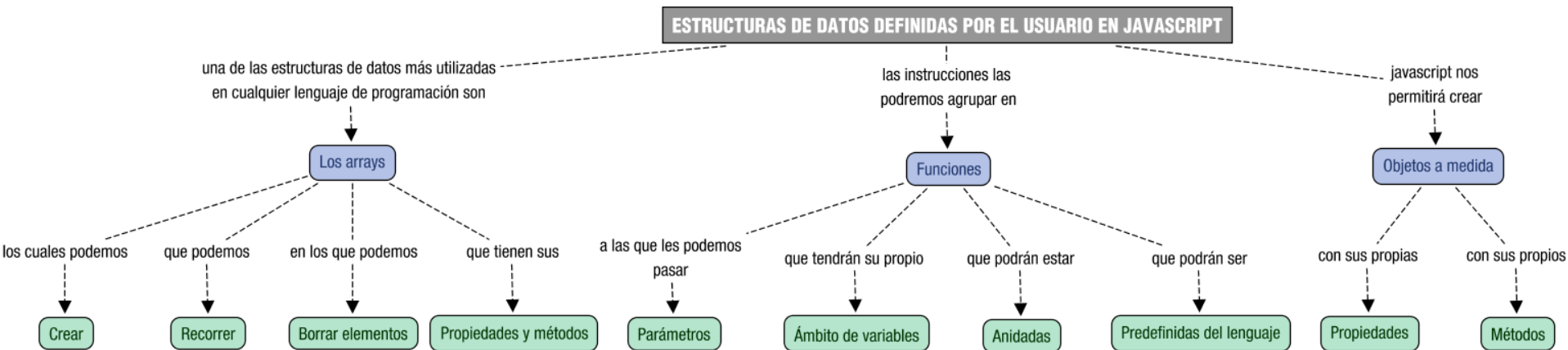


ESTRUCTURAS DEFINIDAS POR EL USUARIO EN JAVASCRIPT

Unidad didáctica 3

Mapa conceptual



Estructuras definidas por el usuario en JavaScript

- Estructuras de datos
 - Objeto Array
 - Creación e introducción de datos
 - Recorrido
 - Borrado de elementos
 - Propiedades
 - Métodos
 - Arrays paralelos
 - Arrays multidimensionales
- Funciones
 - Definición
 - Parámetros
 - Ámbito de las variables
 - Funciones anidadas
 - Funciones predefinidas
- Objetos de usuario
 - Creación
 - Propiedades
 - Métodos
 - Objetos literales

Estructuras de datos

Sirven para guardar información más compleja que una variable simple

Estructuras

Listas

Pilas

Colas

Árboles

Conjuntos

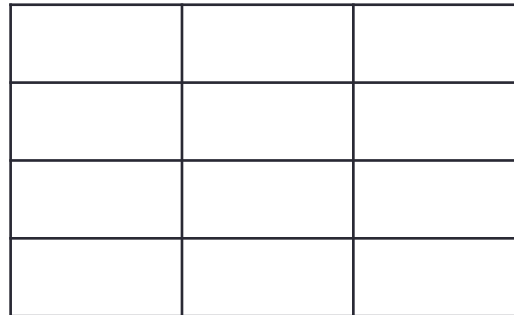
Array

Array - Matriz - Vector

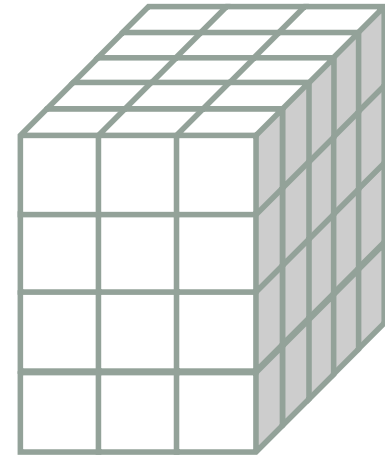
Variable o zona de almacenamiento continua donde se puede guardar más de un valor



1 dimensión



2 dimensiones



3 dimensiones

Array - Matriz - Vector

Multidimensión

Array

ÍNDICE

Referencia de un elemento del Array

Si el índice es numérico ➡ Indexación Base-Cero(0)

Colección ordenada de datos

JavaScript gestiona con arrays los objetos HTML del documento : *links, forms, anchors, images*

Creación e inicialización de un Array

- Constructor (new Array)

```
var misAmigos = new Array();  
    misAmigos[0] = "Pepe";  
    misAmigos[2] = "Ana";
```

// array de longitud 0

// array de longitud 1

// array de longitud 3

???

```
var misAmigos = new Array(40);  
    misAmigos[50] = "Adelino";
```

// array de longitud

???

// array de longitud

???

```
var misAmigos = new Array("Pepe","Juan","Ana"); // array denso
```

- De forma literal *!Cuidado en navegadores antiguos!*

```
var misAmigos = ["Pepe","Juan","Ana"];
```



- Array mixto u Objeto literal

```
var misAmigos = {2:"Pepe",1:"Juan",0:"Ana"};
```

```
var misEnemigos = {"primero":"Cansancio","2":"Dolor",3:13,"cuarto":"Falsos"};
```


Creación e inicialización de un Array

- Array mixto u Objeto literal

```
var misAmigos = {2:"Pepe",1:"Juan",0:"Ana"};
```

```
var misEnemigos = {"primero":"Cansancio","2":"Dolor",3:13,"cuarto":"Falsos"};
```

Otra forma más clara de escribir los arrays mixtos:

```
var misEnemigos = {  
    "primero":"Cansancio",  
    "2":"Dolor",  
    3:13,  
    "cuarto":"Falsos"  
};
```

No olvidar que un array mixto es un objeto literal y de esta forma vemos claramente las propiedades del objeto.

Recorrido de un Array

```
var misAmigos = ["Pepe","Juan","Ana"];
```

```
for (var i=1;i<=3;i++) {  
    alert("mi amigo nº "+i+" se llama "+misAmigos[i-1]);  
}
```

```
for (var i=0;i<3;i++) {  
    alert("mi amigo nº "+(i+1)+" se llama "+misAmigos[i]);  
}
```

```
var i=0;  
while (i<3) {  
    alert("mi amigo nº "+(i+1)+" se llama "+misAmigos[i]),  
    i++;  
}
```

```
for (var amigo in misAmigos) {  
    alert("tengo un amigo llamado "+misAmigos[amigo]);  
}
```

Los arrays de tipo mixto hay que recorrerlos obligatoriamente con for...in ya que no tienen propiedad length.

La propiedad length es siempre el índice cuyo valor es más alto.

Borrado de un elemento de un Array

```
var misAmigos = ["Pepe","Juan","Ana"];
```

```
delete misAmigos[0]; // la longitud del array sigue siendo 3 (el 0 es undefined)
```

```
for (var amigo in misAmigos) {  
    if (misAmigos[amigo]=="Juan") {  
        delete misAmigos[amigo];  
    }  
}
```

Al borrar un elemento del array, se eliminará su índice de la lista de índices del array, pero no se reducirá la longitud del array.

Para un control de la eliminación utilizar el método splice().

El operador delete, se recomienda para arrays que usen texto como índices del array, ya que de esta forma se producirán menos confusiones a la hora de borrar los elementos.

No implica liberación de memoria

El intérprete de JavaScript del navegador es el encargado de gestionar la colección de basura en memoria,

Actividad

Ejercicio 1

Haz un programa que solicite al usuario por pantalla 10 números mayores de 20 que almacenarás en un array

Al terminar deberá visualizar todos los números a excepción del mayor de ellos que deberá haberlo borrado.

Modo de hacerlo

Primero: definir el array

Segundo: pedir números que tendrán que ser mayores a 20 hasta tener los 10

Tercero: buscar el índice donde se encuentra el mayor de todos

Cuarto: borrar ese índice

Quinto: presentar el array por pantalla

Alternativa: suprimir Tercero y modificar Segundo para almacenar la posición donde se encuentra el mayor hasta el momento a medida que se van introduciendo los números