

Conjuntos y Mapas en JavaScript:

Conjuntos

¿Qué son?

Es una colección de elementos únicos, no pueden ser repetidos.

Declaración

```
const names = new Set(["John", "Harry", "Wick", "Jack", "Harry"]);
```

```
const randomWord = new Set("Hiiiiiii");
```

Si mostramos con `console.log(names);` y `console.log(randomWord);` este seria el resultado:

```
Set(4) { 'John', 'Harry', 'Wick', 'Jack' }
```

```
Set(2) { 'H', 'i' }
```

Propiedades

Size

La función `size` devuelve el numero de elementos de el conjunto.

```
Names.size;
```

```
4
```

Métodos

Add

Se utiliza para agregar al conjunto, si ya existe no lo agrega.

```
Names.add("juan");
```

Delete

Elimina un elemento del conjunto introducido por parámetro, no muestra mensaje de error aunque este no exista.

```
Names.delete("juan");
```

Has

Toma un valor por parámetro y devuelve un booleano en función de si lo encuentra.

```
Names.has("Juan");
```

```
True
```

Recorrer conjuntos

El método `forEach` se usa para iterar sobre el conjunto y obtener todos los elementos uno por uno.

```
names.forEach((element) => {  
    console.log(element);  
});
```

El método `entries` devuelve un iterador que contiene matrices de pares clave-valor en el orden de inserción. Aquí, la clave y el valor son los mismos.

```
const entries = names.entries();
```

```
console.log(entries.next().value);  
console.log(entries.next().value);  
console.log(entries.next().value);  
console.log(entries.next().value);  
console.log(entries.next().value);
```

El método `keys` devuelve un iterador de elementos de conjunto en el orden de inserción.

```
const keys = names.keys();
```

```
console.log(keys.next().value);  
console.log(keys.next().value);  
console.log(keys.next().value);  
console.log(keys.next().value);  
console.log(keys.next().value);
```

El método `values` devuelve un iterador de elementos del conjunto en el orden de inserción similar al método `keys`.

```
const values = names.values();
```

```
console.log(values.next().value);  
console.log(values.next().value);  
console.log(values.next().value);  
console.log(values.next().value);  
console.log(values.next().value);
```

Mapas

¿Qué son?

Es una colección de elementos, identificando cada valor con una clave.

Declaración

```
var map = new Map();
```

Propiedades

Size

La función size devuelve el numero de elementos de el mapa.

Métodos

Set

Añade un nuevo elemento.

```
map.set( 'one', 1 );
```

Get

Devuelve el valor de un elemento, se introduce la clave por parámetro.

```
map.get( 'one' );
```

Has

Toma un valor por parámetro y devuelve un booleano en función de si lo encuentra.

```
map.has('two')
```

Delete

Elimina un elemento del conjunto introduciendo por parámetro la clave.

```
map.delete('two')
```

Recorrer mapas

map.keys() devuelve un iterable para las claves.

```
for (let vegetable of recipeMap.keys()) {
```

```
  alert(vegetable);}
```

map.values() devuelve un iterable para los valores.

```
for (let Price of recipeMap.values()) {
```

```
  alert(price);}
```

map.entries() devuelve un iterable para las entradas [clave, valor]. Es el que usa por defecto en for..of.

```
for (let elementoof recipeMap.entries()) {
```

```
  alert(elemento);}
```

El método forEach se usa para iterar sobre el conjunto y obtener todos los elementos uno por uno.

```
recipeMap.forEach( (value, key, map) => {
```

```
  alert(`${key}: ${value}`);
```

```
});
```