



Linkia FP

DAW	M06	T02	Material	Estructuras de almacenamiento
-----	-----	-----	----------	-------------------------------

Declaramos una función con la palabra clave “function”

```
function sumaValores (x , y) {  
    return x + y;  
}  
console.log( sumaValores(3,4) ); //retorna 7
```

Funciones como objetos

En JavaScript las funciones son un tipo de objetos

```
realizaOperacion = function ( operacion, x, y) {  
    return operacion(x,y);  
}  
console.log( realizaOperacion(sumaValores, x, y ) );
```

Las podemos declarar como un objeto

```
let mifunction = new Function('param1', 'param2', 'param3',  
    `console.log("Hola")  
    console.log("Adios:"+param3)` );  
mifunction(3, 4, 7);
```

Función anónima

Podemos crear una función sin nombre y asignarla a una variable

```
var sumaNumeros = function(x,y){  
  return x+y;  
}  
console.log( sumaNumeros(3,4) );
```

Podemos pasar como parámetro una función sin nombre

```
function sumaNumeros (x,y, objetoTipoFuncion) {  
  return objetoTipoFuncion(x,y);  
}  
console.log( sumaNumeros(3,4, function(x,y){ return x+y; }) );
```

Función flecha

No tienen su propio *this*.

Se deben definir antes de utilizar

```
const restaNumeros = (x,y) =>{ return x-y; }  
  
console.log( restaNumeros(3,4) );
```

Definidos en la misma declaración de parámetros

Podemos indicar un valor per defecto asignando a la variable un valor

```
function sumaValores (x=2, y=1) {  
    return x + y;  
}  
console.log(sumaValores() ); //retorna 3  
console.log(sumaValores(3,4) ); //retorna 7
```

Programados en la función

Si el operador OR no recibe como primer valor un *true*, retorna el segundo valor


Como si no se indica un valor es “undefined”, retornará el segundo valor.

```
function sumaValores (x, y) {  
    x = x || 2;  
    y = y || 1;  
    return x + y;  
}  
console.log(sumaValores() ); //retorna 3  
console.log(sumaValores(3,4) ); //retorna 7
```

Como declarar una Array en JS ?

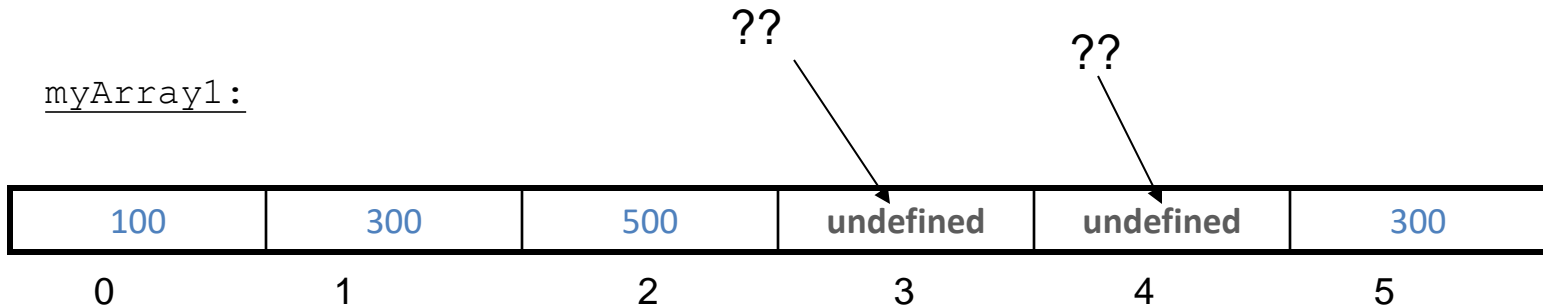
- Indicando el numero de elementos en el constructor Array()

```
var myArray1 = new Array(6);
myArray1[0] = 100;
myArray1[1] = 300;
myArray1[2] = 500;
myArray1[5] = 300;
```



myArray1:

			??	??	
100	300	500	undefined	undefined	300
0	1	2	3	4	5



- Sin indicar el numero de elementos

```
var myArray2 = new Array();
```

```
myArray2[0] = "precio";
```

```
myArray2[1] = 300;
```

```
myArray2[5] = 500;
```

Podría hacer esto?

"precio";	300				500
0	1	2	3	4	5

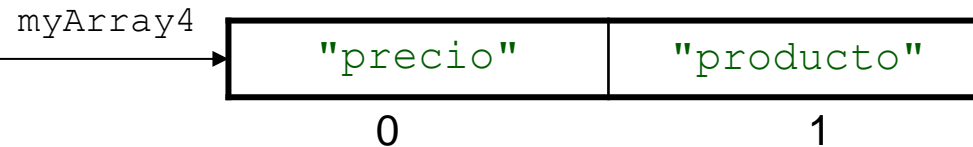
- Con el constructor Array

```
var myArray3 = new Array("precio", "producto");
```



- Como un literal

```
var myArray4 = ["precio", "producto"];
```



- Copiando de otro array

```
var myArray5 = Object.assign([], myArray4);
```



- Bucle FOR: recorre todas las posiciones numéricas aunque estén con “undefined”

```
let miArray =["primer valor",88];  
miArray[5]="último valor";  
  
for(let k=0;k < miArray.length; k++){  
    console.log(k +":"+miArray[k]);  
} //33,88,undefined,undefined, undefined ,último valor
```

- Bucle ForEach recorre solo las posiciones numéricas sin valores “undefined”

```
let miArray =["primer valor",88];  
miArray[5]="último valor";  
  
miArray.forEach(function (elemento, indice, array) {  
    console.log(indice+":"+elemento);  
});  
//primer valor,88 ,último valor
```


- **concat()** : concatena 2 arrays.
- **indexOf()**: busca un valor.
- **push()**: añade un valor al final.
Retorna la nueva longitud
- **unshift()**: añade un valor al inicio.
Retorna la nueva longitud
- **pop()**: quita y retorna el último valor.
- **shift()**: quita y retorna el primer valor.
- **splice()**: permite alterar un array añadiendo y/o borrando elementos.

```
var hombres = ["Pablo", "Jose"] ;  
var mujeres = ["Lucia", "Gemma", "Zelda"] ;  
var humanos = hombres.concat(mujeres) ;
```

```
hombres.lastIndexOf("Jose")
```

```
humanos.push("William") ;
```

```
humanos.unshift("Laura") ;
```

```
var n = humanos.pop() ;  
console.log(humanos.pop()) ;
```

```
console.log(humanos.shift()) ;
```

- En JS las funciones son objetos:
- Pueden tener variables.
- Pueden tener otras funciones.
- Pueden almacenar-se en una variable.
- Pueden crear-se varias instancias de una función.
- Pueden heredar métodos y atributos de otras funciones.

```
var suma = new SumaAleatoria(10);
alert(suma.resultado);
suma.sumaNumero(5);
alert(suma.resultado);

function SumaAleatoria(valor){
    this.resultado="0";
    var alteracion=Math.floor(Math.random()*10);

    function alteraSuma(){
        valor=alteracion+valor;
        return valor;
    }
    this.resultado = alteraSuma();
    alert(valor);

    this.sumaNumero = function (numero){
        this.resultado+=numero;
    };
}
```

- Podemos definir una clase con **class**.
- Puede contener un constructor

```
class SumaAleatoria {
  constructor(valor){
    this.resultado="0";
    var alteracion=Math.floor(Math.random()*10);

    function alteraSuma(){
      valor=alteracion+valor;
      return valor;
    }
    this.resultado = alteraSuma();
    alert(valor);
  }
  sumaNumero = function (numero){
    this.resultado+=numero;
  };
}

var suma = new SumaAleatoria(10);
alert(suma.resultado);
suma.sumaNumero(5);
alert(suma.resultado);
```

```
var arrayObjeto= {  
  'nombre': 'Alf!',  
  'consigna': 'Zumo de gato!'  
};
```

En JS un array asociativa es un objeto

```
var arrayObjeto = new Array();  
arrayObjeto['nombre'] = 'Alf!';  
arrayObjeto['consigna'] = 'Zumo de gato!';
```

```
var txt = associativeArray1['nombre'];
```

- **for/in**: recorre los elementos almacenados en keys asociativas. (propiedades de un objeto)

```
for (var key in arrayObjeto) {  
  var valor = arrayObjeto[key];  
}
```

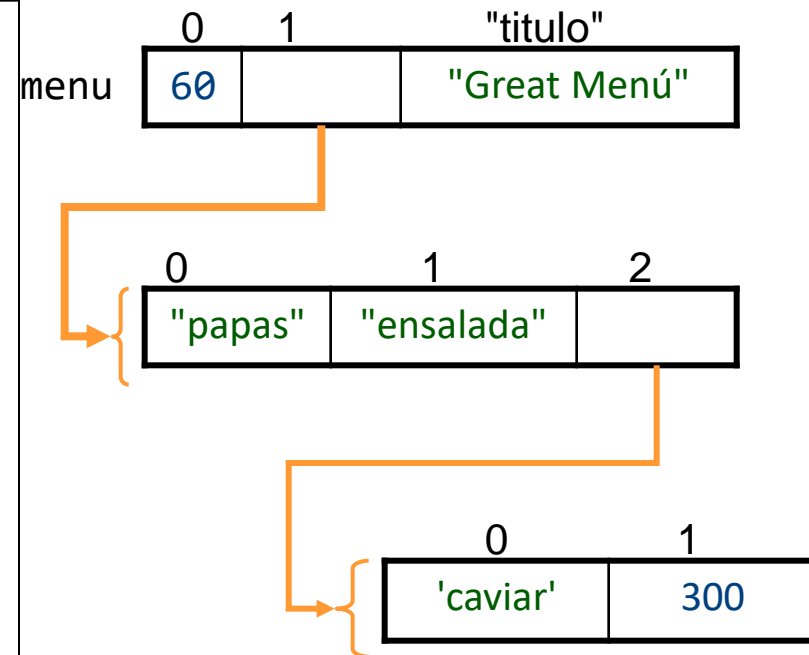
- **delete**: elimina un elemento de un objeto según su key

```
delete arrayObjeto['nombre'];
```

```
var menu = new Array();
menu[0]=60;
menu["titulo"] ="Great Menú";
menu[1] = new Array("papas", "ensalada");
```

```
var primeros = new Array();
primeros[0]="papas";
primeros[1]="ensalada";
menu[1] = primeros;
```

```
primeros[2]= new Array();
primeros[2][0]="caviar";
primeros[2][1]= 300;
```



```
primeros[2]= ["caviar", 300];
menu[1][2]= ["caviar", 300];
```

```
var myArray2 = new Array();  
myArray2[0] = "precio";  
myArray2[1] = 300;
```

```
var myArray3 = new Array("precio", 300);
```

```
var myArray4 = ["precio", 300];
```

```
var associativeArray1 = new Array();  
associativeArray1['nombre']='Alf!';  
associativeArray1['consigna']='Zumo de gato!';  
associativeArray1[0]=23;
```

```
var associativeArray1 = {  
  'nombre': 'Alf!',  
  'consigna': 'Zumo de gato!'  
};
```



Linkia FP

Formación Profesional
Oficial a Distancia