LENGUAJES INTERPRETADOS EN EL CLIENTE

SEMANA 6

SENTENCIAS REPETITIVAS

- En JavaScript disponemos de los bucles for(), do-while() y while()

Descripción	for()	do-while	While
Sintaxis	for (expresión Inicial; condicion; expresión Incremento){ // sentencias a ejecutar }	do { *//sentencia* } while (*condición*);	while (*condicion*){ *//sentencia* *}*
Ejemplo	for (i=1;i<=5;i++) { n += i; funcion(n); }	do { i += 1; document.write(i); } while (i < 5);	while (i < 5) { alert(i); i ++; }

DEFINICIÓN DE ARREGLO O MATRIZ

- Al igual que en otros lenguajes un arreglo o matriz es un tipo de dato compuesto
- Cada valor almacenado en un arreglo es denominado elemento y cada elemento tiene una posición, que puede ser numérica o no. Este indicador de posición del elemento es denominado índice del arreglo o matriz.
- Los elementos de una matriz en JavaScript comienzan desde la posición 0 y no la 1, cuando hablamos de arreglos indexados numéricamente.

ASPECTOS BÁSICOS SOBRE MATRICES

- En JavaScript, además de los matrices indexados numéricamente, existen las matrices o arreglos asociativos.
- Una matriz asociativa es aquella cuyos elementos son referenciados por índices de tipo cadena, en lugar de ser referenciados por índices numéricos.
- Para acceder a un elemento de una matriz debe utilizarse el nombre de la matriz y, a continuación, y entre corchetes su índice numérico o asociativo.

MATRICES Y ARREGLOS

- Los arreglos (matrices) se emplean para almacenar múltiples valores en una sola variable, en JavaScript, una matriz también funciona como una lista, una pila o una cola. Los elementos en una matriz tienen su propia enumeración o sub-índice, con el que podemos acceder a un elemento del array.
- Las matrices se construyen con corchetes, que contiene una lista de elementos separados por comas.

- Sintaxis

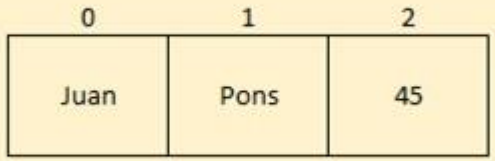
Var arreglo = new Array(); ó var arreglo = [];

Ejemplo

var colors= [“Red", “Blue", “Green", “White"];

alert(colors[o]); // Red

var miVector3 = new Array("Juan", "Pons", 45);

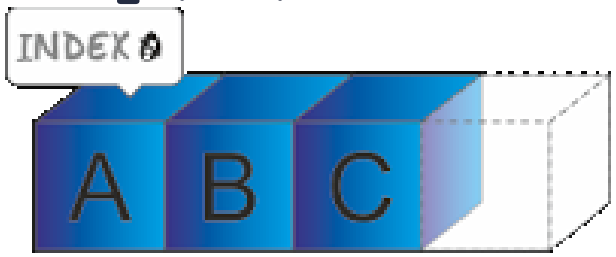


Representación gráfica de un array de 3 posiciones

ESPECIFICAR EL TAMAÑO DE LA MATRIZ

- Es posible especificar el **número de elementos** que podrá contener una matriz al momento de declararlo si se especifica entre corchetes o entre los paréntesis del constructor Array, un solo valor numérico:

```
var enteros = [25];
var enteros = new Array(25);
```



ASIGNAR ELEMENTOS A UNA MATRIZ

- Para introducir valores en un arreglo puede utilizar varias estrategias:
  1. Asignando un dato a un elemento del arreglo de forma directa.
  2. Asignando una lista de valores de una vez en el arreglo.
  3. Pasando los argumentos en el constructor Array de una sola vez.

ASIGNAR ELEMENTOS A UNA MATRIZ

```
//Directamente
musica[0] = "rock";
musica[1] = "pop";
musica[2] = "disco";

//Utilizando corchetes
musica = ["rock", "pop", "disco"];

//Utilizando el constructor Array()
musica = new Array("rock", "pop", "disco");
```

ASIGNAR ELEMENTOS A UNA MATRIZ

- La asignación a los elementos de un arreglo no se hace necesariamente en el código fuente.
- Por lo general, los valores a los elementos se asignarán cuando se esté ejecutando el script por parte del usuario.
- Para asignar valores a una matriz por parte de los usuarios, resulta conveniente hacer uso de sentencias repetitivas como el ***for***, el ***while*** o el ***do-while***.

ASIGNAR ELEMENTOS A UNA MATRIZ

```
//Utilizando lazo for
for(var i=0; i<10; i++){
   notas[i] += parseFloat(prompt("Nota: ",""));
}
//Utilizando lazo while
var i=0;
while(i<10){
   notas[i] += parseFloat("Nota: ","));
   i++;
}
```

ASIGNAR ELEMENTOS A UNA MATRIZ

- Para acceder a los elementos almacenados en un arreglo se utiliza el identificador del arreglo y a continuación, entre corchetes el índice del arreglo, numérico o asociativo.

```
mayor = numeros[10];

monedasv = monedas['elsalvador'];

alert(notas['apsI']['ricardoelias']);
```

- También pueden utilizarse los ciclos ***for***, ***while***, ***do-while*** y ***for-in*** para hacer un recorrido por todos o por algunos de los elementos de un arreglo.

ASIGNAR ELEMENTOS A UNA MATRIZ

- Una precaución con respecto al acceso a los elementos de una matriz es tener el cuidado de no acceder a posiciones de la matriz no establecidas o indefinidas.
- Esto se debe a que JavaScript permite asignar valores a índices de arreglo de forma no consecutiva.
- El valor que será mostrado cuando se acceda a una posición de arreglo no definida será el valor ***undifined***.

AÑADIR O MODIFICAR ELEMENTOS DE UNA MATRIZ

- Para agregar un nuevo elemento a una matriz no es necesario asignar más memoria para en el arreglo.
- Los nuevos valores se agregan de forma directa, tal y como se crean la primera vez.
- Es por esto que no tiene sentido definir un arreglo como de un tamaño específico, el único propósito de hacerlo sería la claridad del código.

AÑADIR O MODIFICAR ELEMENTOS A UNA MATRIZ

- Ejemplo:

```
var planetas = ["Marte", "Saturno", "Júpiter"];

var jovianos = planetas;

jovianos[0] = "Neptuno";
```

- Al acceder a los elementos del arreglo planetas se nos mostrarían los planetas: Neptuno, Saturno y Júpiter. El valor de Marte sería reemplazado por el de neptuno.
- La razón es que se al hacer la asignación de una variable arreglo a otra en la instrucción **jovianos = planetas**, la asignación se realiza por referencia.

ELIMINAR ELEMENTOS DE UNA MATRIZ

- Para eliminar un elemento específico de una matriz puede utilizar el operador ***delete***.
- Este operador establece el elemento invocado al valor ***undefined***.
- El tamaño del arreglo no es modificado al utilizar el operador ***delete***.

ELIMINAR ELEMENTOS DE UNA MATRIZ

- Ejemplo:

```
var colores = ["rojo", "verde", "azul"];
delete colores[1];
alert("El valor de colores[1] es: " + colores[1]);
```

El valor del tamaño del arreglo seguirá siendo 3, incluso si se borra el último elemento del arreglo colores.

OBTENER EL TAMAÑO DE UNA MATRIZ

- Para obtener el tamaño de un arreglo se puede utilizar la propiedad ***length*** para facilitar la tarea.
- La propiedad ***length*** recupera el índice de la siguiente posición disponible (sin ocupar) al final del arreglo.
- Este valor obtiene el índice de esa posición incluso si existen posiciones con índice menor sin ocupar.
- En otras palabras, ***length*** devuelve el índice del primer espacio disponible después del último elemento establecido con un valor.

OBTENER EL TAMAÑO DE UNA MATRIZ

- Ejemplo:

```
var saludos = new Array();
saludos[5] = "Buenas noches";
alert(saludos.length);
```

- El valor devuelto al aplicar ***length*** al arreglo saludos será 6, aunque únicamente se haya establecido valor para el índice 5 y a pesar de que los índices 0, 1, 2, 3 y 4 estén indefinidos.

ARREGLOS O MATRICES ASOCIATIVAS

- En JavaScript los índices de los arreglos pueden ser literales de cadena y no solamente números como en otros lenguajes.
- Para poder recorrer los elementos de una matriz indexada con cadenas puede utilizarse la sentencia repetitiva *for-in*, al igual que como se hace con objetos. De hecho en JavaScript una matriz es considerada como un tipo especial de objeto.

ARREGLOS O MATRICES ASOCIATIVAS

- Ejemplo:

```
var tags = new Array();
 tags['html']  = "Inicio de un documento HTML";
 tags['head']  = "Cabecera del documento HTML";
 tags['title'] = "Título del documento HTML";
 tags['body']  = "Cuerpo del documento HTML";
 …
 for(etiqueta in tags){
    tabla += "<tr>";
    tabla += "<td>" + etiqueta + "</td>\n<td>" + tags[etiqueta] +
"</td>\n";
    tabla += "</tr>";
 }
```

MATRICES MULTIDIMENSIONALES

- JavaScript admite trabajar con matrices multidimensionales; sin embargo, estas no están definidas explícitamente en el núcleo del lenguaje.
- Para poder simular el trabajo con matrices multidimensionales JavaScript utiliza matrices de matrices. Lo que significa que se pueden definir como elementos de una matriz, otra matriz.

MATRICES MULTIDIMENSIONALES

```
var i,j;
var paresimpares = [
        [0,2,4,6,8,10,12,14,16,18,20],
        [1,3,5,7,9,11,13,15,17,19]
];
for(i=0; i<paresimpares.length; i++){
   document.write("<h4>[");
   for(j=0; j<paresimpares[i].length; j++)
      document.write(" " + paresimpares[i][j] + " ");
   document.write("]</h4>");
}
```

MÉTODOS PARA EL MANEJO DE MATRICES

- **join**(). Convierte todos los elementos de una matriz en cadenas y luego, los concatena. Se puede especificar en un argumento opcional el carácter de separación entre los elementos. Si no se especifica se asume por defecto la coma como caracter de separación.

```
var num = [1, 2, 3];
var strnum = num.join();   //Salida: s="1,2,3"
var letras = ['a', 'e', 'i', 'o', 'u'];
var strlet = letras.join("-"); //Salida:strlet="a-e-i-o-u"
```

MÉTODOS PARA EL MANEJO DE MATRICES

- **concat**(). Crea y devuelve una matriz que contiene los elementos de la matriz original sobre la que se ha aplicado el método concat(), seguidos por cada uno de los argumentos proporcionados en concat().

```
var impares = [1,3,5];

alert(impares.concat(7,9));//Salida:1,3,5,7,9
```

MÉTODOS PARA EL MANEJO DE MATRICES

- **sort**(). Ordena los elementos de una matriz de forma lexicográfica. Esto lo hace convirtiendo primero los elementos de la matriz en cadenas y luego aplica el orden lexicográfico.

```
var numeros = [14,52,3,45,36];

alert(numeros.sort());//Salida:14,3,36,45,52
```