**Number(*dato*)** Función que devuelve el resultado de convertir *dato* en un número (que puede ser NaN Not a Number).

**isNaN**(valor) devuelve true cuando valor vale NaN y false en caso contrario

**parseInt(*cadena*)** Convierte cadena en entero**,** si no puede devuelve NaN

**parseFloat(*cadena*)** Convierte cadena en número decimal**,** si no puede devuelve NaN

**ARRAYS**

**var nombre\_array = new Array ();**

var vocales = new Array ('a','e','i','o','u');

alert (vocals);

for (i=0;i<**vocales.length**;i++)

document.write (vocales[i]);

var tabla = new Array (3); // tabla de 3 filas y 5 columnas

for (i=0;i<3;i++)

tabla[i]=new Array(5);

Métodos:

**a1.indexOf (valor)** Devuelve el índice que ocupa el elemento *valor* en a1, -1 si no lo encuentra

**a1.lastIndexOf (valor)** Devuelve el índice que ocupa el último *valor* en a1, -1 si no lo encuentra

**a1.join(separador)** Devuelve una cadena de caracteres compuesta por los elementos de a1 separados mediante la cadena *separador*.

**a1.sort()** Ordena alfabéticamente

**a1.concat(a2)** Devuelve un array con los elementos de a1 seguidos de los de a2

**a1.shift()** Elimina el primer elemento de a1 y devuelve este elemento

**a1.pop()** Elimina el último elemento de a1 y devuelve este elemento

**a1.reverse()** Invierte el orden de los elementos de un array

**a1.unshift(*elemento*[*,elemento,…*])**Añade nuevos elementos al inicio de a1 y devuelve la longitud de a1

**a1.push(*elemento*[*,elemento,…*])**Añade nuevos elementos al final de un a1 y devuelve la longitud de a1

**a1.slice(inicio[,fin])** Devuelve los elementos comprendidos entre el índice *inicio* incluido y el índice *fin* excluido de a1.

**a1.splice(i,n)** Elimina n elementos desde el índice i (n puede valer 0)

**a1.splice(i,n, *elemento*[*,elemento,…*]))** Además añade elemento a partir de i

**El objeto String**

Su única propiedad es ***length***que contiene el número de caracteres que componen la cadena

Métodos:

**cadena.charAt(posicion)** Devuelve el carácter que ocupa la posición *posición*.

Como en arrays:

**cadena.concat(*cadena1*,…,*cadenaN***)

**cadena.indexOf(*aguja*[*,inicio*])**

**cadena.lastIndexOf(*aguja*[*,fin*])**

**cadena.slice(*inicio*[,*fin*])**

**cadena.split (separador)** Devuelve un array cuyos elementos son los fragmentos de la cadena comprendidos entre el carácter o cadena *separador*.

**cadena.substr (inicio[,cantidad])** Devuelve *cantidad* número de caracteres a partir de *inicio* (incluido)

**cadena.substring (inicio[,final])** Devuelve *cantidad* número de caracteres a partir de *inicio* (incluido).

**cadena.toLowerCase()** *Devuelve* la cadena convertida en minúsculas.

**Cadena.toUpperCase()** *Devuelve* la cadena convertida a mayúsculas.

**cadena.trim() , cadena.trimLeft(),cadena.trimRight()** Eliminan los espacios al principio y al final de la cadena, al principio solamente o al final solamente, respectivamente.

**cadena.search(cadena1)** Busca una coincidencia en una cadena y devuelve la posición de la coincidencia

**cadena.replace(cadena1, cadena2)** Busca una coincidencia en una cadena y si existe la reemplaza por otra cadena pasada por parámetro.

**El objeto Date**

var fecha = new Date(); fecha contendrá la fecha y hora del sistema

var fecha = new Date(*milisegundosDesdeEpocaUNIX*); construye la fecha correspondiente

var fecha = new Date(año,mes,dia[,hora,minuto,segundo,milisegundo]);

Donde todos los parámetros son números enteros. El año debe expresarse con 4 cifras. El mes es un entero entre 0 (enero) y 11 (diciembre). El día es un entero entre 1 y 31. hora, minuto, segundo, milisegundo no son obligatorios.

Para saber si una fecha se ha construido correctamente podemos preguntar por **isNaN(Fecha)** de devolverá false o true dependiendo de que lo haya sido o no.

Métodos:

**fecha.getDate()** Devuelve el día del mes entre 1 y 31.

**fecha.getDay()** Devuelve el día de la semana (0-6). El 0 corresponde al domingo

**fecha.getMonth()** Devuelve el mes (0-11).

**fecha.getFullYear()** Devuelve el año expresado con 4 dígitos. Existe el método

**fecha.getHours()** Devuelve la hora (0-23).

**fecha.getMinutes()** Devuelve los minutos (0-59)

**fecha.getSeconds()** Devuelve los segundos (0-59)

**fecha.getMiliseconds()** Devuelve los milisegundos (0-59)

**fecha.getTime()** Devuelve los milisegundos transcurridos desde la época UNIX

hasta la fecha a la que se aplica

**fecha.setDate()** Establece el día del mes.

**fecha.setMonth()** Establece el mes (0-11).

**fecha.setFullYear()** Establece el año expresado con 4 dígitos.

**fecha.setHours()** Establece la hora (0-23).

**fecha.setMinutes()** Establece los minutos (0-59).

**fecha.setSeconds()** Establece los segundos (0-59).

**fecha.setMiliseconds()** Establece los milisegundos (0-999).

**fecha.setTime ()** Establece la fecha como el número de milisegundos transcurrido desde la época UNIX

**fecha.toDateString()** Convierte la fecha del objeto Date en una cadena de caracteres

**fecha.toTimeString()** Convierte el tiempo del objeto Date en una cadena de caracteres

**El objeto Math**

Métodos

**Math.ceil (valor)** Devuelve el entero inmediatamente mayor o igual que *valor*.

**Math.floor(valor)** Devuelve el entero inmediatamente menor o igual que *valor*.

**Math.max(val1, …, valN)** Devuelve el máximo de los valores recibidos como argumentos

**Math.min(val1, …, valN)** Devuelve el mínimo de los valores recibidos como argumentos

**Math.pow(base,exponente)** Devuelve el resultado de elevar base a exponente

**Math.random()** Devuelve un número aleatorio entre 0 y 1, 1 no incluido

**Math.round(valor)** Devuelve el resultado de redondear valor al entero más próximo

**Math.sqrt(valor)** Devuelve la raíz cuadrada de valor.

Devuelven **NaN** cuando no han podido efectuar el cálculo matemático

**Ejemplo función para generar números aleatorios en un intervalo (extremos incluidos)**

function aleatorio (min,max){

return Math.floor(Math.random()\*(max - min +1)+min);

}

**El objeto Number**

**num.toFixed(n)** Formatea el número con n dígitos decimales

**Formularios:**

03 Ejemplo Formulario.html

**Expresiones regulares:**

Vosotros

**Objetos del navegador**

**Window**

* Abrir y cerrar una ventana (**open(), close()**), escribir en ella, poner el foco (**focus())**
* Métodos **alert, confirm y prompt**
* Mover y redimensionar una ventana (Métodos **resizeBy, resizeTo, moveBy, moveTo**)
* Temporizadores
  + **setInterval, clearInterval**
  + **setTImeout**

**Location**

* Propiedad **href**
* Métodos **:** **assign(),reload(),replace()**

**Gestión de eventos**

Uso de:

**variable this (**para referirse al elemento HTML que ha provocado el evento)

**objeto event y su propiedad type** (para referirse al evento)

Eventos y manejadores de eventos más comunes:

**onclick, onmouseover, onmouseout**

**onblur, onchange, onfocus, onload, onreset, onselect, onresize, onsubmit**

**onkeydown, onkeypress, onkeyup**

*Para realizar una acción al salir de una caja (input text), ya sea porque han pulsado intro o porque han salido mediante el tabulador (la caja pierde el foco) podemos hacer lo siguiente:*

*<input type=”text” onblur=”realizarAcccion()” onkeypress=”if (event.keyCode==13) realizarAccion()”>*

**DOM**

**Acceso a sus elementos:**

**var e = document.getElementById (“id”);**

**Y una vez en el elemento a cualquiera de sus propiedades**

**e.value (ej. para un <input text>)**

**e.innerHTML (ej. para un <div>, <p>,...)**

**e.id**

**...**

**e.style.propiedad propiedades de estilo**

**var e = document.getElementsByName()**

Devuelve los elementos que comparten el mismo valor del atributo name

**var e = document.getElementsByTagName()**

Devuelve los elementos HTML de un tipo

**var e = document.getElementsByClassName(),**

Devuelve los elementos que pertenecen a una clase determinada

**e.length**

**e[i].value**

**e[i].innerHTML**

**e[i].id**

**e[i].style.propiedad**

**…**

**e=document.forms** -> colección con todos los formularios del documento

e[i].reset() para inicializar

**e=document.images** -> colección con todas las imágenes del documento

**e=document.links** -> colección de todos los enlaces del documento

**Gestionar los atributos de un elemento**

* **getAttribute** (nombre) lee el valor del atributo nombre
* **setAttribute**(nombre,valor) crea/modifica el atributo nombre
* **removeAttribute**(nombre) elimina el atributo nombre
* **hasAttribute**(nombre) averigua si existe el atributo nombre

**Explorar el arbol de nodos**

<html>

<head>

<script type="text/JavaScript">

function comprobar(){

e=document.getElementById("id1");

alert(e.**hasChildNodes());** // true o false según tenga o no nodos hijos

hijos=e.**childNodes**; // son los nodos hijos de e

alert(hijos.length); // cuantos tiene

alert(e.**firstChild**.innerHTML); // es el hijo mayor o primero

alert(e.**lastChild**.innerHTML); // es el hijo menor o último

p=document.getElementById("p2");

alert(p.**previousSibling**.innerHTML); // es el hermano izquierdo o anterior de p

alert(p.**nextSibling**.innerHTML); // es el hermano derecho o siguiente de p

alert(p.**parentNode**.id); // es el nodo padre de p

}

</script>

</head>

<body onload="comprobar()">

<div id="id1"><p> Primero </p><p id="p2"> Segundo </p><p> Tercero </p></div>

</body>

</html>

**Creación y eliminación de nodos**

* **document.createElement(*etiqueta*)**

Devuelve un objeto de tipo nodo del tipo de elemento correspondiente a la *etiqueta* HTML. Este método crea el nodo, pero no lo inserta en ninguna posición del DOM, por lo que no será visible hasta que lo ubiquemos.

* **appendChild(nodo)**

Inserta el *nodo* en el DOM como hijo último del elemento actual.

* **insertBefore(nodo1,nodo2)**

Inserta el *nodo1* como hermano inmediatamente mayor de *nodo2* dentro del elemento actual.

* **removeChild(nodo)**

Elimina el nodo del DOM y lo devuelve como una referencia que podemos recoger en una variable para rescatarlo posteriormente.

* **replaceChild(nuevo,viejo)**

Sustituye el nodo *viejo* por el *nuevo* dentro del elemento actual.

<html>

<head>

<script>

var n=1;

function anadir(){

var p=document.**createElement**('p');

p.innerHTML='Este párrafo no existía en el HTML original '+n;

e=document.getElementById("principal");

e.**appendChild**(p);

n++;

}

function eliminar(){

var p=document.getElementsByTagName('p')[0];

e=document.getElementById("principal");

e.**removeChild**(p);

}

function cambiar(){

var p1=document.getElementsByTagName('p')[0];

var p=document.createElement('p');

p.innerHTML='Este párrafo sustituye al anterior';

e=document.getElementById("principal");

e.**replaceChild**(p,p1);

}

function intercalar(){

var p1=document.getElementsByTagName('p')[1];

var p=document.createElement('p');

p.innerHTML='Este se intercala';

e=document.getElementById("principal");

e.**insertBefore**(p,p1);

}

</script>

</head>

<body>

<input type="button" value="Añadir nodo" onclick="anadir()">

<input type="button" value="Elimina nodo" onclick="eliminar()">

<input type="button" value="Cambiar nodo" onclick="cambiar()">

<input type="button" value="Poner nodo entre 1º y 2º" onclick="intercalar()">

<div id="principal">

<p>Primer párrafo</p>

<p>Segundo párrafo</p>

<p>Tercer párrafo</p>

</div>

</body>

</html>