

JavaScript

básico

Introducción

¿Qué es JavaScript?

- Es EL LENGUAJE de scripts en la Web

- .Es un lenguaje de script, interpretado

- .(no se compila, lo interpreta directamente el navegador),

- .sin coste por licencia.

- No tiene nada que ver con Java.

- Se puede ejecutar también fuera del navegador:

En un entorno consola [Mozilla Rhino](#)

En un entorno servidor [node.js](#)

¿Para qué sirve JavaScript?

- Herramienta de programación (sentencias condicionales, bucles...)
- Lee, añade y modificación elementos y atributos en una página HTML:
`document.getElementById('miEnlace').href="http://www.soplaelcierzo.com";`
- Puede reaccionar a eventos:
 - Página que termina de cargar
 - Click en un elemento HTML
 - ...
- Validación de datos en formularios (ahorra procesamiento al servidor y agiliza la respuesta a errores para el usuario).
- Detección del navegador del usuario
- *Creación de cookies*
-

Etiquetas de JavaScript en el HTML

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

```
//aquí iría nuestro código de JavaScript
```

```
</script>
```

// Comentarios en JavaScript


Las etiquetas **<script>** de apertura y cierre son necesarias.

El **type** por defecto es **"text/javascript"** así que no es necesario indicarlo.

language="JavaScript" está obsoleto, ¡el **type** es suficiente!

JavaScript dentro del documento html

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Mi primer javascript </title>
  </head>
  <body>
    <script type="text/javascript">
      ....
    </script>
  </body>
</html>
```



```
<script type="text/javascript">
  var x = 3;
  alert('Hola Mundo, hola variable ' + x);
</script>
```

JavaScript en fichero externo

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo JavaScript en fichero externo</title>
    <script src="fichero-externo.js"></script>
  </head>
  <body>
  </body>
</html>
```

En un fichero separado:

.Para ahorrar código si se van a ejecutar los mismos scripts en varias páginas html y mejorar su mantenimiento.

- Para aprovechar mejor la caché de los navegadores → carga más rápida de la página web
- El script de JavaScript externo debe tener extensión .js

JavaScript como elemento inline

```
<html>
  <head>
  </head>
  <body>
    <p onclick="funcion();">
      Ejemplo de Javascript
    </p>
  </body>
</html>
```

En los propios elementos html

- Existen muchos atributos en html relacionados con eventos
- Ensucia el código html y complica el mantenimiento del javascript
- Solo lo utilizaremos para algunos casos especiales

Etiqueta `<noscript>`

El JavaScript puede estar deshabilitado por el usuario en alguna ocasión en el navegador.

Si la página web requiere JavaScript para su correcto funcionamiento, se puede mostrar un mensaje de aviso al usuario indicándole que debería activar JavaScript para disfrutar completamente de la página

```
<head>
    ....
</head>
<body>
    <noscript>
        <p>Bienvenido a Mi Sitio</p>
        <p>
            Esta página requiere para su funcionamiento el uso de JavaScript.
            Si lo has deshabilitado intencionadamente, por favor vuelve a activarlo.
        </p>
    </noscript>
</body>
```

Ejemplo I – JavaScript en el head

Observa que el navegador renderiza la página línea a línea.

En este caso, debido a la **función alert** hace una parada esperando la interacción del usuario.

```
<!DOCTYPE html>
  <head>
    <meta charset="utf-8">
    <title>Ejemplo de código JavaScript en el propio documento</title>
    <script type="text/javascript">
      alert("Hola Mundo");
    </script>
  </head>
  <body>
    <p>Mi primer programa con Javascript</p>
  </body>
</html>
```

Ejemplo II – JavaScript en el body

El navegador mostrará la alerta inmediatamente después de renderizar el párrafo dentro del body.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ejemplo de código JavaScript en el propio documento</title>
  </head>
  <body>
    <p>Mi primer programa con Javascript</p>
    <script type="text/javascript">
      alert("Hola Mundo");
    </script>
  </body>
</html>
```

Colocación idónea del JavaScript

El lugar clásico ha sido entre las etiquetas `<head>` `</head>`, pero tiene dos problemas:

- El script no puede tener todavía acceso al DOM del documento.
- Retrasa la carga de la página

Actualmente se hace **al final del body**, pero tiene otro problema:

.La página html puede estar disponible para el usuario antes de que el JavaScript se haya cargado: validación formularios, etc...

También se puede optar por soluciones mixtas.

Ejercicios

Ejercicio 1

Crea el ejemplo II anterior pero con el JavaScript como un elemento inline, de modo que al hacer clic en el párrafo salga la ventana de alerta.

Ejercicio 2

Realiza el ejemplo I anterior pero con el JavaScript en un archivo externo llamado codigo.js.

Después del primer mensaje, se debe mostrar otro mensaje que diga "Hola Mundo por segunda vez"

Añadir algunos comentarios que expliquen el funcionamiento del código

Añadir en la página un mensaje de aviso para los navegadores que no tengan activado el soporte de JavaScript

Ejercicio 1 solucionado

¡Ojo con las comillas!

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ejercicio 1</title>
  </head>
  <body>
    <p onclick="alert('Hola Mundo')">
      Mi primer programa con Javascript
    </p>
  </body>
</html>
```

Ejercicio 2 solucionado

Página html:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ejercicio 2 - Archivo externo y varios mensajes</title>
    <script type="text/javascript" src="codigo.js"></script>
  </head>
  <body>
    <noscript>
      Esta página <strong>requiere</strong> el uso de JavaScript
    </noscript>
    <p>Esta página muestra 2 mensajes</p>
  </body>
</html>
```

codigo.js:

```
// Al cargarse el archivo JavaScript, se muestra un mensaje
alert("Hola Mundo!");
// Despues del primer mensaje, se muestra otro mensaje seguido
alert("Hola Mundo por segunda vez!");
```

Pedir ayuda en javaScript

*Pierde tiempo en formular bien la pregunta, se específico.
Omite texto como “Hello”, “Thank you”, o “I'm getting mad this doesn't work”
(además los editores lo eliminarán de tu pregunta)*

Se responde muy rápido:

<http://stackoverflow.com/questions/1986896/what-is-the-difference-between-call-and-apply>

El usuario que pregunta da una respuesta como válida y los usuarios las valoran como útiles o no.

También me podéis preguntar a mi, pero no soy tan rápido ni tengo tanto conocimiento :-)

Suele ser útil adjuntar en las preguntas el código o un jsfiddle:

<http://stackoverflow.com/questions/18278511/highcharts-add-points-but-not-to-all-categories>

Hay otras opciones en español, pero no es lo mismo:

<http://soyprogramador.com/>

Programación básica

Elementos del lenguaje

¡JavaScript diferencia MAYÚSCULAS y minúsculas!

Sentencias:

```
document.write("¡Hola Mundo!");
```

Acaban en ";" pero es opcional: también pueden acabar con salto de línea.

Aviso de continuación de sentencia en otra línea:

```
document.write("Hello \nWorld!");
```

Comentarios:

```
// comentario de una línea
```

```
/* comentario
```

```
    Multilínea */
```

Se ignoran los espacios extras en blanco:

```
nombre="Pepito";
```

```
nombre = "Pepito";
```

Son equivalentes

Variables: declaración e inicialización

Declaración de variables

```
var x;
```

```
var _nombrecoche;
```

Las variables no se inicializan (su valor está vacío). ¡Habrás que inicializarlas!

Declaración e inicialización de variables

```
var x=5;
```

```
var _nombrecoche="volvo";
```

Las variables se pueden declarar e inicializar a la vez. Si es texto, ¡entre comillas!

Se pueden utilizar variables sin haber sido declaradas:

```
var numero_1;
```

```
var numero_2;
```

```
numero_1 = 3;
```

```
numero_2 = 1;
```

```
resultado = numero_1 + numero_2;
```

Variables: Numéricas y texto

Numéricas

```
var iva = 16;      // variable tipo entero
var total = 234.65; // variable tipo decimal
```

Cadenas de texto

```
/* El contenido de texto1 tiene comillas simples, por lo que
se encierra con comillas dobles */
```

```
var texto1 = "Una frase con 'comillas simples' dentro";
```

```
/* El contenido de texto2 tiene comillas dobles, por lo que
se encierra con comillas simples */
```

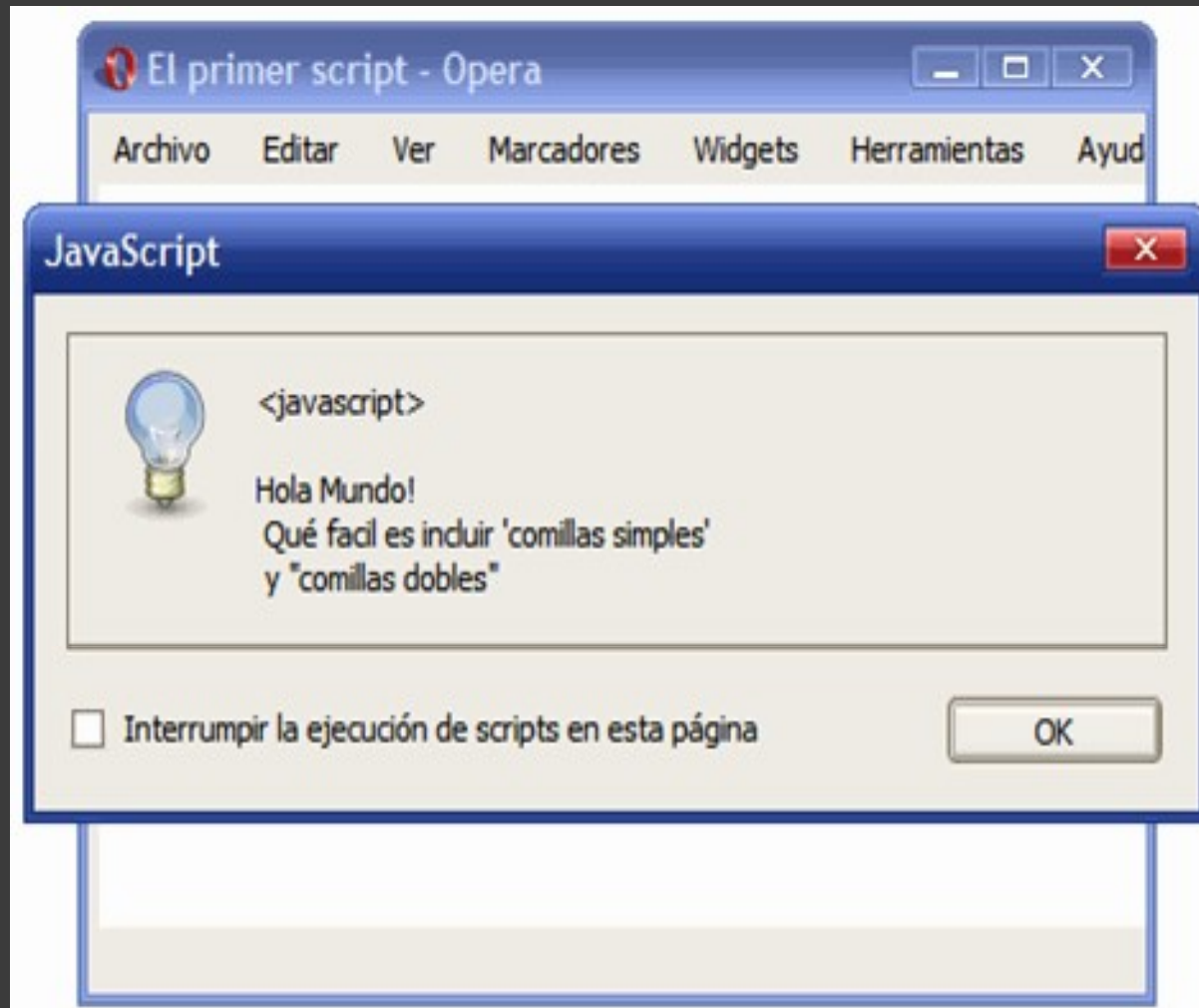
```
var texto2 = 'Una frase con "comillas dobles" dentro';
```

Caracteres especiales:

\'	comilla simple
\"	comillas
\&	ampersand
\\	backslash
\n	nueva línea
\r	retorno de carro
\t	tabulador
\b	backspace
\f	form feed

Ejercicio 3

Crea una página que guarde un texto en una variable y lo muestre luego por pantalla:



Ejercicio 3 solucionado

```
<!DOCTYPE html>
  <head>
    <meta charset="utf-8" />
    <title>Ejercicio 2 - Mostrar mensajes complejos</title>
    <script type="text/javascript">
      var msg = "Hola Mundo! \n Qué facil es incluir \'comillas simples\' \n
      alert(msg);
    </script>
  </head>
  <body>
    <p>Esta página muestra un mensaje complejo</p>
  </body>
</html>
```

Variables: Arrays

```
var dia1 = "Lunes";  
var dia2 = "Martes";  
...  
var dia7 = "Domingo";
```



Un array es una colección de variables,
que pueden ser todas del mismo tipo o cada una
de un tipo diferente.

```
var dias = ["Lunes", "Martes", "Miércoles", "Jueves", "Viernes", "Sábado", "Domingo"];
```

```
var nombre_array = [valor1, valor2, ..., valorN];
```

```
var diaSeleccionado = dias[0]; // diaSeleccionado = "Lunes"
```

```
var otroDia = dias[5]; // otroDia = "Sábado"
```

Ejercicio 4

Crear un array llamado meses y que almacene el nombre de los doce meses del año.

Mostrar por pantalla los doce nombres utilizando la función alert().

Ejercicio 4 solucionado

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8" />
```

```
    <title>Ejercicio 3 - Arrays simples</title>
```

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

```
      var meses = ["Enero", "Febrero", "Marzo", "Abril", "Mayo", "Junio",
```

```
      alert(meses[0]); alert(meses[1]);alert(meses[2]);alert(meses[3]);
```

```
      alert(meses[4]); alert(meses[5]);alert(meses[6]);alert(meses[7]);
```

```
      alert(meses[8]); alert(meses[9]);alert(meses[10]);alert(meses[11]);
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
    <p>Esta página muestra los 12 meses del año</p>
```

Variables: Booleanos

Puede tener 2 valores: false o true.

Ejemplos:

```
var clienteRegistrado = false;  
var ivaIncluido = true;
```

Funciones

```
function nombre_función (var1,var2,...,varX)
{
    //aquí iría el código de la función
}
```

Ejemplo:

```
function multiplicar(a,b)
{
    x=a*b;
    return x;
}
```

Llamada a la función (se guarda en la variable resultado, declarada previamente):

```
resultado=multiplicar(2,3);
```

Las variables son locales a las funciones donde son declaradas.

Las variables definidas fuera de las funciones, pueden ser accedidas desde todas.

Ámbito de las variables

Variables locales: Declaradas dentro de una función. Visibles solo desde la función.

Variables globales: Declaradas fuera de las funciones. ¡O sin declarar!

```
function creaMensaje() {  
  var mensaje = "Mensaje de prueba";  
  alert(mensaje);  
}  
CreaMensaje();  
//variable local, el alert tiene mensaje
```

```
function creaMensaje() {  
  var mensaje = "Mensaje de prueba";  
}  
creaMensaje();  
alert(mensaje);  
//variable local, el alert vacío
```

```
var mensaje = "Mensaje de prueba";  
  
function muestraMensaje() {  
  alert(mensaje);  
}  
//variable global, el alert tiene mensaje
```

```
function creaMensaje() {  
  mensaje = "Mensaje de prueba";  
}  
creaMensaje();  
alert(mensaje);  
//variable global, el alert tiene mensaje
```

Funciones como variables

Las funciones en JavaScript se pueden asociar a variables:

```
function elevarAlCuadrado (x) {return x*x;}
```

Se podría definir como:

```
var elevarAlCuadrado = function (x) {return x*x;}  
    x=a*b;  
    return x;  
}
```

Se llaman funciones lambda o funciones anónimas.

Los métodos en los objetos no serán otras cosas que propiedades de los objetos que están asociadas a funciones.

Operadores: Asignación

Operadores de asignación:

= += -= *= /= %=

```
var numero1 = 3;  
var numero2 = 4;
```

```
/* Error, la asignación siempre se realiza a una variable,  
   por lo que en la izquierda no se puede indicar un número */  
5 = numero1;
```

```
// Ahora, la variable numero1 vale 5  
numero1 = 5;
```

```
// Ahora, la variable numero1 vale 4  
numero1 = numero2;
```

Operadores: Incremento y decremento

```
var numero = 5;  
++numero;  
alert(numero); // numero = 6
```

```
var numero = 5;  
numero=numero +1;  
alert(numero); // numero = 6
```

```
var numero = 5;  
--numero;  
alert(numero); // numero = 4
```

```
var numero = 5;  
numero=numero - 1;  
alert(numero); // numero = 4
```

Los operadores incremento y decremento se pueden ejecutar como prefijo o sufijo:

```
var numero1 = 5;  
var numero2 = 2;  
numero3 = numero1++ + numero2;  
// numero3 = 7, numero1 = 6
```

```
var numero1 = 5;  
var numero2 = 2;  
numero3 = ++numero1 + numero2;  
// numero3 = 8, numero1 = 6
```

Operadores lógicos

Negación

```
var visible = true;  
alert(!visible); // Muestra "false" y no "true"
```

AND

```
var valor1 = true;  
var valor2 = false;  
resultado = valor1 && valor2; // resultado = false
```

```
valor1 = true;  
valor2 = true;  
resultado = valor1 && valor2; // resultado = true
```

OR

```
var valor1 = true;  
var valor2 = false;  
resultado = valor1 || valor2; // resultado = true
```

```
valor1 = false;  
valor2 = false;  
resultado = valor1 || valor2; // resultado = false
```


Operadores matemáticos

Operadores:

+ - * / %

Ejemplos:

```
var numero1 = 10;  
var numero2 = 5;  
resultado = numero1 / numero2; // resultado = 2  
resultado = numero1 % numero2; // resultado = 0
```

```
numero1 = 9;  
numero2 = 5;  
resultado = numero1 % numero2; // resultado = 4
```

```
var numero1 = 5;  
numero1 += 3; // numero1 = numero1 + 3 = 8  
numero1 -= 1; // numero1 = numero1 - 1 = 4  
numero1 *= 2; // numero1 = numero1 * 2 = 10  
numero1 /= 5; // numero1 = numero1 / 5 = 1  
numero1 %= 4; // numero1 = numero1 % 4 = 1
```

Operadores relacionales

Operadores aritméticos:

+ - * / % ++ --

Operadores de asignación:

= += -= *= /= %=

Concatenación de texto:

```
txt1="Vaya día";  
txt2="tan bonito";  
txt3=txt1+" "+txt2;
```

Si añades un número y un string, el resultado será un string. Ej: 5 + "5" = "55"

Operadores de comparación

== === != > <
 >= <=

=== igualdad "exacta" (tanto en valor como en tipo)

Operadores lógicos

&& || !

Sentencias condicionales: if-else

Sentencias condicionales

if

```
<script type="text/javascript">
    //Escribe "A desayunar" si son las 9
    var d=new Date();
    var time=d.getHours();
    if (time==9)
    {
        document.write("<b>¡A desayunar!</b>");
    }
</script>
```

If-else

-similar-

if-else if-else

```
<script type="text/javascript">
    var d = new Date()
    var time = d.getHours()
    if (time>6 && time<=14)
    {
        document.write("<b>Buenos días</b>");
    }
    else if (time>14 && time<22)
    {
        document.write("<b>Buenas tardes</b>");
    }
    else
    {
        document.write("<b>Buenas noches</b>");
    }
</script>
```

Sentencias condicionales: switch

Sentencias condicionales switch

```
<script type="text/javascript">  
    //Recibes diferente saludo en función del día de la semana en que estemos  
    //Domingo=0, Lunes=1 etc.  
    var d=new Date();  
    dia=d.getDay();  
    switch (dia)  
    {  
        case 5:  
            document.write("Por fin viernes");  
            break;  
        case 6:  
            document.write("Sábado fiestuqui");  
            break;  
        case 0:  
            document.write("A dormir que es domingo");  
            break;  
        default:  
            document.write("¡A ver si llega el fin de semana!");  
    }  
</script>
```

Ventanas popup

Alert box

Presentan un botón de ok. Simplemente muestran información.

```
<html>
  <head>
    <script type="text/javascript">
      function mostrar_alerta()
      {
        alert("¡Hola!" + '\n' + "Me has pulsado!");
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="mostrar_alerta()" value="Mostrar ventana de alerta" />
  </body>
</html>
```

Ventanas popup

Prompt box

Se utilizan para que el usuario introduzca un valor antes de entrar a la página.

```
<html>
  <head>
    <script type="text/javascript">
      function disp_prompt()
      {
        var nombre=prompt("Por favor escribe aquí tu nombre","Pepito p
        if (nombre!=null && nombre!="")
        {
          document.write("Hola " + nombre + "¿Qué tal estás?");
        }
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="disp_prompt()" value="Mostrar un prompt box" />
  </body>
</html>
```

Ventanas popup

Confirm box

Presentan un botón de ok y otro de cancelar. Devuelven verdadero (ok) o falso (cancel).

```
<html>
```

```
  <head>
```

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

```
      function disp_confirm()  
      {
```

```
        var r=confirm("Pulsa un botón");
```

```
        if (r==true)
```

```
        {
```

```
          document.write("¡Has pulsado el botón de OK!");
```

```
        }
```

```
        else
```

```
        {
```

```
          document.write("¡Has pulsado el botón de cancelar!");
```

```
        }
```

```
      }
```

```
    </script>
```

```
  </head>
```

```
  <body>
```

```
    <input type="button" onclick="disp_confirm()" value="Mostrar un confirm box" />
```

```
  </body>
```

```
</html>
```

Bucle for

```
<html>
  <body>
    <script type="text/javascript">
      var i=0;
      for (i=0;i<=10;i++)
      {
        document.write("El número es " + i);
        document.write("<br />");
      }
    </script>
  </body>
</html>
```

Resultado:

El número es 0

El número es 1

El número es 2

....

El número es 10

Bucle while

```
<html>
  <body>
    <script type="text/javascript">
      var i=0;
      while (i<=10)
      {
        document.write("El número es " + i);
        document.write("<br />");
        i=i+1;
      }
    </script>
  </body>
</html>
```

Resultado:

El número es 0

El número es 1

El número es 2

....

El número es 10

Bucle do...while

```
<html>
  <body>
    <script type="text/javascript">
      var i=0;
      do
      {
        document.write("El número es " + i);
        document.write("<br />");
        i=i+1;
      }
      while (i<0);
    </script>
  </body>
</html>
```

Resultado:
El número es 0

break

```
<html>
  <body>
    <script type="text/javascript">
      var i=0;
      for (i=0;i<=10;i++)
      {
        if (i==3)
        {
          break;
        }
        document.write("El número es " + i);
        document.write("<br />");
      }
    </script>
  </body>
</html>
```

Resultado:

El número es 0

El número es 1

El número es 2

continue

```
<html>
  <body>
    <script type="text/javascript">
      var i=0
      for (i=0;i<=4;i++)
      {
        if (i==3)
        {
          continue;
        }
        document.write("El número es " + i);
        document.write("<br />");
      }
    </script>
  </body>
</html>
```

Resultado:

El número es 0

El número es 1

El número es 2

El número es 4

for (variable in object)

```
<html>
  <body>
    <script type="text/javascript">
      var x;
      var mis_coches = new Array();
      mis_coches[0] = "Saab";
      mis_coches[1] = "Volvo";
      mis_coches[2] = "BMW";
      for (x in mis_coches)
      {
        document.write(mis_coches[x] + "<br />");
      }
    </script>
  </body>
</html>
```

Eventos

Todos los elementos de una página tienen asociados eventos que pueden disparar funciones de JavaScript.

Algunos de los más habituales:

<code>onload</code>	Cuando el usuario entra a la página
<code>onUnload</code>	Cuando el usuario abandona la página
<code>onMouseOver</code>	<code>onMouseOut</code> Para crear animaciones

Para formularios:

<code>onFocus</code>	Un elemento coge el foco
<code>onBlur</code>	Un elemento pierde el foco
<code>onChange</code>	Un elemento cambia
<code>onSubmit</code>	Para validar los campos de un formulario antes de enviarlos

try – catch - finally

```
<html>
  <head>
    <script type="text/javascript">
      var txt=""
      function mensaje()
      {
        try
        {
          adddlerert("¡Bienvenido a mi página!");
        }
        catch(err)
        {
          txt="Ha habido un error en esta página.\n\n";
          txt+="Error: " + err.description + "\n\n";
          txt+="Click OK para continuar.\n\n";
          alert(txt);
        }
      }
    </script>
  </head>
  <body>
    <input type="button" value="Ver mensaje" onclick="mensaje()" />
  </body>
</html>
```

throw

```
<html>
  <body>
    <script type="text/javascript">
      var x=prompt("Introduce un número entre 0 y 10:","");
      try
      {
        if(x>10)
          throw "Error1";
        else if(x<0)
          throw "Error2";
      }
      catch(er)
      {
        if(er=="Error1")
          alert("¡Error! El valor es muy alto");
        if(er == "Error2")
          alert("¡Error! El valor es muy bajo");
      }
    </script>
  </body>
</html>
```


Evento onerror

Este evento se dispara cuando hay un error de script en la página

Se crea una variable global onerror y se asocia a una función que maneje el error (realmente

La función se llama desde el manejador de eventos de onerror con 3 argumentos: msg (mensaje de error)
url (la url de la página que ha causado el error)
line (la línea en la que ha ocurrido el error).

```
onerror=handleErr
```

```
function handleErr(msg,url,l)
{
    //Handle the error here
    return true or false
}
```

Ejemplo captura de error con onerror

```
<html>
  <head>
    <script type="text/javascript">
      onerror=handleErr;
      var txt="";
      function handleErr(msg,url,l)
      {
        txt="Ha habido un error en esta página.\n\n";
        txt+="Error: " + msg + "\n";
        txt+="URL: " + url + "\n";
        txt+="Línea: " + l + "\n\n";
        txt+="Pulsa ok para continuar.\n\n";
        alert(txt);
        return true;
      }
      function mensaje()
      {
        adddlert("¡Bienvenido!");
      }
    </script>
  </head>
  <body>
    <input type="button" value="View message" onclick="mensaje()" />
  </body>
</html>
```