

# JavaScript

## 8 – Map y Set

# Map

- Asociación clave-valor
- El índice o clave puede ser de cualquier tipo

```
let map = new Map( );  
map.set( "Marcos", 33 );  
map.set( "Ana", 25 );  
  
console.log(map.size); // 2  
console.log(map.get( "Ana" )); // 25  
console.log(map.has( "Marcos" )); // true  
map.delete( "Marcos" );
```

## Iterar sobre Map

- Podemos utilizar el bucle `for..of` o el método `forEach`

**let** hobbies = **new** Map(); // Almacenará una persona con un array de hobbies (string)

`hobbies.set("Juan", ["Tenis", "Informática", "Cine"]);`

`hobbies.set("María", ["Música", "Caminar"]);`

`hobbies.set("Paco", ["Boxeo", "Comer", "Dormir"]);`

```
for(let entry of hobbies) {  
  console.log(entry[0] + ": " + entry[1].join(", "));  
}
```

```
for(let [nombre, hobArray] of hobbies) { // Mejor  
  console.log(nombre + ": " + hobArray.join(", "));  
}
```

```
hobbies.forEach((hobArray, nombre) => { // Mejor  
  console.log(nombre + ": " + hobArray.join(", "));  
});
```

# Set

- Similar a Map → No guarda valores, sólo claves

- Concepto de almacén
  - No hay valores repetidos
  - Un valor está o no está

```
let set = new Set( );  
set.add( "Ana" );  
set.add( "Pepe" );  
set.add( "Juan" );  
set.add( "Irene" );  
set.delete( "Juan" );  
console.log( set.size ); // 3
```

```
set.add( "Pepe" ); // Ya existe, no añade  
console.log( set.size ); // 3  
console.log( set.has( "Irene" ); // true
```

## Iterar sobre Set

- Similar a iterar sobre Map

```
for(let value of set) {  
    console.log(value);  
}
```

```
set.forEach(value => {  
    console.log(value);  
});
```