

PRIMERA CLASE

JAVASCRIPT es un lenguaje de programación que se utiliza para mejorar la presentación de nuestras páginas HTML. No tiene nada que ver con JAVA. No necesita compilación. Es un lenguaje interpretado. El código puede ser de tres tipos:

- Scripts inmediatos: que se ejecutan nada más cargar la página y van dentro del Body.
- Scripts diferidos: que se cargan con la página pero no se ejecutan hasta que el usuario hace algo (pulsar un botón, una tecla, etc). Van dentro del Body.
- Scripts híbridos: se definen tanto en el Head como en el Body.

Para incluir un script en nuestras páginas tenemos que usar la etiqueta cerrada `<SCRIPT>`. Su único parámetro (obligatorio) es *Language = Javascript*.

En Javascript no hay tipos de datos. Para definir una variable utilizamos la palabra clave *var*. Las sentencias se separan con punto y coma. Normalmente todas las instrucciones son en minúsculas. El nombre de una variable siempre debe empezar por una letra; no puede contener espacios en blanco ni puntos y su nombre puede ser tan largo como queramos.

EJEMPLO:

```
<HTML>
```

```
<HEAD>
```

```
<SCRIPT language = Javascript>
```

```
<!-- Comentario en HTML
```

```
function HolaMundo () {
```

```
  alert (Hola Mundo);
```

```
}
```

```
// -->
```

```
</SCRIPT>
```

```
</HEAD>
```

```
<BODY>
```

```
<FORM>
```

```
<INPUT type = button name = botón value = Púlsame onClick = HolaMundo ()·>
```

```
</FORM>
```

```
</BODY>
```

</HTML>

ÁMBITO DE UNA VARIABLE

Globales: son aquellas que se utilizan en todo el script y en todas las funciones.

Locales: solamente existen dentro de un bloque de código.

Parámetro de una función: son las mismas que las locales.

EJEMPLO:

<SCRIPT>

```
var a = 7;
```

```
function f1 () {
```

```
var b = 2;
```

```
}
```

```
function f2 (c) {
```

```
(sentencias)
```

```
}
```

</SCRIPT>

TIPOS DE OPERADORES

Aritméticos: +, -, *, /, %, ++, --. Algunas abreviaturas son +=, -=, *=, /=. (a = a+5 a+= 5).

Lógicos: ==, !=, <, >, <=, >=.

Relacionales: sirven para hacer expresiones complejas &&, ||, ! (AND, OR, NOT).

Separadores aritméticos: los corchetes y los paréntesis los utilizaremos para marcar prioridades; la coma separa variables del mismo tipo y los distintos parámetros que se les pasa a una función; y el punto y coma finaliza una línea de código.

CONTROL DE FLUJOS EN JAVASCRIPT

Principalmente hay tres tipos:

For: cuyo formato es

```
For (inicialización; condición; incremento) {
```

```
(sentencias)
```

```
}
```

EJEMPLO:

```
<SCRIPT language = Javascript>
```

```
var a;
```

```
for (a = 0; a<100; a++) {
```

```
document.write (a + );
```

```
}
```

```
</SCRITP>
```

while: tanto en el anterior como en este si la condición no se cumple la primera vez no se ejecuta el código. Su formato es

```
while (condición) {
```

```
(sentencias)
```

```
}
```

do while: el código se ejecutará en este caso al menos una vez. Su formato es

```
do {
```

```
(sentencias)
```

```
} while (condición);
```

if: cuyo formato es

```
if (condición) {
```

```
(sentencias)
```

```
}
```

```
[else {
```

```
(sentencias)
```

```
}};
```

break: fuerza la salida de un bloque de código.

EJEMPLO:

```
For (a = 0; a<100; a++) {
```

```
Documento.write (a + );
```

```
If (a == 49) {
```

```
Break};
```

```
}
```

OBJETOS DE JAVASCRIPT

Se definen objetos que están relacionados con la fecha actual, la ventana del navegador, etc. Cada objeto tiene por un lado unas propiedades (el color de fondo de una página por ejemplo) y por el otro lado tiene métodos (funciones que se pueden realizar con ese objeto). Sobre estos objetos se pueden producir eventos. Los eventos que se pueden producir en Javascript son:

onBlur: se produce cuando un elemento pierde el foco.

onClick: se produce cuando pinchamos sobre ese elemento.

onChange: se produce cuando el usuario cambia el valor de algún elemento.

onFocus: se produce cuando un elemento coge el foco.

onLoad: se produce al cargar esta página en el navegador.

onMouseOver: se produce cuando pasamos el ratón por encima.

onMouseOut: se produce cuando el ratón salga de ese elemento.

onSelect: se produce cuando seleccionamos un elemento.

onSubmit: se produce cuando pulsamos el botón Enviar de un formulario.

onUnload: se produce cuando el usuario abandona una web.

Los objetos integrados en Javascript son:

Anchor:

Button:

Checkbox:

Date:

Document:

Elements (array):

Form:

Frame:

Hidden:

History:

Link:

Location:

Math:

Navigator:

Password:

Radio:

Select:

Reset:

String:

Submit:

Text:

Textarea:

Window:

Los eventos se pueden producir en todo el documento y para que eso pase debemos aplicar este evento en la etiqueta Body. La segunda zona donde se produce un evento es en un formulario y para esto, el evento lo debemos incluir en la etiqueta Form. La tercera zona donde se pueden producir eventos es sobre un elemento eventual como un botón o un campo de texto de un formulario.

Cada elemento de un formulario puede llevar determinados eventos:

<SELECT> onBlur, onChange, onFocus.

Type = checkbox onClick

Type = radio onClick

Type = reset onClick

Type = button onClick

Type = submit onClick

Type text onBlur, onChange, onFocus, onSelect

<TEXTAREA> onBlur, onChange, onFocus, onSelect

Fuera de un formulario:

Documento <BODY> onLoad, onUnload

<FRAME> onLoad, onUnload

<A href> onClick, onMouseOver, onMouseOut

<FORM> onSubmit

SEGUNDA CLASE

OBJETO STRING

Es un tipo de objeto de Javascript que nos permite trabajar con cadenas de caracteres. La única propiedad que tiene es *length* la cual contiene el número de caracteres de ese objeto. La sintaxis para usar las propiedades es:

nombre objeto.propiedad

Los métodos de este objeto son:

Anchor () crea un elemento ancla (enlace) e irá donde le diga el objeto string. Con cadena, visualizaremos lo que queramos que se vea en la web.

Big () muestra la cadena a la que hacemos referencia con una fuente grande.

Blink () muestra la cadena parpadeando.

Bold () muestra la cadena en negrita.

CharAt (índice) muestra el carácter que esté en la posición que indique índice.

Fixed () muestra la cadena en un tipo de letra de ancho fijo.

Fontcolor () muestra esa cadena en el color especificado en formato HTML.

IndexOf (subcadena, inicio) devuelve la posición o el índice de la primera letra o subcadena que coincida con subcadena a partir de inicio.

Italics () muestra la cadena en letra cursiva.

LastIndexOf (subcadena) devuelve la posición de la última letra o subcadena que coincida con el parámetro que hemos pasado.

Link (URL) hace lo contrario que el método anchor. Nos aparecerá subrayado el contenido del objeto string.

Small () muestra la cadena con un tipo de fuente pequeño.

Strike () muestra la cadena con el texto tachado.

Sub () muestra la cadena en subíndice.

Sup () muestra la cadena en superíndice.

Substring (inicio, final) nos devuelve el texto contenido entre la posición inicial y final.

ToLowerCase () nos devuelve la cadena en minúsculas.

ToUpperCase () nos devuelve la cadena en mayúsculas.

EJEMPLO:

<BODY>

<SCRIPT language = Javascript>

<!

var frase = Hoy es Lunes;

*document.write (frase = + frase + `
');*

*document.write (Longitud = + frase.length + `
');*

*document.write (Negrita = + frase.bold + `
');*

*document.write (Negrita = + frase.bold () + `
');*

*document.write (Mayúsculas = + frase.toUpperCase () + `
');*

//-- >

</SCRIPT>

</BODY>

OBJETO MATH

Lo utilizaremos para efectuar cálculos matemáticos o funciones trigonométricas. Sus propiedades son:

E: contiene el número e.

LN2: contiene el logaritmo natural de 2.

LN10: contiene el logaritmo natural de 10.

LOG2E: contiene el logaritmo de e en base 2.

LOG10E: contiene el logaritmo de e en base 10.

PI: contiene el número pi.

SQRT1_2: contiene la raíz cuadrada de ½.

SQRT2: contiene la raíz cuadrada de 2.

Sus métodos son:

Abs (n) devuelve el valor absoluto de n.

Acos (n) devuelve el arcocoseno de n expresado en radianes.

Asin (n) devuelve el arcoseno de n expresado en radianes.

Atan (n) devuelve la arcotangente del ángulo n.

Ceil (n) devuelve el menor valor entero que es igual o mayor que n.

Cos (n) devuelve el coseno del ángulo n.

Exp (n) devuelve el número e elevado a n.

Floor () devuelve el valor entero mayor que sea igual o menor que n.

Log (n) calcula el logaritmo natural de n en base e.

Max (x,y) devuelve el mayor de x e y.

Min (x,y) devuelve el menor de x e y.

Pow (x,y) calcula la potencia x elevado a y.

Random () devuelve un número aleatorio entre cero y uno.

Round (n) devuelve el número redondeado.

Sin (n) devuelve el seno del ángulo n expresado en radianes.

Sqrt (n) devuelve la raíz cuadrada de n.

Tan (n) nos devuelve la tangente del ángulo n.

EJEMPLO: crear una web con un botón y al pulsarlo que salgan los números de la lotería primitiva.

OBJETO DATE

Se usa para trabajar con fechas y con horas. Por ejemplo:

Var fecha = new Date (); (Fecha y hora del equipo)

También se puede poner:

Var fecha = new Date (cadena); siendo cadena (día, mes, año, hh, mm, ss)

La tercera manera de ponerlo es:

Var fecha = new Date (año, mes, día, hh, mm, ss);

Este objeto no tiene propiedades. Sus métodos son:

GetDate () nos devuelve el día del mes. Un número entre 1 y 31.

GetDay () nos devuelve el día de la semana siendo el cero el domingo.

GetHours () nos devuelve la hora.

GetMinutes () nos devuelve los minutos comprendidos entre 0 y 59.

GetMonth () nos devuelve el número del mes siendo el cero enero.

GetSeconds () nos devuelve los segundos comprendidos entre 0 y 59.

GetTime () nos devuelve los milisegundos que han pasado desde el 1 de enero de 1970 hasta la fecha indicada.

GetYear () devuelve el año.

SetDate () establecemos el día del mes.

SetHours () establecemos las horas.

SetMinutes () establecemos los minutos.

SetMonth () establecemos el mes.

SetSeconds () establecemos los segundos.

SetTime (ms) devuelve la fecha equivalente a los milisegundos que le pasemos a partir del 1 de enero de 1970.

SetYear () establecemos el año.

ToLocaleString () nos devuelve una cadena que contiene la fecha actual.

Parse () le pasamos una cadena y lo convierte a un objeto de tipo date.

EJEMPLO:

<BODY>

<SCRIPT langua = Javascript>

var fecha = new Date ();

document.write (Hoy estamos a + fecha.getDate ());

document.write (del mes nº + fecha.getMonth () + 1);

document.write (del año + fecha.getYear () + 1900);

document.write (y son las + fecha.getHours () + horas + fecha.getMinutes () + Minutos);

</SCRIPT>

</BODY>

OBJETO WINDOW

Hace referencia a la ventana del navegador, al programa. Las propiedades de este objeto son:

DefaultStatus: contiene el mensaje a mostrar en la barra de estado del navegador.

Frames: es un array que contiene todos los frames de una página.

Length: contiene el número de frames.

Name: contiene el nombre de la ventana.

Parent: se utiliza para hacer referencia a alguna ventana que contenga la etiqueta frameset.

Self: hace referencia a la ventana activa.

Status: hace referencia a la barra de estado del navegador.

Top: hace referencia a una ventana superior del navegador.

Window: es lo mismo que self, hace referencia a la ventana activa.

Document: hace referencia al documento que estamos viendo en el navegador.

Frame: hace referencia al frame activo. Devolvería el nombre del frame activo.

Location: contiene la URL del documento que se muestra.

Los métodos son: (pudiendo poner window.método () o método a secas)

Alert (mensaje) muestra una ventana de alerta con el mensaje que pongamos y un botón de aceptar.

Close () cierra una ventana.

Confirm (mensaje) muestra una ventana con un botón de aceptar o cancelar.

Open (URL, nombre ventana, [opciones]) abre una nueva ventana y carga en esa ventana el documento que refleje URL. Con nombre ventana le asignamos un nombre para poder cerrarla después. Las opciones son:

Toolbar = yes/no :si queremos que la ventana nueva tenga la barra de herramientas.

Location = yes/no : hace referencia al campo donde introducimos la URL donde introducimos la URL de la página.

Directories = yes/no : hace referencia a los botones de directorio.

Status = *yes/no* : hace referencia a la barra de estado.

Menubar = *yes/no* : hace referencia a la barra de los menús.

Scrollbar = *yes/no* : si queremos que la ventana tenga barras de desplazamiento.

Resizable = *yes/no* : si queremos que la ventana se pueda redimensionar.

Width = *número* : ancho de la ventana.

Height = *número* : alto del navegador.

TERCERA CLASE

Prompt (mensaje [cadena]) nos muestra una ventana con un botón de aceptar, otro de cancelar y una caja de texto. Para recoger el dato introducido, este método se puede guardar en una variable. Por ejemplo: *var cad = prompt (Escribe tu nombre)*

SetTimeout (ms) retrasa la ejecución de una instrucción los milisegundos que le indiquemos. También hay que guardarlo en una variable.

Focus () coloca el enfoque en una determinada ventana. Hace referencia a la ventana activa. Por ejemplo: *ventana2.focus ()*.

Blur () quita el enfoque de la ventana activa que indiquemos.

Scroll () recorre la ventana.

OBJETO LOCATION

Este objeto hace referencia al campo dirección de la ventana del navegador. Además de ser independiente, es una propiedad del objeto window. No tiene ningún método. Sólo tiene propiedades. El formato es:

Protocol//hostname : port pathname search hash

Protocol hace referencia al tipo de protocolo incluyendo los dos puntos.

Hostname hace referencia a la dirección de la página siendo su número IP o su nombre del dominio.

Port hace referencia al puerto por el que se realiza la conexión. Por defecto en HTTP es 80.

Pathname hace referencia a la ruta del directorio de la página.

Search hace referencia a la ruta del directorio de la página. Es equivalente al parámetro TARGET de HTML.

Hash especifica un nombre de ancla (enlace a una misma página).

EJEMPLO:

<BODY>

* *

Al final de la página hay un botón

```
<HR>
```

```
<SCRIPT language = Javascript>
```

```
<!
```

```
for (var i = 1; i<40; i++) {
```

```
document.write(más abajo+'<BR><P>');
```

```
//-->
```

```
</SCRIPT>
```

```
<FORM name = formulario>
```

```
<INPUT type = button name = botón value = volver arriba onClick = location.hash = `top`>
```

```
</FORM>
```

```
</BODY>
```

EJEMPLO:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Nueva ventana </TITLE>
```

```
<SCRIPT language = javascript>
```

```
<!
```

```
function abrir () {
```

```
  miventana = open (, Mi_ventana, toolbar = no, directories = no, menubar = no);
```

```
  miventana.document.write(<HEAD><TITLE>Una nueva ventana</TITLE></HEAD>);
```

```
  miventana.document.write(<CENTER><H1><B>Cabeceras</B></h1> </CENTER>);
```

```
}
```

```
//-->
```

```
</SCRIPT>
```

```
</HEAD>
```

<BODY>

<FORM>

<INPUT type = button name = botón value = Pincha aquí onClick = abrir ()>

</FORM>

</BODY>

</HTML>

OBJETO DOCUMENT

Hace referencia al archivo HTML que vemos en el navegador. Las propiedades que tiene son:

Alinkcolor hace referencia al color de un enlace cuando pinchemos en él.

*Anchor*s es un array que contiene las anclas que hay en el documento.

Bgcolor hace referencia al color de fondo de nuestra página.

Cookie hace referencia a una cookie.

*Form*s es un array que contiene los formularios que hay en el documento.

LastModified hace referencia a la última fecha y hora de la modificación de la página.

LinkColor se refiere al color del enlace cuando aún no lo hemos visitado.

*Link*s es un array que hace referencia a los enlaces que hay en nuestra página.

Location contiene la URL de nuestro documento.

Title se refiere al título de la página.

Vlinkcolor se refiere al color de un enlace cuando ya lo hemos visitado.

Length (sólo para anchors, forms, links) contiene el número de elementos de ese array.

Los métodos son:

Clear () borra la página de nuestro navegador.

Close () cierra el documento pero no cierra la ventana.

Open () vuelve a hacer activos todos esos elementos.

Write () imprime una cadena en la página.

Writeln () imprime una cadena en la página haciendo un salto de línea aunque suele funcionar en sistemas operativos UNIX.

EJEMPLO: crear una página con un botón que ponga cambiar y que cada vez que pulsemos el botón nos cambie el color de fondo de la página.

OBJETO HISTORY

Contiene las direcciones que el usuario ha visitado en la ventana actual. Sólomente tiene una propiedad:

Length contiene el número de registros del objeto history.

Los métodos son:

Back () va a la página anterior.

Forward () va a la página siguiente.

Go () va al registro que nosotros le indiquemos.

Se puede hacer referencia a este objeto de tres maneras diferentes:

Document.history : ventana actual

Nombre marco.history

Ventana.history

OBJETO NAVIGATOR

Contiene información sobre el navegador que está utilizando el usuario. No tiene métodos y tiene tres propiedades:

AppCodeName contiene el código del navegador.

AppName contiene el nombre del navegador.

AppVersion contiene la versión del navegador.

EJEMPLO:

<BODY>

<SCRIPT language = Javascript>

document.write(Código: +navigator.appCodeName+'<P>');

document.write(Nombre: +navigator.appName+'<P>');

document.write(Versión: +navigator.appVersion+'<P>');

</SCRIPT>

</BODY>

OBJETO LINK

Se refiere a un texto o imagen que se hay incluido en la etiqueta `<A HREF>`. Las propiedades de cada objeto link son las mismas que el objeto location y tampoco tiene métodos.

Href hace referencia a lo que es la URL completa de una página

EJEMPLO:

```
<HTML>

<HEAD>

<TITLE> 2 Frames con 1 click </TITLE>

</HEAD>

<FRAMESET cols = 40%, *>

<FRAME name = fr1 src = pg1.htm>

<FRAMESEt rows = 50%, *>

<FRAME name = fr2 src = >

<FRAME name = fr3 src = >

</FRAMESET>

</FRAMESET>

<BODY> Sin frames </BODY>

</HTML>

<HTML>

<HEAD>

<TITLE> Página 1 </TITLE>

<SCRIPT language = Javascript>

<

function cargar (p1, p2){

parent.fr2.location.href = p1;

parent.fr3.location.href = p2;

}
```

```
//-->

</SCRIPT>

<BODY>

<FORM>

<INPUT type = button value = 2 en 1 onClick = cargar (`pg2.htm','pg3.htm')>

</FORM>

</BODY>

</HTML>
```

TERCERA CLASE

OBJET FORM

Es un objeto que nos permite denominar formularios y modificar o comprobar el valor de sus campos antes de enviarlos a un servidor. Podemos utilizar la *forms* que es un array para acceder al contenido de un formulario escribiendo:

Document.forms[n].nombre elemento.contenido

Document.formulario.nombre elemento.contenido

Todos los elementos de un formulario tienen dos propiedades: *name* y *value*.

Aparte, los elementos *checkbox* y *radio* tienen una propiedad más:

Checked contiene el estado de ese elemento. Si están marcadas nos va a devolver *true*. Si no, devolverán *false*.

EJEMPLO:

```
<HTML>

<HEAD>

<TITLE> Objeto FORM </TITLE>

<SCRIPT language = Javascript>

function ventana (){

var cadena = ;

if(documentn.forms[0].mayorDeEdad.checked){

cadena = mayor de edad;
```

```

}

else {

cadena = menor de edad;

}

alert(El nombre del campo de texto es:+document.forms[0].nombre.name+,su contenido
es:+document.formulario.nombre.value+\n y eres+cadena);

}

</SCRIPT>

</HEAD>

<BODY>

<CENTER>

<FORM name = formulario>

<INPUT type = text name = nombre> <P>

<INPUT type = checkbox name = mayorDeEdad checked> Mayor de Edad <P>

<INPUT type = button name = botón value = Pulsar onClick = ventana ()>

</FORM>

</CENTER>

</BODY>

</HTML>

```

COOKIES

Son un mecanismo de Javascript que permiten a un servidor enviar información a un cliente y recuperarla de él. Saber desde dónde nos han visitado, cuándo, etc. Otra de las ventajas es que podemos hacer que caduquen. El formato que tienen es el de una cadena de texto con un tamaño máximo de 4 Kb (4096 bytes). En nuestro disco duro puede haber como máximo 300 cookies y el servidor sólo puede tener un máximo de 20 en nuestro ordenador. El formato será:

Nombre = valor; [expires = caducidad]

Si no ponemos fecha de caducidad, se borrará al cerrar el navegador. En el Netscape, las cookies se almacenan en el fichero cookies.txt y en el Explorer todas las cookies se quedan en la carpeta cookies que cuelga de windows. Tenemos que tener una función que mande la cookie al servidor y otra función que permita recuperar esa cookie a nuestro ordenador.

EJEMPLO:

```
<HTML>

<HEAD>

<TITLE> Cookies </TITLE>

<SCRIPT language = Javascript>

function contador () {

var numero = 0;

var fecha = new Date(99,12,31);

if (!(numero = consultarCookie(VisitaALaPagina))){

numero = 0;

}

numero++;

mandarCookie(VisitaALaPagina, numero, fecha);

if(numero == 1) {

document.write(es la primera vez que me visitas);

}

else {

document.write(Has visitado la página+numero+veces);

}

}

function consultarCookie(nombre) {

var busqueda = nombre + =;

var i,f;

if(document.cookie.length>0){

i = document.cookies.indexOf(busqueda);

if(i! = -1){
```

```

i+ = busqueda.length;

j = document.cookie.indexOf(;;i);

if(j == -1) {

j = document.cookie.length;

}

return unescape(document.cookie.substring(i,j));

}

}

}

function mandarCookie(nombre, valor, caducidad) {

document.cookie = nombre++escape(valor)+((caducidad == null)?(;;expires =
+caducidad.toGMTstring()));

}

</SCRIPT>

</HEAD>

<BODY>

<SCRIPT language = Javascript>

contador();

</SCRIPT>

</BODY>

</HTML>

```