JAVASCRIPT

MAP , SET, WEAK-SET, WEAK-MAP

Map

Map is a collection of keyed data items, just like an `Object`. But the main difference is that `Map` allows keys of any type.

Methods and properties are:

new Map() – creates the map.
map.set(key, value) – stores the value by the key.
map.get(key) – returns the value by the key, undefined if key doesn’t exist in map.
map.has(key) – returns true if the key exists, false otherwise.
map.delete(key) – removes the value by the key.
map.clear() – removes everything from the map.
map.size – returns the current element count.

Ex.

```
let map = new Map();

map.set('name', 'Rutvik');
map.set(surname, 'Patel');
map.set(age, 22); // a boolean key

map.get('name')  //  Rutvik
map.get('surname') // ‘patel’

map.size // 3
```

Set

A JavaScript Set is a collection of unique values.

Each value can only occur once in a Set.

A Set can hold any value of any data type.

Method Description

new Set() Creates a new Set

add() Adds a new element to the Set

delete() Removes an element from a Set

has() Returns true if a value exists

clear() Removes all elements from a Set

forEach() Invokes a callback for each element

values() Returns an Iterator with all the values in a Set

keys() Same as values()

entries() Returns an Iterator with the [value,value] pairs from a

Example:

```
const letters = new Set(["a","b","c"]);
```

Example:

```
letters.add(e);
letters.add(f);
letters.add(a); //error
```

We can use the Iterator object to access the elements.

Example:

```
const myIterator = letters.values();
for (const entry of myIterator) {
console.log(entry);
}
```

WeakMap:

-JavaScript WeakMap object is used to store the elements as key-value pair where keys are weakly referenced.

- WeakMap object can not contains the primitive type elements. It

- An object’s presence as a key in a WeakMap does not prevent the object from being garbage collected.

- Once an object used as a key has been collected, itscorresponding values in any WeakMap become

- candidates for garbage collection as well as long as they aren'tcan contains only object type elements.strongly referred to elsewhere.

Example:

```
let weakMap = new WeakMap();
let obj = {};
weakMap.set(obj, "ok");
```

WeakMap Methods:

- delete(): To delete the specified element from a WeakMap object.
- get(): To return the value of specified key.- has(): To indicate whether the Weakmap object contains the

specified value element.

- set(): To add or update the key-value pairs to WeakMap object.

WeakSet:

- JavaScript WeakSet object is used to store only the weakly held objects.
- WeakSet object can not contains the primitive type elements. It can contains only object type elements.

Example:

```
var ws = new WeakSet();
var foo = {};
var bar = {};
ws.add(foo);
ws.add(bar);
```

WeakSet Methods:

- add(): To add a new object to the end of WeakSet object.
- delete(): To delete the specified object from the WeakSet object.
- has(): To indicate whether the WeakSet object contains the

specified object element.