

Javascript

Apuntes de guerrilla



Index

Introducción en html	3-4
Variables	4-6
Operadores	6-7
Vectores	7-9
Matrices	9-10
Sentencias y estructuras	10-12
Vectores	10
Matrices	10
Códigos útiles cortos	11-13
Códigos de ejemplo	14-21

Introducir en html

Introducción en el mismo archivo html

Se puede definir en cualquier parte del archivo html, entre las etiquetas html, pero se especifica que se introduzca entre las etiquetas head.

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

Introducción en archivo .js

Se puede especificar los scripts en un archivo a parte.

```
<html>
  <head>
    <script type="text/javascript" src="scripts.js" />
  </head>
  <body>
  </body>
</html>
```

Comentar código javascript

Para que versiones antiguas de navegadores que no son compatibles con javascript es recomendable comentar nuestro código con las etiquetas de comentarios de html.

```
<head>
  <script type="text/javascript">
    <!--
      //aquí el código javascript
    -->
  </script>
</head>
```

Etiqueta noscript

En caso de que el usuario tenga desactivado javascript podremos definir un mensaje para mostrar añadiéndolo entre las etiquetas body.

```
<head>
  <script type="text/javascript">
    //Código javascript
  </script>
</head>
<body>
  <noscript>
    <p> Esta página contiene javascript, habilítalo </p>
  </noscript>
</body>
```

Comentarios

Los comentarios pueden ser de una línea con dos barras (//) o de varias líneas con una barra y asterisco (/*)

```
<script type="text/javascript">
  //Comentario de una línea
  /* Comentario enjaulado */
</script>
```

Variables

Tipos de variables

Tipo de datos	Información representada	Asignación
numérico	Datos enteros y decimales	var variable = 1 ;
cadena	Cadenas de texto de 1 a muchos caracteres.	var variable = "a"; var variable = "cadena";
booleano	Acepta true-false 0-1	var variable = true;
nulo	Variable nulo	var variable = null;
indefinidas	Simplemente no tienen valor	var variable;
No numéricos	Cuando se hacen operaciones con valores incompatibles. Da el valor NaN (Not a number)	var variable = 'cadena'-234;

Definir variable

Las variables se definen con var y su tipo se acopla al introducir un dato en ella, las variables son sensibles a minúsculas y mayúsculas.

```
<script type="text/javascript">  
    var variable;  
    var variable_numerica = 1;  
    var variable_cadena_texto = "texto";  
</script>
```

Calculos con variables

Las variables numéricas se pueden usar en cálculos matemáticos.

```
<script type="text/javascript">  
    var variable1 = 1, var variable2 = 2;  
    variable1 = variable1 + variable2;  
</script>
```

Carácteres especiales en cadenas

Los caracteres especiales que se encuentran dentro de las comillas dobles de una cadena se deben especificar con una contra barra.

```
<script type="text/javascript">  
    var variable = "Cadena mas \* otra cadena \*";  
</script>
```

Carácteres especiales en cadenas

Representación	Función
\n	Salto de línea
\t	Tabulación
\r	Retorno de carro (CR intro)
\b	Retroceder un carácter/espacio
\f	Salto de página
\0nnn	Carácter representado en octal, n es un valor entre 0 y 7
\xnn	“ “ hexadecimal, n es un valor entre 0 y F
\unnnn	“ “ unicode, n es un valor entre 0 y F
\a	Alarma
\f	Nueva página de impresora

Concatenar cadenas de texto

Se pueden concatenar las cadenas con comilals dobles y variables con strings fuera de las comillas dobles.

```
<script type="text/javascript">
  var variable = "Cadena numero 1";
  var variable2 = "Cadena numero 2";
  var variable3 = variable + variable2 + "Cadena numero 3";
</script>
```

Propiedad length

Para saber cuantos caracteres hay en un string para poder recorrerlas usaremos su atributo length.

```
string.length;
```

Operadores

Operadores numéricos

Operador	Uso	Operación
+	A+B	Suma
-	A-B	Resta
*	A*B	Multiplicación
/	A/B	División
%	A%B	Módulo o Resto
++	A++	Incremento
--	A--	Decremento
+=	A+=B	A = A + B
-=	A-=B	A = A - B

Operadores lógicos

Operador	Uso	Operación
&& o &	A&& B o A&B	A AND B.El resultado será true si ambos son true sino false
o	A B o A B	A OR B.EL resultado será false si ambos son false sino true
!	!A	Not A.Se invierte el resultado del operando. True es false.

Operadores racionales

Operador	Uso	Operación
<	A<B	A menor que B
>	A>B	A mayor que B
<=	A<=B	A mayor o igual que B
>=	A>=B	A mayor que o igual que B
!=	A!=B	A distinto que B
==	A == B	A igual que B

Vectores

Definir vectores

Definiremos el objeto array para definir un vector.

```
<script type="text/javascript">  
    var vector = new Array();  
</script>
```

Introducción de datos al definir

Introduciremos los valores entre comillas y todos dentro de un paréntesis. Los vectores son multitypos (pueden contener tanto strings como numeros).

```
<script type="text/javascript">  
    var vector = new Array("valor1","valor2",156);  
</script>
```

Introducción después de definir

Introduciremos los valores en sus índices correspondientes.

```
<script type="text/javascript">  
    var vector = new Array();  
    vector[0] = "valor1";  
    vector[1] = "valor2";  
    vector[2] = "valor2";  
    vector[3] = "valor4";  
    vector[4] = 5;  
</script>
```

Propiedad length

Para saber cuantas entradas hay en un vector para poder recorrerlas usaremos su atributo length.

```
vector.length;
```

Propiedad prototype

Puedes definir variables y métodos a un vector. Por ejemplo para pasar todo el vector a mayúsculas.

```
<script type="text/javascript">
  var vector = new Array();
  vector.prototype.cambiarMayusculas=function(){
      for (i=0;i<this.length;i++) {
          this[i]=this[i].toUpperCase();
      }
}

vector.cambiarMayusculas();
</script>
```

Funciones predefinidas de vectores

Funcion	Explicación	Uso
concat()	Concatena 2 o mas vectores	vector_con_vector1_vector2 = vector1.concat(vector2);
indexOf()	Devuelve el index dentro del vector del valor	vector.indexOf("valor5");
lastIndexOf()	Devuelve el index empezando por el final del vector	vector.lastIndexOf("valor3");
join()	Convierte el vector string, separado los valores por el valor asignado en el paréntesis	vector.join(",");
pop()	Elimina la última entrada del vector y devuelve su contenido	vector.pop();
shift()	Elimina la primera entrada del vector y devuelve su contenido	vector.shift();
unshift()	Añade los valores asignados entre los paréntesis al principio del vector y devuelve el valor de length	Vector.unshift();

push()	Introduce el valor asignado entre los paréntesis al final del vector	vector.push("valor_ultimo");
slice()	Devuelve los valores comprendidos entre los index asignados entre los paréntesis devolviendo un vector	vector1 = vector2(4,23);
splice()	Lo mismo que hace slice pero elimina el contenido del primer vector	vector1 = vector2(4,23);
sort()	Ordena alfabéticamente o por valor numérico el contenido del vector	vector.sort();
toString()	Convierte el vector en un string separado por comas	vector.toString();
reverse()	Invierte el orden de los valores	vector.reverse();
valueOf()	Devuelve el mismo vector en forma de vector, copia el vector	vector1 = vector2.valueOf();

Matrices

Definir matriz

Javascript no directamente las matrices pero se puede asignar un vector dentro de una posición de un vector.

```
<script type="text/javascript">
  var matriz = new Array();
  matriz[0] = new Array("valor1","valor2");
</script>
```

Estructuras de control

Estructura if

Para poder comparar dos variables o valores usaremos la sentencia if.

```
<script type="text/javascript">  
    if(variable1 == variable 2){  
        //sentencias a ejecutar  
    }  
</script>
```

Estructura if...else

Para usar un if...else lo escribimos de la siguiente forma.

```
<script type="text/javascript">  
    if(variable){  
        //sentencias a ejecutar  
    }else{  
        //sentencias a ejecutar si no...  
    }  
</script>
```

Estructura if...else if

Lo intrudciremos de la siguiente forma.

```
<script type="text/javascript">  
    if(variable1 != variable 2){  
        //sentencias a ejecutar  
    }else if (variable1 <= 0){  
        //sentencias a ejecutar sí si...  
    }  
</script>
```

Condiciones anidadas

Para anidar en un if diferentes opciones usaremos los caracteres especiales.

```
<script type="text/javascript">  
    if( (variable1 != variable 2) && (variable1 == variable3) ){  
        //sentencias a ejecutar  
    }  
</script>
```

Estructura switch case

La estructura switch la formaremos así.

```
<script type="text/javascript">  
    switch (variable){  
        case "valor1":  
            //sentencias a ejecutar  
            break;  
        case 1:  
            //sentencias a ejecutar  
            break;  
        Default:  
            //sentencias a ejecutar si no se produce  
            ninguna de las anteriores  
    }  
</script>
```

Bucle while

El bucle while lo hacemos de la siguiente forma.

```
<script type="text/javascript">  
    while(variable1 == 0){  
        //sentencias a ejecutar  
    }  
</script>
```

Bucle do...while

El bucle while lo hacemos de la siguiente forma.

```
<script type="text/javascript">  
    do{  
        //sentencias a ejecutar  
    } while(variable1 == 0)  
</script>
```

Bucle for

El bucle for lo hacemos de la siguiente forma.

```
<script type="text/javascript">  
    for ( i=0 ; i < 5 ; i++ ){  
        //sentencias a ejecutar  
    }  
</script>
```

Bucle for in

Con este bucle podremos acceder a los valores de un vector.

```
<script type="text/javascript">  
    for (i in vector){  
        vector[i];  
    }  
</script>
```

Funciones de entrada y salida

Función prompt

Para la entrada de datos tenemos la función prompt() que nos mostrará un cuadro de dialogo que nos dejará introducir un dato, lo que pongamos entre el paréntesis se mostrará en la ventana en forma de explicación.

```
<script type="text/javascript">  
    var entrada = prompt("Este texto saldrá en la ventana");  
</script>
```

Función alert

Para la salida la función alert() que nos mostrara el contenido de los paréntesis en una ventana.

```
<script type="text/javascript">  
    alert("Este texto saldrá en la ventana");  
</script>
```

Crear funciones

Creando función simple

Para crear una función simple usaremos function, el nombre de la función y en caso de tener parámetros definir-los en los paréntesis. Después llamaremos a la función con su nombre y los paréntesis.

```
<script type="text/javascript">  
    function nombre_funcion (parametro1,parametro2){  
        //sentencias a ejecutar  
    }  
    nombre_funcion(1,variable);  
</script>
```

Creando objeto function

En javascript existe el objeto function, se puede crear añadiendo los parámetros y sentencias añadiendolo al constructor.

```
<script type="text/javascript">  
    var suma = new Function("x", "y", "return (x + y)");  
    alert(suma(2,3));  
</script>
```

Retorno de la función

Para retornar una variable, una cadena o lo que se necesite usaremos return.

```
<script type="text/javascript">  
    var variable = 2;  
    function nombre_funcion (parametro1,parametro2){  
        return (parametro1 + parametro2);  
    }  
    alert(nombre_funcion(1,variable));  
</script>
```

Llamar a variable ajena, this

Para acceder a variables y funciones de afuera de la función usaremos el prefijo `this`.

```
<script type="text/javascript">
  var variable;
  function funcion1(){ alert("mensaje") };

  function funcion2(){
    this.funcion1();
    this.variable;
  }
</script>
```

Pasar función anónima a variable

Podemos pasar una función anónima a una variable para luego ejecutarla como si de otra función se tratase.

```
<script type="text/javascript">
  Function nombre_funcion(){
    return function(){
      alert("Texto");
    }
  }
  var variable = nombre_funcion();
  variable();
</script>
```

Crear clases

Crear una pseudoclase

Javascript no soporta clases o no las implementa, por lo tanto no podemos crearlas, pero se pueden anidar funciones y variables en un objeto de función, y a este podemos añadirle funciones y variables como si fuera una clase, este no tendrá todos los beneficios como un objeto normal.

```
<script type="text/javascript">  
  function pseudoclase() {  
    var color;  
    var numero;  
    this.getColor= function () {  
      return this.color;  
    }  
    this.getNumero = function(){  
      return this.numero;  
    }  
    this.setColor = function(dcolor){  
      this.color = dcolor;  
    }  
    this.setNumero = function(dnumero){  
      this.numero = dnumero;  
    }  
  }  
</script>
```

Crear objetos

Crear objeto de la pseudoclase

Después de crear la función podemos crear objetos de ella como si de una clase se tratase.

```
<script type="text/javascript">  
  //Aquí debe estar la definición anterior de la pseudoclase  
  var objeto = new pseudoclase();  
  objeto.setNumero(1);  
  objeto.setColor("Rojo");  
  alert(objeto.getColor() + objeto.getNumero());  
</script>
```

Añadir propiedad o método a objeto

Con la creación del objeto tenemos la propiedad prototype para poder crear variables o métodos en el objeto directamente en el objeto.

```
<script type="text/javascript">
  //Teniendo definida la pseudoclase "persona".
  var manolo=new persona();
  manolo.prototype.dni;
  manolo.dni="23232323b";
  manolo.prototype.saludar = function(){
    alert("Hola");
  }
  alert(manolos.dni);
  manolo.saludar();
</script>
```


Objeto String

Funciones de String

Funcion	Descripción	Uso
charAt()	Devuelve el carácter que está en el index indicado del string.	cadena.charAt("1")
charCodeAt()	Devuelve el código UNICODE del carácter especificado en el index.	cadena.charCodeAt("1")
concat()	Concatena el string especificado con los introducidos en el paréntesis y devuelve la unión.	cadena1.concat(cadena2)
fromCharCode()	Transforma el código UNICODE en carácter .	cadena.fromCharCode(12)
indexOf()	Devuelve el index del carácter especificado en los paréntesis. Si el carácter no está en el string devuelve -1.	cadena.indexOf("a")
lastIndexOf()	Lo mismo que indexOf() pero empezando a contar por el final.	cadena.lastIndexOf("a")
match()	Utiliza la expresión regular que se introduce en los paréntesis para devolver el resultado.	cadena.match(regex)
replace()	Reemplaza el primer valor , o expresión regular, introducido en los paréntesis por el segundo valor introducido.	cadena.replace("valor1", "valor2")
search()	Busca la expresión regular o cadena dentro del string y devuelve el index de su primera posición.	cadena1.search("cadena2")
slice()	Recorta el string empezando por la posición de index especificada y acabando en la segunda posición de index especificada. Devuelve un nuevo string.	Cadena.slice(1,4)
split()	Crea un vector separando con el carácter especificado entre los paréntesis.	cadena.split(",")
substr()	Igual que slice()	cadena.substr(5,10)
substring()	Igual que slice() pero el primer número especifica donde empieza el string devuelto y el segundo la posición donde termina.	cadena.substring(4,6)
toLowerCase()	Convierte todos los caracteres de un string a minúsculas.	cadena.toLowerCase()
toUpperCase()	Convierte todos los caracteres de un string a mayúsculas.	cadena.toUpperCase()
valueOf()	Devuelve el contenido del string	cadena.valueOf()

Funciones de visualización en propiedades html		
anchor()	Muestra la cadena como un ancla con la etiqueta name con el valor especificado entre los paréntesis.	cadena.anchor("nombre")
big()	Muestra la cadena con un tamaño mayor.	cadena.big()
bold()	Muestra el string en negrita.	cadena.bold()
blink()	Muestra la cadena con parpadeo.	cadena.blink()
fixed()	Muestra la cadena con un tamaño mas pequeño.	cadena.fixed()
fontColor()	Muestra la cadena con el color especificado entre los paréntesis.	cadena.fontColor("green")
fontSize()	Muestra la cadena con el tamaño especificado entre los paréntesis.De 1 a 7	cadena.fontSize(1)
italics()	Muestra el string en cursiva.	Cadena.italics()
link()	Muestra la cadena como un enlace, con el atributo href especificado en el paréntesis.	cadena.link("url")
small()	Muestra la cadena mas pequeña.	cadena.small()
strike()	Muestra la cadena tachada.	cadena.strike()
sub()	Muestra la cadena posicionada hacia abajo.	cadena.sub()
sup()	Lo mismo que sub() pero hacia arriba.	cadena.sup()

Ejemplos de uso

Separar numero de letra del dni

Validamos dni comprobando que tiene 9 caracteres y separamos los números (8) de la letra).

```
<script type="text/javascript">
    var dni = prompt("Introduce el dni");
    If( dni.length == 9 )
        var letra = dni.substring(8,9);
        var numeros = dni.substring(0,8);
    }
    alert("Los numeros del dni son: " + numeros + " y la letra es: + "
        letra");
</script>
```

Transformar texto a minúsculas y en enlace

Validamos dni comprobando que tiene 9 caracteres y separamos los números (8) de la letra).

```
<script type="text/javascript">
    var cadena = "HTTP://WWW.GOOGLE.ES";
    var enlace = cadena.toLowerCase(cadena);
    var cadena2 = "ir a google";
    alert(cadena2.link(enlace));
</script>
```

Substituir comas por espacios

Para remplazar todas las comas tendremos que usar la función replace tantas veces como comas haya en el string.

```
<script type="text/javascript">
    var cadena = "cadena,con,comas";
    while( cadena.indexOf(",") != -1 ){
        cadena = cadena.replace(","," ");
    }
    alert(cadena);
</script>
```

Objeto Number

Propiedades de Number

Propiedad	Descripción	Uso
MAX_VALUE	Devuelve la cantidad de caracteres que puede contener el número.	numero.MAX_VALUE
MIN_VALUE	Lo mismo pero con el número mas bajo.	numero.MIN_VALUE
NEGATIVE_INFINITY	Devuelve un negativo infinito	numero.NEGATIVE_INFINITY
POSITIVE_INFINITY	Devuelve un positivo infinito.	numero.POSITIVE_INFINITY
NaN	Comprueba si el valor del numero es NaN.	nodo.NaN

Funciones de Number

Función	Descripción	Uso
toExponential()	Devuelve el numero en notación exponencial con el recorte especificado entre los paréntesis.	numero.toExponential(10)
toFixed()	Recorta los decimales especificando la cantidad deseada entre los paréntesis.	numero.toFixed(3)
toPrecision()	Devuelve el número recortado hasta el número especificado entre los paréntesis.	numero.toPrecision(5)
toString()	Convierte el número a string.	numero.toString()
valueOf()	Devuelve el contenido del número.	numero.valueOf()

Ejemplos de uso

Recortar decimales de resultado

Si queremos recortar los decimales por completo de una operación que da como resultado decimales no deseados usaremos toFixed().

```
<script type="text/javascript">
  var numero1 = 50;
  var numero2 = 23;
  var numero3 = numero1/numero2;
  numero3 = numero3.toFixed(0);
  alert(numero3);
</script>
```

Objeto Math

Propiedades de Math

Propiedad	Descripción	Uso
E	Devuelve el numero de euler.	math.E
LN2	Devuelve el logaritmo natural de 2.	math.LN2
LN10	Devuelve el logaritmo natural de 10.	math.LN10
LOG2E	Devuelve la base 2 del logaritmo de E.	math.LOG2E
LOG10E	Devuelve la base 10 del logaritmo de E.	math.LOG10E
PI	Devuelve el PI.	math.PI
SQRT1_2	Devuelve la raíz cuadrada de ½.	math.SQRT1_2
SQRT2	Devuelve la raíz cuadrada de 2.	math.SQRT2

Funciones de Math

Función	Descripción	Uso
pow()	Devuelve el primer número especificado entre los paréntesis potenciado por el segundo numero especificado.	Math.pow(3,5)
arcos()	Devuelve el arccoseno del número especificado entre los paréntesis.	Math..arcos(0.8)
arcsine()	Devuelve el arcoseno del número especificado entre los paréntesis.	Math.asin(5)
atan()	Devuelve el arcotangente del número especificado entre los paréntesis.	Math.atan(3)
atan2()	Calcula el angulo de la radiante entre los dos puntos especificados.	Math.atan2(3,5)
ceil()	Redondea un decimal hacia arriba.	Math.ceil(5.3)
floor()	Redondea un decimal hacia abajo.	
exp()	Devuelve E elevado al número especificado entre los paréntesis.	Math.exp(2)
abs()	Devuelve el valor absoluto especificado en los paréntesis.	Math..abs(10)
log()	Devuelve el logaritmo natural en base E.	Math.log(2)
max()	Devuelve el número con mayor valor de todos los especificados entre ellos paréntesis y separados por comas.	Math.max(23,6,10,45,)
min()	Igual que max() pero con el número de menor valor	Math.min(4,12,32)

round()	Redondea el decimal especificado entre los paréntesis al entero mas próximo.	Math.round(2.6)
random()	Devuelve un número decimal aleatorio de 0 a 1.	Math.random()
cos()	Devuelve el coseno del número especificado entre los paréntesis.	Math.cos(5)
sin()	Devuelve el seno del número especificado entre los paréntesis.	Math.sin(9)
sqrt()	Devuelve la raíz cuadrada el número especificado entre los paréntesis.	Math.sqrt(8)
tan()	Devuelve la tangente del angulo especificado entre los paréntesis.	Math.tan(180)

Ejemplos de uso

Calcular máximo y mínimo de los números de un vector

Para calcularlo tendremos que especificar las entradas del vector separadas por comas dentro de las funciones max() y min().

```
<script type="text/javascript">
    var numero = new Array("10", "50", "30");
    alert(Math.max(numero[0], numero[1], numero[2]));
    alert(Math.min(numero[0], numero[1], numero[2]));
</script>
```

Simular tirada de un dado de 6 caras

Usaremos la función random para generar el aleatorio.

```
<script type="text/javascript">
    var aleatorio = Math.random();
    aleatorio = aleatorio * 10;
    aleatorio %= 5;
    aleatorio += 1;
    alert(aleatorio.toFixed(0));
</script>
```

Objeto Date

Funciones de Date

Función	Descripción	Uso
getDate()	Devuelve el día del mes (1-31).	fecha.getDate()
getMonth()	Devuelve el número del mes (0-11).	fecha.getMonth()
getFullYear()	Devuelve el año entero con 4 cifras.	fecha.getFullYear()
getMilliseconds()	Devuelve los milisegundos (0-999).	fecha.getMilliseconds()
getSeconds()	Devuelve los segundos (0-59).	fecha.getSeconds()
getMinutes()	Devuelve los minutos (0-59).	fecha.getMinutes()
getHours()	Devuelve la hora (0-23).	fecha.getHours()
getDay()	Devuelve el día de la semana (0-6).	fecha.getDay()
getTime()	Devuelve los milisegundos pasados desde las 12 de la noche del 1 de Junio de 1970.	fecha.getTime()
setDate()	Establece el día del mes.	fecha.setDate(6)
setMonth()	Establece el mes.	fecha.setMonth(0)
setFullYear()	Establece el año.	fecha.setFullYear(2013)
setMilliseconds()	Establece los milisegundos:.	fecha.setMilliseconds(450)
setSeconds()	Estableces los segundos.	fecha.setSeconds(34)
setMinutes()	Estable los minutos.	fecha.setMinutes(12)
setHours()	Establece la hora.	fecha.setHours(0)
setTime()	Establece el tiempo transcurrido en milisegundos desde las 12 de la noche del 1 de Junio de 1970.	fecha.setTime(21341241)
toDateString()	Devuelve un string con la fecha.	fecha.toDateString()
toString()	Devuelve un string con la fecha y hora.	fecha.toString()
toTimeString()	Devuelve un string con la hora.	fecha.toTimeString()

Ejemplos de uso

Mostrar la hora

Para poder usar una fecha tenemos que definir un objeto Date. Después usamos la función `toString()` pero recortaremos el string para tener solamente la hora ya que esa función nos devuelve mas datos.

```
<script type="text/javascript">  
    var fecha = new Date();  
    var hora = fecha.toString();  
    hora = hora.substring(0,8);  
    alert(hora);  
</script>
```

Mostrar la fecha

Para mostrar la fecha usaremos los getters del objeto date y cambiaremos sus valores para una lectura mas humana.

```
<script type="text/javascript">  
    var dias = new Array  
        ("domingo","lunes","martes","miercoles",  
         "jueves","viernes","sabado");  
    var meses = new Array  
        ("enero","febrero","marzo","abril","mayo","junio",  
         "julio","agosto","septiembre",  
         "octubre","noviembre","diciembre");  
    var fecha = new Date();  
    var diasemana = fecha.getDay();  
    var diames = fecha.getDate();  
    var mes = fecha.getMonth();  
    var anyo = fecha.getFullYear();  
    alert("Estamos a " + dias[diasemana] + " dia " + diames + " de " +  
        meses[mes] + " del " + anyo);  
</script>
```


Objeto RegExp

Estructura de expresión regular

Una expresión regular sirve para comparar una cadena con un patrón de caracteres, este patrón se creara con una codificación específica. Para crear la expresión podemos crear un objeto de regex o añadir el patrón a una variable.

```
<script type="text/javascript">
  //Crear objeto regex
  var expresion = new reg(patron,modificador);
  //asignarlo a una variable
  var expresion2 = /patron/modificador;
</script>
```

Modificadores

Modificador	Descripción	Uso
i	El patrón no tendrá sensibilidad hacia las mayúsculas o minúsculas.	/patron/i
g	La expresión devolverá todas las coincidencias.	/patron/g
m	Realiza coincidencias en varias líneas.	/patron/m

Corchetes y paréntesis

Tipo	Descripción	Uso
[xyz]	Cualquier carácter (letras minúsculas y mayúsculas , cifras o caracteres especiales) que este entre los corchetes será buscado individualmente y retornado.	/[AaZz2]/
[^xyz]	Igual que el anterior pero este devolverá los que no coincidan con los caracteres o números indicados.	/[^abcd]/
[x-y]	Comparará con cualquier carácter que esté en el rango puesto entre x – y.	/[0-9]/
(x y)	Buscará una de las alternativas especificadas.	/(1 2)/

Metacaracteres

Metacaracter	Descripción	Uso
.	Busca un solo carácter de cualquier tipo a excepción de una nueva línea o fina de línea.	/A./
\w	Busca cualquier carácter que sea de tipo letra.	/\w/
\W	Busca cualquier carácter que no sea de tipo letra.	/\W/
\d	Busca cualquier carácter que sea de tipo cifra.	/\d/
\D	Busca cualquier carácter que no sea de tipo cifra.	/\D\
\s	Busca cualquier espacio en blanco.	/\s/
\S	Busca cualquier carácter que no sea un espacio en blanco.	/\S/
\b	Busca la ocurrencia al principio o final de cualquier palabra.	/\b/
\B	Busca la ocurrencia que no este al principio o final de cualquier palabra.	/\B\
\0	Busca los caracteres NULL.	/\0/
\n	Busca las nuevas líneas.	/\n/
\f	Busca los avances de página.	/\f/
\r	Busca los carros (tecla intro).	/\r/
\t	Busca los tabulados.	/\t/
\v	Busca los tabulados verticales.	/\v/
\nnn	Busca el carácter especificado en octal.	/\543/
\xnn	Busca los números en hexadecimal.	/\x09/
\unnnn	Busca el carácter en UNICODE	/\u4244/

Cuantificadores

Expresión	Descripción	Uso
n+	El carácter o combinación de caracteres se tendrán que repetir al menos una vez en la cadena.	/P+/
n*	El carácter o combinación de caracteres se encontrarán en la cadena 0 o mas veces.	/9*/
n?	El carácter o combinación de caracteres se encontrarán 0 o 1 vez.	/\{Ab\}?
n{x}	El carácter o combinación de caracteres se repetirá las veces que se especifique entre los paréntesis.	/B{3}/
n{x,y}	El carácter o combinación de caracteres se repetirá el rango veces que se especifique entre los paréntesis.	/12{1,5}/

n{x,}	El carácter o combinación de caracteres se repetirá de lo que se especifique entre los paréntesis a infinito..	/U{3,}/
n\$	El carácter o combinación de caracteres se buscará que coincida con el final de la cadena.	/f2\$/
^n	El carácter o combinación de caracteres se buscará que coincida con el principio de la cadena.	/^ho/
?=n	El carácter o combinación de caracteres que tenga a continuación en cualquier posición la combinación de n.	/ho(=?= la)/
?!n	El carácter o combinación de caracteres que no tenga a continuación en cualquier posición la combinación de n.	/ho(?!lo)/

Propiedades de RegExp

Propiedad	Descripción	Uso
global	Devuelve true si el modificador g está activado en el patrón.	patron.global
ignoreCase	Devuelve true si el modificador i está activado en el patrón.	patron.ignoreCase
multiline	Devuelve true si el modificador m está activado en el patrón.	patron.multiline
lastIndex	Devuelve el index de las ocurrencias encontradas en la cadena.	patron.lastIndex
source	Devuelve el texto especificado del patrón.	patron.expresión

Funciones de RegExp

Función	Descripción	Uso
compile()	Compila la expresión del patrón introducido entre los paréntesis.	patron.compile(patron)
exec()	Ejecuta el patrón en la cadena especificada entre los paréntesis, devuelve la primera coincidencia.	patron.exec(cadena)
test()	Ejecuta el patrón en la cadena especificada entre los paréntesis, devuelve true en caso de encontrar la ocurrencia y false en caso contrario.	patron.test(cadena)

Ejemplos de uso

Almacenar ocurrencia de cadena

En este ejemplo sacaremos todos los números que se encuentren en la cadena y los mostraremos.

```
<script type="text/javascript">
  var cadena = "De este párrafo tan solo queremos sacar los
  numeros que aquí se guardan 12321.";
  var patron=/[0-9]+/;
  alert(patron.exec(cadena));
</script>
```

Validar cadena

Podremos validar una cadena para ver si cumple los requisitos explícitos del patrón. En este caso validaremos la introducción de un dni.

```
<script type="text/javascript">
  var dni = prompt("Introduce los 8 numeros y la letra del dni");
  var patron=/^[0-9]{8}[a-zA-Z]{1}$/;
  if (patron.test(dni)){
    alert("El dni " + dni + " es correcto.");
  }else{
    alert("El dni " + dni + " no es correcto.");
  }
</script>
```

En este caso validaremos un correo electrónico. Solo se permitirán letras mayúsculas y minúsculas, dígitos, guión medio y bajo. El tamaño máximo de la parte de nombre será hasta 30 caracteres y el del dominio será hasta 30 también.

```
<script type="text/javascript">
  var email = prompt("Introduce tu correo, recuerda que tanto el
  nombre del correo como el dominio solo
  pueden contener los siguientes caracteres
  especiales: - (guión medio) _(guión bajo)");
  var patron=/^[0-9A-Za-z_\-]{1,30}@{1}[0-9A-Za-z_\-]{1,30}\.
  {1}[a-zA-Z]{3}$/;
  if (patron.test(email)){
    alert("El email " + email + " es correcto.");
  }else{
    alert("El email " + email + " no es correcto.");
  }
</script>
```

Remplazar cadena por otra

Combinando la expresión regular y la función replace podemos cambiar las ocurrencias que nosotros deseemos por otras. En este caso cambiaremos un número de teléfono por otro.

```
<script type="text/javascript">
var cadena = "El número de telefono es 645434211";
var patron=/\b(9|6){1}[0-9]{8}/;
cadena = cadena.replace(patron,"933333333");
alert(cadena);
</script>
```

Lista de expresiones regulares útiles

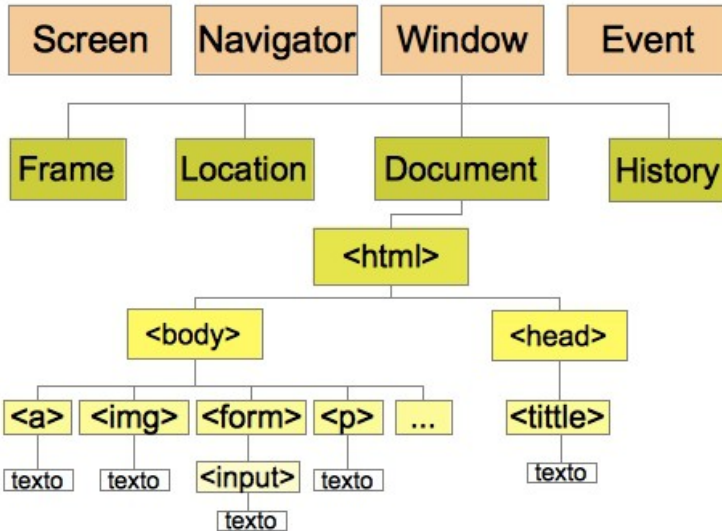
Aquí una lista con algunas expresiones regulares útiles para programación web.

Expresión	Descripción
<code>^(https?:\w)?([\da-z\.-]+)\.([a-z.]{2,6})([\w \?=-]*)*\w?\$/</code>	Validar URL
<code>^[_a-z0-9-]+(\.[_a-z0-9-]+)*@[a-z0-9-]+(\.[a-z0-9-]+)*\.[a-z]{2,3}\$</code>	Validar email con puntos, guiones bajos y medios.
<code>(?=^.{8,}\$)((?=.*d)((?=.*W+))(![.!\n])(?=.*[A-Z])(?=.*[a-z]).*\$</code>	Validar contraseña segura, debe contener al menos una minúscula, mayúscula y un dígito o carácter especial, también debe ser de al menos 8 caracteres.
<code>\b(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\.(25[0-5] 2[0-4][0-9] [01]?[0-9][0-9]?)\b/</code>	Validar IP. De 0.0.0.0 a 255.255.255.255
<code>/.*(<.*?>)/</code>	Cambiando la "etiqueta" por el nombre que queramos buscaremos etiquetas XHTML.
<code>^\+?\d{1,3}?[-.]?(?:(\d{2,3}))?[-.]?\d\d\d[-.]?\d\d\d\d\$</code>	Valida cualquier número de teléfono.
<code>/^#(?:[a-f\d]{3}){1,2}\$/i</code>	Valida colores hexadecimales (#333, 333, #333333 o 333333)

DOM (Document object model)

Estructura de DOM

El documento html se estructura como un árbol en el que las etiquetas html se anidan en otras etiquetas y de esta forma crean las ramas del árbol.



Para acceder a el texto del elemento "<a>" tendríamos que acceder primero a window/document/<html>/<body>/<a>/texto de la misma forma que accedemos a un sistema de ficheros estructurado por carpetas. Realmente no será necesario nombrar para acceder a todos los elementos, ya que algunos se pueden obviar.

Objeto Window

Propiedades de window

Propiedad	Descripción	Uso
closed	Devuelve true si la ventana se ha cerrado.	window.closed
status	Devuelve el texto de la barra de estado.	window.status
top	Devuelve true si se trata de la ventana principal, en caso de encontrarse en un frame o iframe devuelve false.	window.top
self	Devuelve la ventana actual.	window.self
innerHeight	Devuelve o establece la altura de la ventana.	window.innerHeight
innerWidth	Devuelve o establece la anchura de la ventana.	window.innerWidth
length	Devuelve el número de frames y iframes de la ventana.	window.length
name	Devuelve el nombre de la ventana.	window.name
parent	Devuelve la ventana padre de la ventana actual.	window.parent
opener	Devuelve la referencia de la página desde donde se abrió la página actual.	window.opener
outerHeight	Devuelve o establece la altura de la ventana contando scrollbars y toolbars.	window.outerHeight
outerWidth	Devuelve o establece la anchura de la ventana contando scrollbars y toolbars.	window.outerWidth
pageXOffset	Devuelve los pixels de desplazamiento empezando por la esquina interior izquierda de arriba de forma horizontal.	window.pageXOffset
pageYOffset	Devuelve los pixels de desplazamiento empezando por la esquina interior izquierda de arriba de forma vertical.	window.pageYOffset
location	Devuelve el objeto location.	window.location
screen	Devuelve el objeto screen.	window.screen
document	Devuelve el objeto document	window.document
frames	Devuelve todos los frames y iframes de la ventana.	window.frames[]
history	Devuelve el objeto history.	window.history
navigator	Devuelve el objeto navigator.	window.navigator
locationbar	Devuelve objeto de la barra de direcciones.	window.locationbar
menubar	Devuelve objeto del menú de la ventana.	window.menubar

personalbar	Devuelve el objeto de la barra personal de la ventana.	window.personalbar
scrollbars	Devuelve el objeto de las barras de desplazamiento de la ventana.	window.scrollbars
statusbar	Devuelve objeto de la barra de estado.	window.statusbar
event	Devuelve los eventos de la ventana.	window.event
screenLeft	Devuelve la coordenada x de la ventana relativa a la pantalla.	window.screenLeft
screenX	Devuelve la coordenada x de la ventana relativa a la pantalla.	window.screenX
screenTop	Devuelve la coordenada y de la ventana relativa a la pantalla	window.screenTop
screenY	Devuelve la coordenada y de la ventana relativa a la pantalla	window.screenY

Funciones de window

Función	Descripción	Uso
alert()	Crea una ventana emergente con el texto que se introduzca entre los paréntesis.	alert("Error")
prompt()	Crea una ventana para introducción de datos con el texto que se introduzca entre los paréntesis	prompt("introduzca su nombre")
confirm()	Crea una ventana de confirmación con el texto que se introduzca entre los paréntesis, en caso de clicar sobre aceptar devolverá true, si es cancelar false.	confirm("Desea continuar?")
find()	Muestra una ventana de búsqueda.	find()
blur()	Quita el foco de la ventana actual.	window.blur()
focus()	Da el foco a la ventana actual.	windows.focus()
close()	Cierra la ventana actual.	window.close()
home()	Vuelve a la página por defecto especificada por el navegador.	home()
stop()	Para la carga de la página.	stop()
open()	Abre una nueva ventana. Especificamos dentro de los paréntesis los parámetros en este orden: URL,nombre,especificaciones,remplaza(opcional)	window.open("http://lapagina.es","width=200,height=200")

print()	Devuelve el contenido de la ventana actual.	window.print()
moveBy(x,y)	Mueve la ventana a las coordenadas especificadas entre los paréntesis según la posición actual.	window.moveBy(23,43)
moveTo(x,y)	Mueve la ventana a las coordenadas especificadas entre los paréntesis..	window.moveTo(23,43)
resizeBy(x,y)	Restablece el tamaño de la ventana en pixels.	window.resizeBy(200,200)
resizeTo(x,y)	Restablece el tamaño de la ventana en pixels.	window.resizeTo(200,200)
scrollBy(x,y)	Desplaza la vista de la ventana a la posición especificada.	window.scrollBy(345,300)
scrollTo(x,y)	Desplaza la vista de la ventana a la posición especificada.	window.scrollTo(323,123)
setInterval()	Ejecuta la función o expresión introducida como parámetro entre los paréntesis y en el tiempo establecido en milisegundos introducido en el segundo parámetro. Esta función se ejecutará cada intervalo de tiempo.	setInterval(funcion(),1000)
setTimeout()	Ejecuta la función o expresión introducida como parámetro entre los paréntesis y en el tiempo establecido en milisegundos en el segundo parámetro. Esta función se ejecutará suena vez pasado el tiempo especificado.	setTimeOut(funcion(),3000)
clearInterval()	Para el intervalo creado. Se especifica el id del intervalo entre los paréntesis.	clearInterval(2)
clearTimeout()	Para la función de timeout. Se especifica el id del intervalo entre los paréntesis.	ClearTimeout(2)
captureEvents()	Captura los eventos que se especifiquen entre los paréntesis.	window.captureEvents(evento)
releaseEvents()	Deja de capturar los eventos especificados entre los paréntesis.	window.releaseEvents(evento)
routeEvents()	Establece el evento, especificado entre los paréntesis, en una posición en la jerarquía de eventos.	window.routeEvents(evento)

Ejemplos de uso

Abrir ventana según confirmación

Primero pediremos al usuario confirmación, después crearemos una ventana nueva con los parámetros deseados. En caso de hacer click en cancelar devuelve el tamaño en pixeles de la ventana actual.

```
<script type="text/javascript">
    if (confirm("Desa abrir otra ventana?")){
        window.open('archivo.html',
            'Nueva ventana','height=300,width=300');
    }else{
        alert("Altura: " + window.innerHeight +
            " Anchura: " + window.innerWidth);
    }
</script>
```

Objeto Navigator

Propiedades de navigator

Propiedad	Descripción	Uso
appName	Devuelve el nombre del navegador usado.	navigator.appCodeName
appName	Devuelve el nombre del núcleo del navegador usado.	navigator.appName
appVersion	Devuelve la versión del navegador usado.	navigator.appVersion
cookieEnabled	Devuelve true si están habilitadas las cookies en el navegador.	navigator.cookieEnabled
onLine	Devuelve true si el navegador está online.	navigator.onLine
platform	Devuelve el sistema operativo donde está instalado el navegador.	navigator.platform
userAgent	Devuelve toda la información del navegador del cliente.	navigator.userAgent
language	Devuelve el idioma usado en el cliente.	navigator.language

Funciones de Navigator

Función	Descripción	Uso
javaEnabled()	Devuelve true si java está habilitado en el navegador.	navigator.javaEnabled()

Ejemplos de uso

Verificar navegador

En caso de que el navegador sea Firefox nos mostrará toda la información del cliente, en caso de ser Internet explorer nos mostrará si java está habilitado.

```
<script type="text/javascript">
  var navegador = navigator.appCodeName;
  if ( navegador == "Mozilla" ){
    alert(navigator.userAgent);
  }
  if ( navegador == "Explorer" ){
    alert(navigator.javaEnabled());
  }
</script>
```

Objeto Screen

Propiedades de screen

Propiedad	Descripción	Uso
availHeight	Devuelve la altura de la pantalla. Excluyendo barras.	screen.availHeight
availWidth	Devuelve la anchura de la pantalla. Excluyendo barras.	screen.availWidth
colorDepth	Devuelve la profundidad de bits de la paleta de colores.	screen.colorDepth
pixelDepth	Devuelve la resolución de pantalla.	screen.pixelDepth
height	Devuelve la altura de la pantalla, incluyendo barras.	screen.height
width	Devuelve la anchura de la pantalla, incluyendo barras.	screen.width

Ejemplos de uso

Comprobar tamaño de pantalla

Comprobaremos cual es el ancho de la pantalla y si es menor al que se necesita para mostrar la página avisaremos al usuario.

```
<script type="text/javascript">
    if (screen.width < 1500){
        alert("Tu pantalla es demasiado pequeña para
            mostrar la página.");
    }
</script>
```

Objeto History

Propiedades de history

Propiedad	Descripción	Uso
length	Devuelve todas las páginas almacenadas en el historial de la página.	history.length
current	Devuelve la URL completa de la entrada actual en el historial.	history.current
next	Devuelve la URL completa de la siguiente entrada en el historial.	history.next
previous	Devuelve la URL completa de la anterior entrada en el historial.	history.previous

Funciones de history

Función	Descripción	Uso
back()	Vuelve a la anterior página del historial.	history.back()
forward()	Muestra la siguiente página del historial.	history.forward()
go()	Carga la página, especificando su URL dentro de los paréntesis o su index en el historial.	history.go(34)

Ejemplos de uso

Cargar la última página del historial

Cargaremos la última página abierta por el usuario en la ventana de nuestra página en caso de que el usuario acepte.

```
<script type="text/javascript">
    if(confirm("desea ir a la última página visitada?")){
        history.go(history.length);
    }
</script>
```

Objeto Location

Propiedades de location

Propiedad	Descripción	Uso
hash	Devuelve la porción de anclaje de un URL, incluyendo el signo numeral (#).	location.hash
host	Devuelve el nombre del host y el puerto	location.host
hostname	Devuelve el hostname de la URL.	location.hostname
href	Devuelve la ruta de la página actual.	location.href
pathname	Devuelve la ruta del archivo de la página.	location.pathname
port	Devuelve el puerto usado.	location.port
protocol	Devuelve el protocolo utilizado.	location.protocol
search	Devuelve la consulta de la URL.	location.search

Funciones de location

Función	Descripción	Uso
assign()	Carga la página especificada entre los paréntesis.	location.assign("http://pagina.es")
reload()	Recarga la página actual.	location.reload()
replace()	Reemplaza la actual página por la especificada entre los paréntesis.	location.replace("http://pagina.es")

Ejemplos de uso

Redireccionar

Redireccionaremos al usuario al entrar en nuestra página.

```
<script type="text/javascript">  
    location.replace("http://google.es");  
</script>
```

Objeto Document

Propiedades de document

Propiedad	Descripción	Uso
anchors	Devuelve todas las anclas (<a>) del documento.	document.anchors[]
applets	Devuelve todos applets del documento.	document.applets[]
body	Devuelve la etiqueta body del documento.	document.body
cookie	Devuelve las cookies asociadas al documento.	document.cookie
domain	Devuelve el dominio donde se encuentra el documento.	document.domain
forms	Devuelve todos los formularios del documento.	document.forms[]
images	Devuelve todas las imágenes del documento.	document.images[]
embeds	Devuelve todos los objetos embeds del documento.	document.embeds[]
links	Devuelve todos los links del documento.	document.links[]
referrer	Devuelve la URL donde se accede al documento.	document.referrer
title	Devuelve el título del documento.	document.title
URL	Devuelve la URL completa del documento.	document.URL
bgColor	Devuelve el color de fondo del documento.	document.bgColor
fgColor	Devuelve el color del texto del documento.	document.fgColor
linkColor	Devuelve el color de los links del documento.	document.linkColor
vlinkColor	Lo mismo que links pero con enlaces visitados.	document.vlinkColor
alinkColor	Lo mismo que links pero con enlaces activos.	document.alinkColor
activeElement	Devuelve el elemento del documento que tiene el foco.	document.activeElement
defaultCharSet	Devuelve el juego de caracteres usado.	document.defaultCharSet
lastModified	Devuelve la fecha de la última modificación del documento.	document.lastModified

fileCreateDate	Devuelve la fecha de creación del documento.	document.fileCreateDate
fileModifiedDate	Devuelve la fecha de la modificación.	document.fileModifiedDate
fileSize	Devuelve el tamaño de bytes que ocupa el documento, sin imágenes.	document.fileSize
fileUpdatedDate	Igual que fileModifiedDate.	document.fileUpdatedSize
plugins	Devuelve todos los plugins usados en el documento.	document.plugins[]
protocol	Devuelve el protocolo usado en el documento, por defecto HTTP	document.protocol
readyState	Devuelve el estado de carga del documento.	document.readyState
security	Devuelve el certificado usado en el documento.	document.security
styleSheets	Devuelve todas las etiquetas de style del documento	document.styleSheets[]

Funciones de document

Funcion	Descripción	Uso
open()	Abre un documento del tipo indicado.	document.open("tipo", "opcion")
close()	Cierra la entrada de datos al documento.	document.close()
write()	Escribe en el documento lo introducido dentro de los paréntesis.	document.write("texto")
writeln()	Igual que write pero escribe con un salto de línea.	document.writeln("texto")
getElementsByName()	Devuelve los elementos que contienen el valor del paréntesis en la etiqueta name	document.getElementsByName("nombre")
getElementById()	Devuelve el elemento que contiene el id especificado en el paréntesis.	document.getElementById("id")[]
getElementsByTagName()	Devuelve los elementos que tengan la etiqueta especificada entre los paréntesis.	document.getElementsByTagName("etiqueta")[]
createElement()	Crea una etiqueta especificada.	document.createElement('p')
createEvent()	Crea un evento especificando el tipo de evento.	document.createEvent("MouseEvent")

Ejemplos de uso

Escribir en el documento

Escribiremos en el documento la fecha de la última modificación del documento.

```
<script type="text/javascript">  
    document.write(document.lastModified);  
</script>
```

Cambiar color de fondo

Usaremos la propiedad bgColor para cambiar el color de fondo del documento.

```
<script type="text/javascript">  
    document.bgColor = "green";  
</script>
```

Acceder a elemento formulario

Accederemos al primer valor del formulario y cambiaremos el color de este. Para que cambie de color un elemento el script debe ejecutarse después de cargarse la página, existen otras formas de hacerlo, colocar la etiqueta script después del body funcionará ya que se cargarán todos los elementos del body antes de aplicar cambios con javascript en ellos.

```
<body>  
    <form action="#" method="POST" name="formulario" >  
        <input type="text" value="Color">  
    </form>  
</body>  
<script type="text/javascript">  
    var elemento=document.forms[0].elements[0];  
    elemento.style.color = "blue";  
</script>
```

Acceder a elemento por id

Añadiremos el atributo id a la etiqueta html y la llamaremos con getElementById().

```
<script type="text/javascript">  
    var elemento=document.getElementById('prueba');  
</script>  
<body>  
    <p id="prueba"> Elemento </p>  
</body>
```

Nodos del DOM

Lista de nodos

Aquí una lista de los nodos genéricos que existen en el DOM, no están todos.

Nodo	Descripción	Acceso
Document	Es único y es el elemento padre de todos los elementos, el documento mismo.	document
Element	Representa una etiqueta de dentro del body de la página. Accedemos desde el nombre del elemento (etiqueta name).	document.nombre
Attr	EL atributo del elemento (etiqueta) y su contenido.	document.nombre.value
Text	El texto que contiene el nodo o etiqueta.	Document.nombre.text
Comment	Comentarios del documento.	document.comment

Propiedades de los nodos

Propiedad	Descripción	Uso
attributes	Devuelve los atributos del nodo	nodo.attributes
childNodes	Devuelve la lista de nodos hijos del nodo	nodo.childNodes
firstChild	Devuelve el primer hijo del nodo	nodo.firstChild
lastChild	Devuelve el último hijo del nodo	nodo.lastChild
localName	Devuelve el nombre de la etiqueta	nodo.localName
nextSibling	Devuelve el nodo contiguo del mismo nivel de nodos al nodo especificado.	nodo.nextSibling
previousSibling	Devuelve el nodo anterior del mismo nivel de nodos al nodo especificado	nodo.previousSibling
nodeValue	Devuelve el valor del nodo	nodo.nodeValue
parentNode	Devuelve el nodo padre del nodo especificado	nodo.parentNode
nodeName	Devuelve la etiqueta del nodo	nodo.nodeName
textContent	Devuelve el texto contenido en la etiqueta y el contenido de la etiqueta de todos los nodos hijos	nodo.textContent
hasChildNodes	Devuelve true si el nodo tiene nodos hijos y false si no tiene hijos	nodo.ChidNodes
tagName	Devuelve el tipo de etiqueta que es el nodo.	nodo.tagName

Funciones de los nodos

Función	Descripción	Uso
<code>appendChild()</code>	Añade un nodo hijo al nodo especificado entre los paréntesis.	<code>nodo.appendChild(nodo2)</code>
<code>cloneNode()</code>	Devuelve el nodo elegido clonado, se tiene que poner true entre los paréntesis.	<code>nodo.cloneNode(true)</code>
<code>compareDocumentPosition()</code>	Compara las posiciones de dos nodos en el documento.	<code>nodo.compareDocumentPosition(nodo2)</code>
<code>hasAttributes()</code>	Devuelve true si el elemento contiene atributos.	<code>nodo.hasAttributes()</code>
<code>setAttribute()</code>	Añade un atributo al elemento, especificando en el primer parámetro el tipo de atributo y en el segundo el valor del atributo.	<code>Nodo.setAttribute("id", "prueba")</code>
<code>getAttribute()</code>	Devuelve el atributo especificado entre los paréntesis.	<code>nodo.getAttribute(atributo)</code>
<code>removeAttribute()</code>	Elimina la etiqueta especificada entre los paréntesis.	<code>nodo.removeAttribute("id")</code>
<code>getAttributeNode()</code>	Igual que <code>getAttribute()</code> .	<code>nodo.getAttributeNode(atributo)</code>
<code>getElementsByTagName()</code>	Devuelve los elementos que tengan la etiqueta especificada entre los paréntesis.	<code>Document.getElementsByTagName("etiqueta")[]</code>
<code>hasChildNodes()</code>	Devuelve true si el elemento tiene nodos hijos.	<code>nodo.hasChildNodes()</code>
<code>insertBefore()</code>	Inserta el elemento especificado en el primer parámetro delante del elemento especificado en el segundo parámetro.	<code>nodo.insertBefore(nodouno, nodoexistente)</code>
<code>isEqualNode()</code>	Compara si dos nodos tienen los mismos nodos hijos, son del mismo tipo, si tienen los mismos atributos y los mismos valores de los atributos.	<code>nodo.isEqualNode(nodo2)</code>
<code>removeChild()</code>	Elimina el nodo hijo especificado entre los paréntesis.	<code>nodo.removeChild(nodohijo)</code>
<code>replaceChild()</code>	Reemplaza el nodo hijo especificado como segundo parámetro por el nodo nuevo especificado en el primer parámetro.	<code>Nodo.replaceChild(nodouno, nodohijo)</code>

Ejemplos de uso

Añadir elemento

Para añadir un elemento tenemos que crear el elemento, crear el texto que contendrá el elemento, añadir el texto al elemento, acceder al padre de este, en este caso body y después se lo añadiremos al padre.

```
<body>
</body>
<script type="text/javascript">
  var elemento = document.createElement('p');
  var texto = document.createTextNode('Elemento de prueba');
  elemento.appendChild(texto);
  document.body.appendChild(elemento);
</script>
```

Eliminar elemento

Para eliminar un elemento tenemos que acceder al padre de este, después podremos eliminarlo.

```
<body>
  <p>Prueba</p>
</body>
<script type="text/javascript">
  var hijo = document.getElementsByTagName('p')[0];
  document.body.removeChild(hijo);
</script>
```

Eliminar atributos de elemento

Eliminaremos todos los atributos del elemento con 2 funciones.

```
<body>
  <p id="prueba" name="prueba" style="color:blue"> Prueba </p>
</body>
<script type="text/javascript">
  function mostrar(elemento){
    for (i=0 ; i < elemento.attributes.length ; i++){
      document.write( elemento.attributes[i].name
        + "<br />");
    }
  }
  function borrar(elemento){
    for (i=0 ; i < elemento.attributes.length ; i++){
      elemento.removeAttribute
      (elemento.attributes[i].name);
    }
    document.write("Atributos borrados.");
  }
  var hijo = document.getElementsByTagName('p')[0];
  mostrar(hijo);
  borrar(hijo);
</script>
```

Atributos de los Nodos

Propiedades de los Atributos

Propiedad	Descripción	Uso
isId	Devuelve true si el atributo es del tipo Id.	nodo.atributo.isId
name	Devuelve el nombre del atributo	nodo.atributo.name
ownerElement	Devuelve el elemento propietario de ese atributo.	nodo.atributo.ownerElement
specified	Devuelve true si el atributo a sido definido.	nodo.atributo.specified
value	Devuelve el valor del atributo.	nodo.atributo.value

Ejemplos de uso

Cambiar el tipo de input

Podemos cambiar el tipo de input de un formulario accediendo al value del atributo type.

```
<body>
  <input id="cambio" type="button" value="Esto era un botón" />
</body>
<script type="text/javascript">
  var atributo =
    document.getElementById('cambio').attributes[0];
  atributo.value = "text";
</script>
```

Otra forma de hacerlo es con la función setAttribute.

```
<body>
  <input id="cambio" type="button" value="Esto era un botón" />
</body>
<script type="text/javascript">
  document.getElementById('cambio').setAttribute("type", "text");
</script>
```

Eventos

Propiedades de los eventos

Propiedad	Descripción	Uso
bubbles	Devuelve true si el evento es un evento bubleable.	nodo.event.bubbles
cancelable	Devuelve true si el evento es cancelable.	nodo.event.cancelable
currentTarget	Devuelve el elemento al que afecta el evento.	nodo.event.currentTarget
target	Devuelve el elemento al que le afecta el evento.	nodo.event.target
timeStamp	Devuelve el tiempo transcurrido en milisegundos desde las 12 de la noche del 1 de Junio de 1970 hasta que se ejecuta el evento.	nodo.event.timeStamp
type	Devuelve el tipo de evento.	nodo.event.type
charCode	Devuelve el código ASCII de las teclas pulsadas.	nodo.event.charCode
CAPTURING_PHASE	Constante que devuelve la fase del evento capturado actualmente.	event.CAPTURING_PHASE
AT_TARGET	El evento actual se encuentra en la fase de destino, es decir, que está siendo evaluado en el destino del evento.	event.AT_TARGET
BUBBLING_PHASE	Captura la posición de la burbuja actual.	event.BUBBLING_PHASE

Funciones de los eventos

Función	Descripción	Uso
initEvent(type, bubbles, cancelable)	Inicializa propiedades de evento pasandole en los paréntesis el tipo en el primer parámetro, si es emergente en el segundo parámetro y si es cancelable en el tercer parámetro del paréntesis. Previamente el evento tiene que ser creado con document.createEvent().	nodo.event.initEvent("onclick", "true", "true")
preventDefault()	Para el evento ejecutado.	nodo.event.preventDefault()
stopPropagation()	Para el evento ejecutado.	nodo.event.stopPropagation()
addEventListener(event, function, bubble)	Añade un evento al elemento. Como primer parámetro se pasa el tipo de evento, como segundo parámetro la función a ejecutar y el tercero define el orden de flujo de eventos (bubbles).	nodo.addEventListener("evento", "funcion()", true)

attachEvent(event,function)	Lo mismo que addEventListener() pero para Internet Explorer.	nodo.attachEvent("mouseclick","funcion()")
removeEventListener(event,function,boolean)	Se usa del mismo modo que addEventListener() pero sirve para eliminar el evento.	nodo.addEventListener("evento","funcion()",true)
detachEvent(event,function)	Igual que attachEvent() pero elimina el evento en Internet Explorer	nodo.detachEvent("mouseclick","funcion()")
dispatchEvent(evento)	Fuerza la ejecución del evento.	nodo.dispatchEvent("mouseover")
fire()	Ejecuta el evento especificado entre los paréntesis del elemento.	nodo.fire("mouseon")

Ejemplos de uso

Filtrar introducción de teclas en input

Con la propiedad charCode podemos definir un filtro para parar las teclas que no queremos que se escriban y dejar pasar las teclas que si queremos. En este caso solo dejará pasar dígitos.

```

<script type="text/javascript">
    function comprobar(evento,etiqueta){
        if ( !( evento.charCode <= 57 && evento.charCode >=
48          ) ){
            return false;
        }
        return true;
    }
</script>
<body>
    <input type="text" name="intro"
onkeypress="return comprobar(event,this);" value=" "/>
</body>

```


Eventos de ratón

Evento	Descripción	Uso
onclick	Se activa cuando se clicka sobre el elemento.	nodo.onclick
ondblclick	Se activa cuando se hace doble click sobre el elemento.	nodo.ondblclick
onmousedown	Se activa cuando se presiona el elemento, sin soltar el botón del ratón.	nodo.onmousedown
onmouseup	Se activa cuando se suelta el botón del ratón al pulsar sobre un elemento.	nodo.onmouseup
onmousemove	Se activa cuando mueve el puntero sobre el.	nodo.onmousemove
onmouseover	Se activa cuando el puntero pasa por encima del elemento.	nodo.onmouseover
onmouseout	Se activa cuando el puntero sale del elemento.	nodo.onmouseout
Atributos de evento de ratón		
button(0 1 2)	Devuelve o aplica que botón del ratón a sido presionado. Puede ser el derecho el del centro o el izquierdo.	evento.button(2)
clientX	Devuelve la coordenada x del puntero en la página.	evento.clientX
clientY	Devuelve la coordenada y del puntero en la página.	evento.clientY
screenX	Devuelve la coordenada x del puntero en la pantalla.	evento.screenX
screenY	Devuelve la coordenada x del puntero en la pantalla.	evento.screenY
relatedTarget	Devuelve el elemento que desencadeno el evento.	evento.relatedTarget
Función de evento de ratón		
initMouseEvent()	Inicializa los valores del evento de ratón.	event.initMouseEvent(tipo_evento)

Eventos de teclado

Evento	Descripción	Uso
onkeypress	Se activa cuando se pulsa una tecla.	objeto.onkeypress = function(){sentencias}
onkeydown	Se activa cuando se pulsa la tecla sin soltarla.	objeto.onkeydown = function(){sentencias}
onkeyup	Se activa cuando se suelta la tecla después de pulsarla.	objeto.onkeyup = function(){sentencias}
Atributos de evento de teclado		
altKey	Devuelve true si la tecla alt a sido presionada	evento.altKey
shiftKey	Devuelve true si la tecla shift a sido presionada	evento.shiftKey
ctrlKey	Devuelve true si la tecla control a sido presionada	evento.ctrlKey
metaKey	Devuelve true se la tecla meta a sido presionada.	evento.metaKey
relatedTarget	Devuelve el elemento que desencadeno el evento.	evento.relatedTarget
Funciones de evento de teclado		
initKeyboardEvent()	Inicializa los valores del evento de teclado.	event.initKeyboardEvent(tipo _evento)

Ejemplos de uso

Dos formas de añadir un evento a un elemento

La forma no semántica se realiza colcando el evento directamente en la etiqueta html.

```
<script type="text/javascript">
    function pulsar(){
        alert("pulsaste");
    }
</script>
<body>
    <input id="cambio" type="button" value="Ejecutar"
    onclick="pulsar()" />
</body>
```

La primera forma semántica es añadir la función al evento del elemento mediante el acceso con dom. Esta parte del script tendremos que añadirla una vez cargados los elementos del documento.

```
<script type="text/javascript">
    document.getElementById("boton").onclick = function (){
        alert("pulsaste");
    }
</script>
```

También podemos añadir la función por su nombre, pero sin añadirle paréntesis.

```
<script type="text/javascript">
    function pulsar(){
        alert("pulsaste");
    }
    document.getElementById("boton").onclick = pulsar;
</script>
```

La tercera forma es utilizar la función `addEventListener` (`attachEvent` en internet explorer) para añadir el evento al elemento. Esta parte también necesita ser ejecutada una vez cargados todos los elementos del documento web.

```
<script type="text/javascript">
    function pulsar(){
        alert("pulsaste");
    }
    document.getElementById("boton").
addEventListener("click",pulsar,true);
</script>
```



En internet explorer.

```
<script type="text/javascript">
    function pulsar(){
        alert("pulsaste");
    }
    document.getElementById("boton").
attachEvent("click",pulsar);
</script>
```



Eventos de formulario

Evento	Descripción	Uso
onblur	Se activa cuando se quita el foco del elemento.	nodo.onblur
onfocus	Se activa cuando se le da foco al elemento.	nodo.onfocus
onchange	Se activa cuando un elemento de formulario como "select" o con la propiedad "checked" cambia de valor o estado.	nodo.onchange
onreset	Se activa cuando se resetea el formulario.	nodo.onreset
onselect	Se activa cuando se selecciona texto de un elemento del tipo "text".	nodo.onselect
onsubmit	Se activa cuando se envía el formulario.	nodo.onsubmit

Eventos de frames y ventanas

Evento	Descripción	Uso
onabort	Se activa cuando una imagen no se carga del todo en la página.	objeto.onabort
onerror	Se activa cuando no se carga una imagen.	objeto.onerror
onload	Se activa cuando se carga el documento, frame, iframe,...	objeto.onload
onresize	Se activa cuando se restablece el tamaño del documento, frame, iframe,...	objeto.onresize
onscroll	Se activa cuando se desplaza con el scroll por el documento, frame, iframe,...	objeto.onscroll
onunload	Se activa cuando se cierra el documento, frame, iframe,...	objeto.onunload

Atributos de etiquetas y objetos

Etiquetas de enlace

Atributos comunes: Anchor, Area, Base

Atributo	Descripción	Uso
href	Devuelve o establece el atributo href.	nodo.href
target	Devuelve o establece el atributo target.	nodo.target

Atributos comunes: Anchor, Link

Atributo	Descripción	Uso
charset	Devuelve o establece el código de caracteres del documento referenciado en el link.	nodo.charset
hreflang	Devuelve o establece el language del documento referenciado en el link.	nodo.hreflang
media	Devuelve o establece el tipo de pantalla del documento referenciado en el link.	nodo.media
rel	Devuelve o establece el atributo rel del link.	nodo.rel
rev	Devuelve o establece el atributo rev del link.	nodo.rev
type	Devuelve o establece el atributo type del link.	nodo.type
href	Devuelve o establece el atributo href.	nodo.href

Atributos de Anchor

Atributo	Descripción	Uso
name	Devuelve o establece del atributo name del link.	nodo.name

Atributos de Area

Atributo	Descripción	Uso
alt	Devuelve o establece el atributo alt del area.	nodo.alt
coords	Devuelve o establece el atributo coords del area.	nodo.coords
hash	Devuelve o establece la parte del anchor del atributo href del area.	nodo.hash

Atributos de Link

hostname	Devuelve o establece la parte del hostname:port del atributo href del area.	nodo.hostname
pathname	Devuelve o establece la parte del pathname del atributo href del area.	nodo.pathname
port	Devuelve o establece la parte del puerto del atributo href del area.	nodo.port
protocol	Devuelve o establece la parte del protocolo del atributo href del area.	nodo.protocolo
nohref	Devuelve o establece el atributo nohref del area.	nodo.nohref
search	Devuelve establece el la parte consultada del string del href.	nodo.search
shape	Devuelve establece el atributo shape.	nodo.shape

Etiquetas de head y body

Atributo	Descripción	Uso
media	Devuelve o establece el tipo de pantalla del documento referenciado en el link.	nodo.media

Atributos de Body

Atributo	Descripción	Uso
alink	Devuelve o establece el atributo alink.	body.alink
link	Devuelve o establece el atributo link.	body.link
vLink	Devuelve o establece el atributo vLink.	body.vLink
text	Devuelve o establece el atributo text.	body.text
background	Devuelve o establece el atributo background.	body.background
bgcolor	Devuelve o establece el atributo bgcolor.	body.bgcolor

Atributos de Meta

Atributo	Descripción	Uso
content	Devuelve o establece el atributo content	meta.content
httpEquiv	Devuelve o establece el atributo httpEquiv.	meta.httpEquiv
name	Devuelve o establece del atributo name.	meta.name

scheme	Devuelve o establece del atributo name.	meta.scheme
--------	---	-------------

Etiquetas de formulario

Atributos de Form

Atributo	Descripción	Uso
action	Devuelve o establece el atributo action del form.	nodo.action
method	Devuelve o establece el atributo method.	nodo.method
name	Devuelve o establece el atributo name.	nodo.name
target	Devuelve o establece el atributo target.	nodo.target
length	Devuelve el número de elementos del form.	nodo.length
acceptCharset	Devuelve o establece el atributo acceptCharset.	nodo.acceptCharset
enctype	Devuelve o establece el atributo enctype.	nodo.enctype

Funciones de Form

Función	Descripción	Uso
reset()	Resetea el formulario vaciando el contenido de sus inputs.	form.reset()
submit()	Envía el contenido del formulario.	form.submit()

Eventos de Form

Función	Descripción	Uso
onreset	Se activa al resetear el formulario	form.onreset
onsubmit	Se activa al enviar el formulario.	form.onsubmit

Atributos comunes: input
 button, button, checkbox,
 hidden, text, file, password, submit, reset,
 radio, select, option, textarea

Atributos comunes:Checkbox, Radio

Atributo	Descripción	Uso
disabled	Devuelve o establece el valor booleano del atributo disabled.	nodo.disabled
form	Devuelve el nombre del formulario al que pertenece.	nodo.form
type	Devuelve o establece el tipo de botón.	nodo.type
value	Devuelve o establece el valor del botón.	nodo.value
name	Devuelve o establece el nombre del botón.	nodo.name

Atributos comunes:Password, Text, textarea

Atributo	Descripción	Uso
checked	Devuelve o establece el valor booleano del atributo checked, true en caso de estar marcado.	nodo.checked
defaultChecked	Devuelve o establece el valor booleano del atributo defaultChecked, true si por defecto está marcado.	nodo.defaultChecked

Atributos de File

Atributo	Descripción	Uso
accept	Devuelve o establece la lista separada por comas de los contenidos permitidos a enviar.	nodo.accept

Atributos de Textarea

Atributo	Descripción	Uso
defaultValue	Devuelve o establece el valor del nodo.	nodo.defaultValue
maxLength	Devuelve o establece el valor del atributo maxLength.	nodo.maxLength
readOnly	Devuelve o establece el valor booleano del atributo readOnly.	nodo.readOnly
size	Devuelve o establece el valor del tamaño del nodo.	nodo.size

Atributos de Select

Atributo	Descripción	Uso
index	Devuelve o establece el index de la opción.	nodo.index
selected	Devuelve o establece el valor booleano de true en caso de estar seleccionado.	nodo.selected
text	Devuelve o establece el texto de la opción.	nodo.text

Atributos de Option

Atributo	Descripción	Uso
options	Devuelve un array con las diferentes opciones.	nodo.options[]
length	Devuelve la cantidad de opciones que tiene el select.	nodo.length
multiple	Devuelve o establece si está habilitada la multiselección.	nodo.multiple
selectedIndex	Devuelve el index de la opción seleccionada.	nodo.selectedIndex

Funciones de Select

Función	Descripción	Uso
add()	Añade una nueva opción al select. En el primer parámetro el parentesis va la nueva opción y en el segundo parámetro la opción contigua a la que se situará.	nodo.add("opcion", "opcion de antes");
remove()	Elimina una opción pasándole un index.	nodo.remove(1);

Etiquetas de tablas

Atributo	Descripción	Uso
cols	Devuelve o establece la anchura del textarea.	nodo.cols
rows	Devuelve o establece la altura del textarea.	nodo.rows

Atributos de Table

Atributo	Descripción	Uso
cells	Devuelve o establece un vector con todas las	nodo.cells

	columnas (<td> <th>) de la tabla.	
rows	Devuelve o establece un vector con todas las filas (<tr> <tr>) de la tabla.	nodo.rows
tBodies	Devuelve o establece un vector con todas las etiquetas tbody de la tabla.	nodo.tBodies
caption	Devuelve o establece la etiqueta caption de la tabla.	nodo.caption
cellPadding	Devuelve o establece el atributo cellPadding.	nodo.cellPadding
cellSpacing	Devuelve o establece el atributo cellSpacing.	nodo.cellSpacing
frame	Devuelve o establece el atributo frame.	nodo.frame
rules	Devuelve o establece el atributo rules.	nodo.rules
summary	Devuelve o establece el atributo summary.	nodo.summary
tFoot	Devuelve el contenido de las etiquetas <tfoot>.	nodo.Tfoot
tHead	Devuelve el contenido de las etiquetas <thead>.	nodo.tHead

Atributos comunes de tr,td/th

Atributo	Descripción	Uso
vAlign	Devuelve o establece el alineamiento vertical del nodo.	nodo.vAlign
ch	Devuelve o establece el alineamiento de los caracteres del nodo.	nodo.ch
chOff	Devuelve o establece el numero de caracteres al que afectará el ch.	nodo.chOff

Atributos de Td/Th

Atributo	Descripción	Uso
cells	Devuelve o establece un vector con todas las columnas (<td> <th>) de la fila.	tr.cells
chOff	Devuelve o establece el numero de caracteres al que afectará el ch.	tr.chOff
rowIndex	Devuelve el index de la fila dentro de table.	tr.rowIndex
sectionRowIndex	Devuelve el index de la fila dentro de las etiquetas tfoot, thead,tbody.	tr.sectionRowIndex

Atributos de Tr

Atributo	Descripción	Uso
abbr	Devuelve o establece el atributo abbr.	td.abbr
axis	Devuelve o establece el atributo axis.	th.axis
cellIndex	Devuelve el index de la columna dentro de table.	td.cellIndex
colSpan	Devuelve o establece el atributo colSpan.	th.colSpan
rowSpan	Devuelve o establece el atributo rowSpan.	td.rowSpan
headers	Devuelve la lista de headers de las columnas.	th.headers

Objeto Style

Etiquetas de frame/iframe

Atributos de frame/iframe

Atributo	Descripción	Uso
contentDocument	Devuelve el objeto documento que tiene el frame/iframe.	frame.contentDocument
contentWindow	Devuelve el objeto ventana que tiene el frame/iframe.	frame.contentWindow
frameBorder	Devuelve o establece el borde del frame.	iframe.frameBorder
height	Devuelve o establece la altura del frame.	frame.height
width	Devuelve o establece la anchura del frame.	frame.width
longDesc	Devuelve o establece el atributo longdesc.	frame.longdesc
marginHeight	Devuelve o establece el margen a lo alto del frame.	frame.marginHeight
marginWidth	Devuelve o establece el margen a lo ancho del frame.	frame.marginWidth
name	Devuelve o establece el atributo name.	iframe.name

noResize	Devuelve o establece un valor booleano para no permitir la redimensión.	frame.noResize
scrolling	Devuelve o establece un valor booleano para permitir el scroll.	frame.scrolling
src	Devuelve o establece el atributo src.	iframe.src

Atributos de background

Atributo	Descripción	Uso
background	Devuelve o establece las propiedades de background.	nodo.style.background
backgroundAttachment	Devuelve o establece la propiedad de la imagen de fondo (fixed,scroll,inherit)	nodo.style.backgroundAttachment
backgroundColor	Devuelve o establece el color de fondo.	nodo.style.backgroundColor
backgroundImage	Devuelve o establece la url de la imagen de fondo.	nodo.style.backgroundImage
backgroundPosition	Devuelve o establece la posición de la imagen de fondo.	nodo.style.backgroundPosition
backgroundRepeat	Devuelve o establece la posición de la imagen de fondo.	nodo.style.backgroundRepeat

Atributos de border

Atributo	Descripción	Uso
border	Devuelve o establece las propiedades de border.	nodo.style.border
borderColor	Devuelve o establece el color de las 4 esquinas.	nodo.borderColor
borderBottom	Devuelve o establece las propiedades de borderBottom.	nodo.style.borderBottom
borderBottomColor	Devuelve o establece el color del borde.	nodo.style.borderBottomColor
borderBottomStyle	Devuelve o establece el estilo del borde.	nodo.style.borderBottomStyle
borderBottomWidth	Devuelve o establece el ancho del borde.	nodo.style.borderBottomWidth
borderTop	Devuelve o establece las propiedades del borde.	nodo.style.borderTop
borderTopColor	Devuelve o establece el color del borde.	nodo.style.borderTopColor

		lor
borderTopStyle	Devuelve o establece el estilo del borde.	nodo.style.borderTopStyle
borderTopWidth	Devuelve o establece el ancho del borde.	nodo.style.borderTopWidth
borderRight	Devuelve o establece las propiedades del borde	nodo.style.borderRight
borderRightColor	Devuelve o establece el color del borde.	nodo.style.borderRightColor
borderRightStyle	Devuelve o establece el estilo del borde.	nodo.style.borderRightStyle
borderRightWidth	Devuelve o establece el ancho del borde.	nodo.style.borderRightWidth
borderLeft	Devuelve o establece las propiedades del borde	nodo.style.borderLeft
borderLeftColor	Devuelve o establece el color del borde.	nodo.style.borderLeftColor
borderLeftStyle	Devuelve o establece el estilo del borde.	nodo.style.borderLeftStyle
borderLeftWidth	Devuelve o establece el ancho del borde.	nodo.style.borderLeftWidth
outline	Devuelve o establece la propiedad outline.	nodo.style.outline
outlineColor	Devuelve o establece el color del outline.	nodo.style.outlineColor
outlineStyle	Devuelve o establece el estilo del outline.	nodo.style.borderLeftStyle
outlineWidth	Devuelve o establece el ancho del outline.	nodo.style.outlineWidth

Atributos de list

Atributo	Descripción	Uso
listStyle	Devuelve o establece todos los estilos de la lista	nodo.style.listStyle
listStyleImage	Devuelve o establece la imagen de la lista.	nodo.style.listStyleImage
listStylePosition	Devuelve o establece la posición del punto de la lista (outside inside inherit).	nodo.style.listStylePosition
listStyleType	Devuelve o establece el tipo de marcador de la lista.	nodo.style.listStyleType

Atributos de margin/padding

Atributo	Descripción	Uso
margin / padding	Devuelve o establece todos los márgenes.	nodo.style.margin
marginTop	Devuelve o establece el margen de arriba.	nodo.style.marginTop
marginBottom	Devuelve o establece el margen de abajo.	nodo.style.marginBottom
marginRight	Devuelve o establece el margen de la derecha.	nodo.style.marginRight
marginLeft	Devuelve o establece el margen de la izquierda.	nodo.style.marginLeft

Atributos de position

Atributo	Descripción	Uso
bottom	Devuelve o establece la posición desde abajo del nodo.	nodo.style.bottom
top	Devuelve o establece la posición desde arriba del nodo.	nodo.style.top
right	Devuelve o establece la posición desde la derecha del nodo.	nodo.style.right
left	Devuelve o establece la posición desde la izquierda del nodo.	nodo.style.left
clear	Devuelve o establece el atributo clear.	nodo.style.clear
cssFloat	Devuelve o establece el atributo float.	nodo.style.cssFloat
position	Devuelve o establece el tipo de posición (static,relative,absolute o fixed).	nodo.style.position
height	Devuelve o establece la altura del nodo.	nodo.style.height
width	Devuelve o establece la anchura del nodo.	nodo.style.width
clip	Devuelve o establece el tipo de clip (auto,rect(bottom,top,right,left),inherit).	nodo.style.clip
cursor	Devuelve o establece el tipo de cursor.	nodo.style.cursor
display	Devuelve o establece como se mostrará el nodo.	nodo.style.display
visibility	Devuelve o establece como se visualizará el nodo.(visible,hidden,colapse,inherit)	nodo.style.visibility
maxHeight	Devuelve o establece el máximo de altura.	nodo.style.maxHeight
maxWidth	Devuelve o establece el máximo de anchura.	nodo.style.maxWidth
minHeight	Devuelve o establece el mínimo de altura.	nodo.style.minHeight
minWidth	Devuelve o establece el mínimo de anchura.	nodo.style.minWidth

overflow	Devuelve o establece el tipo de metodo a usar en caso de desbordamiento de datos. (visible,hidden,scroll,auto,inherit)	nodo.style.overflow
verticalAlign	Devuelve o establece el alineamiento vertical.	nodo.style.verticalAlign
zIndex	Devuelve o establece la propiedad zIndex.	nodo.style.zIndex

Atributos de table

Atributo	Descripción	Uso
borderCollapse	Devuelve o establece el tipo de separacion del borde con los bordes de los elementos interiores (separate,collapse,inherit).	nodo.style.borderCollapse
borderSpacing	Devuelve o establece el tipo de separación entre columnas.	nodo.style.borderSpacing
captionSide	Devuelve o establece donde se pondrá el título (caption) de la tabla. (top,bottom,inherit)	nodo.style.captionSide
emptyCells	Devuelve o establece el tipo de visibilidad de las columnas vacías (show,hide,inherit)	nodo.style.emptyCells
tableLayout	Devuelve o establece el tipo de espacio entre las columnas. (automatic,fixed,inherit)	nodo.style.tableLayout

Atributos de text

Atributo	Descripción	Uso
textAlign	Devuelve o establece el tipo de alineamiento del texto (left,right,center,justify,inherit)	nodo.style.textAlign
textDecoration	Devuelve o establece el tipo de decoración del texto.(none,underline,overline,linethrough,blink,inherit)	nodo.style.textDecoration
textIndent	Devuelve o establece el indentado del texto.	nodo.style.textIndent
textShadow	Devuelve o establece la sombra del texto. (none,h-shadow,v-shadow,blur,color,inherit)	nodo.style.textShadow
textTransform	Devuelve o establece el tipo de texto. (none,capitalize,uppercase,lowercase,inherit)	nodo.style.textTransform
font	Devuelve o establece las propiedades de la fuente	nodo.style.font
fontFamily	Devuelve o establece la familia tipográfica.	nodo.style.fontFamily
fontSize	Devuelve o establece el tamaño de fuente.	nodo.style.fontSize
fontSizeAdjust	Devuelve o establece el ajuste de fuente.	nodo.style.fontSizeAdjust

fontStyle	Devuelve o establece el estilo de la fuente. (normal,italic,oblique,inherit)	nodo.style.fontStyle
fontVariant	Devuelve o establece variantes de la fuente. (normal,small-caps,inherit)	nodo.style.fontVariant
fontWeight	Devuelve o establece la anchura de la fuente. (normal,lighter,bold,bolder,,inherit)	nodo.style.fontWeight
color	Devuelve o establece el color de la fuente.	nodo.style.fontWeight
direction	Devuelve o establece la dirección de escritura de la fuente.(ltr,rtl,inherit)	nodo.style.direction
unicodeBidi	Devuelve o establece el tipo de escritura soportado, se usa junto direction. (normal,embed,bidi-override,inherit)	nodo.style.unicodeBidi
whiteSpace	Devuelve o establece como se tratarán los espacios en el texto. (normal,nowrap,pre,inherit)	nodo.style.whiteSpace
wordSpacing	Devuelve o establece el espacio entre las palabras en pixeles.	nodo.style.wordSpacing

Etiqueta img

Atributos de img

Atributo	Descripción	Uso
Eventos de img		
align	Devuelve o establece el alineamiento de la imagen.	img.align
alt	Devuelve o establece el atributo alt de la imagen.	img.alt
border	Devuelve o establece el atributo border de la imagen.	img.border
complete	Devuelve un valor booleano, true si la imagen se transfirió completa.	img.complete
height	Devuelve o establece la altura de la imagen.	img.height
width	Devuelve o establece la anchura de la	img.width

	imagen.	
hspace	Devuelve o establece el espacio horizontal de la imagen.	img.hspace
vspace	Devuelve o establece el espacio vertical de la imagen.	img.vspace
longDesc	Devuelve o establece la url donde está la descripción larga de la imagen.	img.longDesc
lowsrc	Devuelve o establece la url donde está la imagen para resoluciones bajas.	img.lowsrc
name	Devuelve o establece el atributo name.	img.name
src	Devuelve o establece el atributo src.	img.src
useMap	Devuelve o establece el atributo useMap.	img.useMap

Evento	Descripción	Uso
onabort	Se activa en caso de que la imagen no se aborte.	img.onabort
onerror	Se activa en caso de que la imagen tenga un error.	img.onerror
onload	Se activa en caso de que la imagen se cargue por completo.	img.onload

Códigos útiles

Tratamiento de imágenes

Menu de imágenes pequeñas que se agrandan

Añadiendo los enlaces de las imágenes nos mostrará un menú con las imágenes pequeñas y una grande, al pulsar sobre una imagen pequeña nos la mostrará grande al lado. Se tienen que añadir las etiquetas html y head para el buen funcionamiento.

```
<style>
    body {background-color:#000000;color:#E27907;font-
    family:Verdana,Arial;font-size:10pt;letter-spacing:2;
    }
    .normal{border:4px solid #000000;}
    .seleccionada {border:4px solid #ff0000;}
</style>
<script language=javascript>
var lastID = 0;
function seleccionar(id) {
    if (lastID > 0) {
        document.getElementById(lastID).className =
        "normal";
    }
    document.getElementById(id).className =
    "seleccionada";
    document.getElementById(0).src =
    document.getElementById(id).src;
    lastID = id;
}
function cargarprimera() { seleccionar(1); }
window.onload = cargarprimera;
</script>
<body>
    <table border=0>
    <tr>
    <td valign=top>
    
    <br />
    <br />
    </td>
    <td width=15> </td>
    <td valign=top>
    <img id=0 src="">
    </td>
    </tr>
    </table>
</body>
```

Fadein a imagen (la imagen aparece con fundido)

Lo que se realiza con javascript es modificar la opacidad de la imagen para que este a 0 y poco a poco suba a 100 para verse completamente.

```
<html>
  <head>
    <script language="Javascript" type="text/javascript">
      var op = 0;
      function opa_menu(){
        op += 1;
        var obj = document.getElementById('imagen');
        if (op < 110){
          obj.style.filter = 'alpha(opacity='+op+')';
          obj.style.MozOpacity= op /100;
          obj.style.opacity = op /100;
        }
      }

      function inicio(){

        document.getElementById('imagen').style.opacity =
        ".01";

        document.getElementById('imagen').style.filter =
        "alpha(opacity=00)";

        document.getElementById('imagen').style.MozOpacity
        = ".01";

          if (op < 110) {
            setInterval(opa_menu,5);
          }

      }

      window.onload = inicio;

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

Shakeimage (Agita la imagen cuando pasa el ratón)

Al pasar el ratón por encima de la imagen esta se agita en medida del parámetro “movimiento”.

```
<html>
  <head>
  <style>
  .shakeimage{ position:relative; }
  </style>
  <script language="JavaScript1.2">
  var movimiento=3;
  var pararlo=0;
  var a=1;
  function init(elemento){
    pararlo=0;
    shake=elemento;
    shake.style.left=0;
    shake.style.top=0;
  }
  function mover_imagen(){
    if ((!document.all&&!document.getElementById)||pararlo==1)
      return;
    if (a==1){
      shake.style.top=parseInt(shake.style.top)+movimiento;
    }
    else if (a==2){
      shake.style.left=parseInt(shake.style.left)+movimiento;
    }
    else if (a==3){
      shake.style.top=parseInt(shake.style.top)-movimiento;
    }
    else{
      shake.style.left=parseInt(shake.style.left)-movimiento;
    }
    if (a<4){a++; }else{a=1;}
    setTimeout("mover_imagen()",50);
  }
  function parar(elemento){
    pararlo=1;
    elemento.style.left=0;
    elemento.style.top=0;
  }
  </script>
  </head>
  <body>
  
  </body>
</html>
```

Slideshow (Cambia de imagen cada cierto tiempo)

Este script crea una presentación de fotos, cuando pasan 3 segundos la imagen cambia con n funcuido. A la vez la imagen será un enlace a una página web.

```
<html>
  <head>
  </head>
  <body>
    <div id="cambiador" style="text-align: center;" ></div>
    <script type="text/javascript">
      var cont = 0;
      var arr = [
        ["http://www.google.es/", "imagen1.jpg"],
        ["http://www.google.com/", "imagen2.jpg"],
      ]

      function cambiarimagen() {
        var elemento =
          document.getElementById("cambiador");
        cont = cont % arr.length;
        elemento.innerHTML = "<a
          href='"+arr[cont][0]+'"'
          target='_blank'><img border=0
          src='"+arr[cont][1]+'"></a>";
        cont++;
      }

      function inicio() {
        cambiaimg_vku();
        setInterval(cambiaimg_vku, 3000);
      }

      window.onload=inicio;
    </script>
  </body>
</html>
```

Webgrafia

Algunas de las páginas han servido con su ejemplo práctico y otras en servido para mejorar la comprensión de los diferentes conceptos de la teoría. De algunas se han copiado frases, códigos o ejemplo y otras simplemente son una buena referencia a tener en cuenta. Todas ellas están aquí.

http://www.codexemplar.org/curso/curso_4_3.php

<http://aprendeweb.16mb.com/javascript/>

<http://www.krispytech.com/blog/?p=304>

<http://www.phpied.com/3-ways-to-define-a-javascript-class/>

<http://helephant.com/2009/01/18/javascript-object-prototype/>

<http://www.regular-expressions.info>

<http://web.ontuts.com/snippets/10-expresiones-regulares-imprescindibles-en-desarrollo-web/>

<http://alexjimenez.wordpress.com/2012/08/01/tips-silverlight-validaciones-con-expresiones-regulares-ip-e-mail-url/>

<http://regexlib.com>

<http://www.comptechdoc.org/independent/web/cgi/javamanual/>

<http://www.w3schools.com/jsref/>

<https://developer.mozilla.org/en-US/learn/javascript>

Al igual que el mapa en el bolsillo, la cantimplora en el cinturón y el machete cruzado en la espalda es esencial para la vida de un guerrillero estos manuales facilitan la lucha constante contra los trabajos de programación.

Los informáticos tenemos la suerte de tener internet cerca de nosotros, pero cuando esa suerte no existe entonces tenemos que recurrir al papel o pdf. Estos pequeños manuales no atienden a explicaciones para principiantes ni avanzados, simplemente sacian las consultas de las dudas que pueden surgir programando en cualquier sitio, en el día a día.

Cuando estas lejos de tu puesto de trabajo, internet no está ahí o simplemente la red no funciona el guerrillero informático tiene el manual en el bolsillo, la botella de agua en la mochila y el portátil cruzado en la espalda.

Autor: Jesús Benages Sales

Contacto: jobinary@hotmail.com

