos conocer un poco más de los **tags propios del HTML**,

Texto

<!-- //-->
Permite escribir comentarios en una página web (no en un programa JavaScript).

**
**
Es equivalente a un [Return]. No existe el correspondiente tag de cierre.

<H1> </H1>
<H2> </H2>
<H3> </H3>
Nos sirven para escribir "títulos".

<P> </P>
Párrafo normal (justificación izquierda).

<P ALIGN="CENTER"> </P>
Párrafo centrado.

Colores

<BODY BGCOLOR="CódigoColor"> </BODY>
La página quedará coloreada con el color de fondo correspondiente a **CódigoColor**
La sintaxis de **CódigoColor** es #XXYYZZ
XX es un número en hexadecimal de 00 a FF correspondiente al rojo
YY es un número en hexadecimal de 00 a FF correspondiente al verde
ZZ es un número en hexadecimal de 00 a FF correspondiente al azul

Ejemplos:

#FF0000	rojo	#0000FF	azul
#9900DD	violeta	#00FF00	verde
#FF00FF	magenta		

<FONT COLOR="CódigoColor"> </FONT>
La letra aparecerá en el color correspondiente a **CódigoColor**.

Otros

<HR>
Dibuja una línea horizontal.

<B> </B>
Letra en negrita.

- Escribe el siguiente programa:

```
<HTML>
<HEAD><TITLE>Esto está en el navegador</TITLE></HEAD>
<!-- PROG061.HTM //-->
<BODY BGCOLOR="#000000">
<FONT COLOR="#FFFFFF">
<H1>Color de Fondo #000000=Negro</H1>
<BR>
<P ALIGN="CENTER">
El color del texto #FFFFFF=Blanco</P>
<BR>
<HR>
Esto está entre dos rayas horizontales
<HR>
<P ALIGN="CENTER">
<B>Esto es Negrita</B></P>
<BR>
Esto es letra normal
</FONT>
</BODY>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog061.htm** y ejecútalo.

- Escribe el siguiente programa:

```
<HTML>
<!-- PROG062.HTM //-->
<BODY BGCOLOR="#FF0000">
<FONT COLOR="#0000FF">
<B>
<BR>
<HR>
<P ALIGN="CENTER">
Pepito Valdemoro Ruiz
<BR>
Avda. Las Babosas 21-25, 2, 4
<BR>
08915 - BADALONA
</P>
<HR>
</B>
</FONT>
</BODY>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog062.htm** y ejecútalo.

Casi todos los programas que hemos hecho hasta ahora se han caracterizado en que su "salida" ha sido siempre utilizando la ventana "alert" de JavaScript. Vamos a ver en el siguiente programa que podemos "enviar la salida" a la misma pantalla del navegador.

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG063.HTM

var nom;
nom=prompt("Escribe tu nombre","");
var salida="";
for(i=2;i<=14;i=i+2)
        {
        salida=salida+"<BR>"+"i= "+i+" Hola "+nom;
        }
document.write("<H1><P align='CENTER'>Uso de DOCUMENT.WRITE</H1>");
document.write("<BR><BR>");
document.write(salida);

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog063.htm** y ejecútalo.

- En un programa JavaScript, la sentencia **document.write("cadena")** escribe en la pantalla del navegador, el valor de "cadena".
  Observa que en "cadena" no podemos incluir los códigos de escape "\n" y "\t", pero sí todos los **tags HTML** que queramos. De esta forma, en lugar de "\n", deberemos escribir **
, que es la orden equivalente en **HTML.

Utilizando los tags propios del HTML, podemos mejorar la salida de cualquier programa JavaScript, en efecto:

- Escribe el siguiente programa:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG063.HTM

var nom;
nom=prompt("Escribe tu nombre","");
var salida="";
for(i=2;i<=14;i=i+2)
        {
        salida=salida+"<BR>"+"i= "+i+" - Hola "+nom;
        }
document.write("<BODY BGCOLOR='#00FF00'><FONT COLOR='#FF0000'>");
document.write("<H1><P align='CENTER'>Uso de DOCUMENT.WRITE</H1>");
document.write("<BR><BR>");
document.write("<B>"+salida+"</B>");

</SCRIPT>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog064.htm** y ejecútalo.

Uso de los tags <FORM> </FORM>

Los llamados tags "formulario": <FORM> </FORM> nos permiten incluir en una página web, otros elementos visuales como "cuadros de texto" y "botones", de gran importancia en la programación en JavaScript.

Cuadros de Texto:

<FORM>
<INPUT TYPE="text" SIZE="10" NAME="pepe">
</FORM>
Crea un cuadro de texto de nombre "pepe" y tamaño 10 caracteres.

<FORM>
<INPUT TYPE="text" onBlur="programa JavaScript">
</FORM>
Crea otro cuadro de texto tal que, al salir de él se ejecuta el programa JavaScript.

- Escribe el siguiente programa:

```
<HTML>
<!-- PROG065.HTM //-->
<BODY>
<FORM>
ESCRIBE LO QUE QUIERAS:
<INPUT TYPE="text" SIZE="50" NAME="uno">
<BR><BR>
PULSA TAB para pasar al siguiente cuadro de texto:
<BR><BR>
VUELVE A ESCRIBIR LO QUE TE VENGA EN GANA:
<INPUT TYPE="text" onBlur="alert('vale');">
<BR>
<INPUT TYPE="text" VALUE="PUES ESO">
</FORM>
</BODY>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog065.htm** y ejecútalo varias veces, observando detenidamente lo que sucede.

- Estudio del **Prog065.htm**
  En principio el fichero anterior es una página web, que no tiene nada que ver con JavaScript (no aparecen los tags <SCRIPT Lan…> </SCRIPT>)
  Pero en el segundo cuadro de texto:
  <INPUT TYPE="text" onBlur="alert('vale');">
  tenemos un pequeño programa JavaScript (una única sentencia), que se ejecutará al abandonar el foco (onBlur) el cuadro de texto.
  - Si observamos los tres cuadros de texto, vemos que los parámetros SIZE, NAME, VALUE, onBlur son opcionales. De hecho un cuadro de texto es el tag: <INPUT TYPE="text">.
  - El primer cuadro de texto tiene el tamaño de 50 caracteres (SIZE="50"). Si no especificamos el tamaño (SIZE) en un cuadro de texto, por defecto es de 20 caracteres..
  - El parámetro NAME de un cuadro de texto, sólo tiene sentido en el caso de un programa JavaScript, para poder referirnos a él (como veremos en los siguientes ejercicios).
  - El parámetro VALUE nos permite referirnos al "valor" o contenido de un determinado cuadro de texto.

Creación de un programa JavaScript "visual"

Vamos a hacer nuestro primer programa "visual"…
El proceso a seguir es:

1º) Incluir en la cabecera (HEAD) de la página HTML, el programa JavaScript en forma de función.
Por ejemplo: Programa que calcula el área de un triángulo

```
<HEAD>
   <SCRIPT LANGUAGE="JavaScript">
   function a(x,y)
     {
     return (parseFloat(x)*parseFloat(y)/2);
     }
   </SCRIPT>
</HEAD>
```

2º) "Programar" en la página web, los elementos visuales que necesitamos para nuestro programa:

- Un cuadro de texto, para "recoger" la base del triángulo:
  Base del triángulo: <INPUT TYPE="text" SIZE="5" NAME="bas">
- Un cuadro de texto, para "recoger" la altura del triángulo:
  Altura del triángulo: <INPUT TYPE="text" SIZE="5" NAME="alt" >
- Un cuadro de texto, donde aparecerá el resultado del programa, es decir el área del triángulo:
  Área del triángulo: <INPUT TYPE="text" SIZE="10" NAME="result">

3º) "Programar" el enlace entre JavaScript y HTML, es decir, la ejecución del programa.
De momento sólo sabemos hacerlo utilizando el evento **onBlur** de un cuadro de texto. Es decir hemos de modificar el segundo cuadro de texto de forma que nos quede:

```
Altura del triángulo: <INPUT TYPE="text" SIZE="5" NAME="alt" onBlur=
"document.forms[0].result.value =a(document.forms[0].bas.value,this.value);">
```

Observa:

- Para referirnos al contenido (valor) del cuadro de texto de nombre "bas", utilizamos la sintaxis:
  document.forms[0].bas.value
- Para referirnos al contenido del cuadro activo, utilizamos la sintaxis: **this.value**

Vamos a ver si funciona:

- Escribe el siguiente programa:

```
<HTML>
<!-- PROG066.HTM //-->
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    function a(x,y)
         {
         return (parseFloat(x)*parseFloat(y)/2);
         }
    </SCRIPT>
</HEAD>

<BODY>
<FORM>
   <BR>
   Base: <INPUT TYPE="text" SIZE="5" NAME="bas">
   <BR>
   Altura: <INPUT TYPE="text" SIZE="5" NAME="alt"
onBlur="document.forms[0].result.value=a(document.forms[0].bas.value,this.value);">
   <BR>
```

```
    <BR>
    Área: <INPUT TYPE="text" SIZE="10" NAME="result">
</FORM>
</BODY>

</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog066.htm** y ejecútalo varias veces.

4º) Para acabar, es conveniente mejorar el diseño visual del programa: títulos, colores, etc.

- A partir del programa anterior consigue el siguiente fichero:

```
<HTML>
<!-- PROG067.HTM //-->
<HEAD>
    <SCRIPT LANGUAGE="JavaScript">
    function a(x,y)
        {
        return (parseFloat(x)*parseFloat(y)/2);
        }
    </SCRIPT>
</HEAD>

<BODY BGCOLOR="#0000FF">
<FONT COLOR="#FF0000">
    <H1><P ALIGN="CENTER">Calculo del Área de un Triángulo</H1>
    <HR>
</FONT>
<FONT COLOR="#FFFFFF"><P>
Escribe el los siguientes cuadros la base y la altura del triángulo.
Para pasar de un cuadro al siguiente pulsa la tecla de tabulación
 [Tab]. Al situarte en el último cuadro, automáticamente aparecerá
 el valor del área.
Para escribir un dato en decimales, debes utilizar el "punto decimal",
 no la coma.
<BR>
<HR>
<FONT COLOR="YELLOW">
<FORM>
    <P ALIGN="CENTER">
    Base del triángulo: <INPUT TYPE="text" SIZE="5" NAME="bas">
    <BR>
    Altura del triángulo: <INPUT TYPE="text" SIZE="5" NAME="alt"
onBlur="document.forms[0].result.value=a(document.forms[0].bas.value,this.value);">
    <BR>
    <BR><HR><P ALIGN="CENTER"><B>
    Área del triángulo: <INPUT TYPE="text" SIZE="10" NAME="result">
<HR>
</FORM>
</FONT></B>
</BODY>

</HTML>
```

- Grábalo con el nombre **Prog067.htm** y ejecútalo varias veces.

- Observa que en lugar de utilizar un código de color, también podemos utilizar el nombre del color en inglés.

Botones

Otro elemento interesante en la programación visual es el uso de botones:

```
<FORM>
<INPUT TYPE=”button” VALUE=”Pepito” onClick=”programaJS;”>
</FORM>
```

El tag <INPUT TYPE...> anterior crea un botón de comando, donde aparece escrito el texto “Pepito” y al hacer clic en dicho botón (onClick), se ejecuta el “programaJS”.

Vamos a ver como funciona...

- Escribe el siguiente programa:

```
<HTML>
<!-- PROG068.HTM //-->
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
  function Saludo()
        {
        alert("Hola Mundo!");
        }
</SCRIPT>
</HEAD>
<BODY>
<P ALIGN="CENTER">
<FORM>
   <INPUT TYPE="button" VALUE="Hazme Clic" onClick="Saludo();">
</FORM>
</P>
</BODY>
</HTML>
```

- Grábalo con el nombre **Prog068.htm** y ejecútalo.

El “saludo” anterior podría ser más sofisticado, en efecto:

- Escribe el siguiente programa:

```
<HTML>
<!-- PROG069.HTM //-->
<HEAD>
<SCRIPT LANGUAGE="JavaScript">
  function Saludo(nom)
        {
        alert("Hola "+nom);
        }
</SCRIPT>
</HEAD>
<BODY>
<P ALIGN="CENTER">
<FORM>
   Escribe tu nombre:
   <INPUT TYPE="text" NAME="x">
   <BR><BR>
```

```
   <INPUT TYPE="button" VALUE="Hazme Clic" onClick="Saludo
(document.forms[0].x.value);">
</FORM>
</P>
</BODY>
</HTML>
```

EJEMPLOS "visuales"

Para acabar este tema se trata de **repetir** una serie de programas ya hechos en capítulos anteriores, pero utilizando las posibilidades "visuales" que acabamos de ver.

Ejemplo1: Tabla de valores de la función $y=x^2-5x+10$
El programa correspondiente era el **Prog023.htm** que ya tienes en *TuCarpeta*:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG023.HTM

var x1,x2,paso;
var salida="";
var y;
x1=prompt("Escribe el menor valor de x","");
x1=parseFloat(x1);
x2=prompt("Escribe el mayor valor de x","");
x2=parseFloat(x2);
paso=prompt("Escribe el incremento de x:","");
paso=parseFloat(paso);
for(i=x1;i<=x2;i=i+paso)
  {
  y=i*i-5*i+10;
  salida=salida+i+" "+y+"\n";
  }
alert(salida);

</SCRIPT>
</HTML>
```

Es conveniente ejecutar el **Prog023.htm**, para tener claro su funcionamiento.

Bien, vamos a seguir el proceso para transformar el programa anterior en "visual":

1º) Programa JavaScript en forma de función:

```
<SCRIPT LANGUAGE="JavaScript">
        function valores(x1,x2,paso)
           {
                var y;
                var salida="";
                x1=parseFloat(x1);
                x2=parseFloat(x2);
                paso=parseFloat(paso);
                for(i=x1;i<=x2;i=i+paso)
                        {
                        y=i*i-5*i+10;
                        salida=salida+"x= "+i+"    y="+y+"<BR>";
                        }
```

```
            document.write(salida);
        }
</SCRIPT>
```

2º) Elementos visuales:

Tres cuadros de texto para introducir x1, x2 y paso
Un botón para ejecutar el programa

Es decir:

```
<FORM>
Primer o mínimo valor de X:<INPUT TYPE="text" NAME="min">
<BR>
Último o máximo valor de X:<INPUT TYPE="text" NAME="max">
<BR>
Incremento o Variación entre los valores de X: <INPUT TYPE="text" NAME="incr">
<BR>
<HR>
<INPUT TYPE="button" VALUE="Tabla de Valores"
onClick="valores(document.forms[0].min.value,document.forms[0].max.value,document.forms[0].incr.v
alue);">
<HR>
<BR>
</FORM>
```

Vamos a ver antes de continuar si funciona **lo fundamental**:

- Escribe el siguiente programa:

```
<HTML>
<!-- PROG070.HTM //-->
<HEAD>
  <SCRIPT LANGUAGE="JavaScript">
      function valores(x1,x2,paso)
        {
            var y;
            var salida="";
            x1=parseFloat(x1);
            x2=parseFloat(x2);
            paso=parseFloat(paso);
            for(i=x1;i<=x2;i=i+paso)
                {
                y=i*i-5*i+10;
                salida=salida+"x= "+i+"    y="+y+"<BR>";
                }
            document.write(salida);
        }
  </SCRIPT>
</HEAD>
<BODY>
<FORM>
Primer o mínimo valor de X:<INPUT TYPE="text" NAME="min">
<BR>
Último o máximo valor de X:<INPUT TYPE="text" NAME="max">
<BR>
Incremento o Variación entre los valores de X: <INPUT TYPE="text" NAME="incr">
<BR>
```

```
<HR>
<INPUT TYPE="button" VALUE="Tabla de Valores"
onClick="valores(document.forms[0].min.value,document.forms[0].max.value,document.forms[0].i
ncr.value);">
<HR>
<BR>
</FORM>
</BODY>
</HTML>
```

- Grábalo en *TuCarpeta* con el nombre **Prog070.htm** y ejecútalo.

3º) Por último se trata de hacerlo "bonito":

- Escribe:

```
<HTML>
<!-- PROG071.HTM //-->
<HEAD>
   <SCRIPT LANGUAGE="JavaScript">
         function valores(x1,x2,paso)
            {
                  var y;
                  var salida="";
                  x1=parseFloat(x1);
                  x2=parseFloat(x2);
                  paso=parseFloat(paso);
                  for(i=x1;i<=x2;i=i+paso)
                        {
                        y=i*i-5*i+10;
                        salida=salida+"x= "+i+"    y="+y+"<BR>";
                        }
                  document.write("<P ALIGN='CENTER'><FONT COLOR='red'><B>"+salida);
            }
   </SCRIPT>
</HEAD>
<BODY BGCOLOR="GREEN"><FONT COLOR="YELLOW"><H1><P ALIGN="CENTER">
Tabla de Valores de la función: Y=X*X-5*X+10</H1>
<FORM>
Primer o mínimo valor de X:<INPUT TYPE="text" NAME="min">
<BR>
Último o máximo valor de X:<INPUT TYPE="text" NAME="max">
<BR>
Incremento o Variación entre los valores de X: <INPUT TYPE="text" NAME="incr">
<BR>
<HR><P ALIGN="CENTER">
<INPUT TYPE="button" VALUE="Tabla de Valores"
onClick="valores(document.forms[0].min.value,document.forms[0].max.value,document.forms[0].i
ncr.value);">
<HR>
<BR>
</FORM>
</BODY>
</HTML>
```

- Grábalo con el nombre **Prog071.htm** y ejecútalo.

- Es interesante probarlo con valores relativamente grandes, por ejemplo: x1=1, x2=200, paso=1

Ejemplo2: Cálculo del factorial de un número

El programa correspondiente era el **Prog021.htm**:

```
<HTML>
<SCRIPT LANGUAGE="JavaScript">

// PROG021.HTM
var salida="";
var fact=1;
var num;
num=prompt("Cálculo del factorial del numero ","");
num=parseInt(num,10);
for(i=1;i<=num;i++) fact=fact*i;
alert("El factorial de "+num+" es "+fact);


</SCRIPT>
</HTML>
```

Ejecuta varias veces el **Prog021.htm**, para tener claro su funcionamiento.

Vamos allá:

1º) Programa en forma de función:

```
<SCRIPT LANGUAGE="JavaScript">
   function factorial(num)
          {
          var fact=1;
          num=parseInt(num);
          for(i=1;i<=num;i++) fact=fact*i;
          return fact;
          }
</SCRIPT>
```

2º) Elementos visuales:
Un cuadro de texto para introducir un número.
Un cuadro de texto para recoger el resultado (el factorial)
Un botón para ejecutar el programa

- Escribe:

```
<HTML>
<!-- PROG072.HTM //-->
<HEAD>

<SCRIPT LANGUAGE="JavaScript">
   function factorial(num)
          {
          var fact=1;
          num=parseInt(num);
```

```
        for(i=1;i<=num;i++) fact=fact*i;
        return fact;
        }
</SCRIPT>

</HEAD>
<BODY>
<FORM>
Escribe el número:<INPUT TYPE="text" NAME="x">
<BR><BR>
<INPUT TYPE="button" VALUE="Calcular el Factorial"
onClick="document.forms[0].f.value=factorial(document.forms[0].x.value);">
<BR><BR>
<INPUT TYPE="text" NAME="f">

</FORM>
</BODY>
</BODY>
</HTML>
```

- Grábalo con el nombre **Prog072.htm** y ejecútalo.

- Vamos a hacer una mejora: **"la posibilidad de volver a empezar"**

Escribe:

```
<HTML>
<!-- PROG073.HTM //-->
<HEAD>

<SCRIPT LANGUAGE="JavaScript">
   function factorial(num)
        {
        var fact=1;
        num=parseInt(num);
        for(i=1;i<=num;i++) fact=fact*i;
        return fact;
        }
</SCRIPT>

</HEAD>
<BODY>
<FORM>
Escribe el número:<INPUT TYPE="text" NAME="x">
<BR><BR>
<INPUT TYPE="button" VALUE="Calcular el Factorial"
onClick="document.forms[0].f.value=factorial(document.forms[0].x.value);">
<BR><BR>
<INPUT TYPE="text" NAME="f">
<BR><BR>
<INPUT TYPE="button" VALUE="OTRO"
onClick="document.forms[0].x.value='';document.forms[0].f.value='';">
</FORM>
</BODY>
</BODY>
</HTML>
```

- Grábalo con el nombre **Prog073.htm** y ejecútalo.

3º) Por último vamos a hacerlo bonito:

- Escribe:

```
<HTML>
<!-- PROG074.HTM //-->
<HEAD>

<SCRIPT LANGUAGE="JavaScript">
   function factorial(num)
          {
          var fact=1;
          num=parseInt(num);
          for(i=1;i<=num;i++) fact=fact*i;
          return fact;
          }
</SCRIPT>

</HEAD>
<BODY BGCOLOR="#000000"><FONT COLOR="WHITE">
<H1><P ALIGN="CENTER">CÁLCULO DEL FACTORIAL DE UN NÚMERO</H1>
<BR>
<H3>El factorial de un número es el producto del número por los sucesivos
enteros anteriores, hasta llegar a la unidad.
Es decir, el factorial de 5 será 5*4*3*2*1 por lo tanto 120.</H3>
<HR>
<FORM>
<H2><P ALIGN="CENTER"><FONT COLOR="RED">
Escribe el número:<INPUT TYPE="text" NAME="x">
<BR><BR>
<INPUT TYPE="button" VALUE="Calcular el Factorial"
onClick="document.forms[0].f.value=factorial(document.forms[0].x.value);">
<INPUT TYPE="button" VALUE="Otra vez"
onClick="document.forms[0].x.value='';document.forms[0].f.value='';">
<BR><BR>
<INPUT TYPE="text" NAME="f">
<BR><BR>

</FORM>
</BODY>
</BODY>
</HTML>
```

- Grábalo con el nombre **Prog074.htm** y ejecútalo.

Ejemplo3: Simplificar una fracción

- Escribe:

```
<HTML>
<!-- PROG075.HTM //-->
<HEAD>
<SCRIPT LANGUAGE="JavaScript">

function MCD(a,b)
{
var resto,aux;
if(a<b)
        {
        aux=a;
        a=b;
        b=aux;
        }
if ((a%b)==0) resto=b;
while((a%b) !=0)
        {
        resto=a%b;
        a=b;
        b=resto;
        }
return resto;
}

</SCRIPT>

</HEAD>

<BODY BGCOLOR="YELLOW"><FONT COLOR="RED">
<H1><P ALIGN="CENTER">SIMPLIFICACIÓN DE FRACCIONES</H1>

<FORM><B>
   <p align="center">Numerador : <input type="text" size="9"
   name="num1"> <br>
   Denominador: <input type="text" size="9" name="den1"

onblur="document.forms[0].num2.value=document.forms[0].num1.value/MCD(document.forms[0]
.num1.value,this.value);document.forms[0].den2.value=this.value/MCD(document.forms[0].num1.v
alue,this.value);">
   <br>
   </p>
   <hr>
   <p align="center"><br>
   <input type="text" size="7" name="num2"> <br>
   <input type="text" size="7" name="den2"> <br><br>
   <INPUT TYPE="button" VALUE="Otra Fracción"
onClick="document.forms[0].num1.value='';document.forms[0].den1.value='';document.forms[0].
num2.value='';document.forms[0].den2.value='';">
</p>
</form>
</body>
</html>
```

- Grábalo con el nombre **Prog075.htm** y ejecútalo.

Observa entre otras cosas que los tags HTML, pueden escribirse en minúscula.

Autoevaluación IV

1º) En el programa "Eval1A.htm" habíamos hecho un programa que restaba dos números.
Has de hacer un programa "visual" de nombre **Eval4A.htm**, que haga lo mismo, de la siguiente forma:
- Dos cuadros de texto para introducir los dos números
- Un botón para ejecutar el programa
- Un cuadro de texto para el resultado
- El programa debe quedar bonito, es decir: títulos, colores, etc.

2º) En el programa "Eval1C.htm" teníamos un programa que nos preguntaba el nombre y la edad y que nos daba por resultado los días vividos hasta el momento.
Repite el programa pero "visualmente" (grábalo con el nombre **Eval4B.htm**) de la siguiente forma:
- Dos cuadros de texto para introducir el nombre y la edad en años.
- Al pulsar [Tab] en el 2º cuadro de texto aparece en un tercer cuadro de texto el nombre introducido en el primer cuadro y los días vividos hasta el momento.
- El programa debe incluir títulos, colores, etc.

3º) En el "Prog028.htm" habíamos hecho un programa que construía una tabla de senos.
Haz un programa de nombre **Eval4C.htm**, que construya la tabla de senos anterior, pero su salida en lugar de ser por un "alert", que sea por la pantalla del navegador. Mejora dicha salida, por ejemplo: letra en negrita, color rojo (fondo negro) y centrado.

4º) Repite el "Eval1F.htm" (área y longitud de una circunferencia), pero "visualmente" de la siguiente forma:
- Un cuadro de texto para "recoger" el radio.
- Dos botones para calcular la longitud y el área
- Dos cuadros de texto para escribir la longitud y el área
- Un b´tón para "volver a empezar".
- El programa debe incluir títulos, colores, etc.

Grábalo con el nombre **Eval4D.htm**

5º) Repite el "Eval1H.htm" (cálculo de un determinante de 2º orden), pero "visualmente".
Grábalo con el nombre **Eval4E.htm**

6º) Repite el "Eval1I.htm" pero "visualmente". Grábalo con el nombre **Eval4F.htm**

7º) Haz un programa "visual" que sirva para pasar un número de una base de numeración cualquiera a otra base de numeración cualquiera. Básate en los ejercicios Eval1K, Eval1L y Eval1M.
Graba el programa con el nombre **Eval4g.htm**

8º) Repite el programa "Eval2H.htm" (cálculo de un cateto a partir de la hipotenusa y el otro cateto), pero "visualmente".
Graba el programa con el nombre **Eval4h.htm**

9º) Repite el programa "Eval2f.htm" (cálculo de los múltiplos de 23 inferiores a 1000 y su suma), pero "visualmente".
Graba el programa con el nombre **Eval4i.htm**

10º) Repite el programa “Eval2p.htm” (producto escalar y vectorial), pero “visualmente”.
Graba el programa con el nombre **Eval4j.htm**

11º) Repite el programa “Eval3a.htm” (cálculo del m.c.m.), pero “visualmente”.
Graba el programa con el nombre **Eval4k.htm**

12º) Repite el programa “Eval3f.htm” (estadística), pero “visualmente”.
Graba el programa con el nombre **Eval4l.htm**

13º) Repite el programa “Eval3i.htm” (productos aleatorios), pero “visualmente”.
Graba el programa con el nombre **Eval4m.htm**

14º) Repite el programa “Eval3l.htm” (discusión de un sistema lineal), pero “visualmente”.
Graba el programa con el nombre **Eval4n.htm**

Hasta aquí, la versión no registrada del manual.

Si deseas la parte que falta, es decir:

Debes adquirir la versión registrada, es decir entera.
Es muy fácil, has de hacer lo siguiente:

1) **Rellena el siguiente formulario con tus datos:**

Nombre y Apellidos:

Dirección:

Código Postal: **Población:**

Versión completa del "JavaScript (Manual FV)"

2) **Envíame el formulario anterior por correo ordinario junto con un "billete" (lo que consideres justo por un disquete, gastos de manipulación, envío y "estímulo por mi parte para que continue colgando en Internet, mis manuales").**

   A mi dirección que es:

   F.V.
   c) Valencia 21-25, 2º , 4ª
   08915 – Badalona (Barcelona)
   España

3) **A vuelta de correo recibirás en tu dirección, un disquete con la versión completa del manual "JavaScript (Manual FV)".**