JavaScript Sets

[❮ Previous](https://www.w3schools.com/js/js_iterables.asp)[Next ❯](https://www.w3schools.com/js/js_maps.asp)

A JavaScript Set is a collection of unique values.

Each value can only occur once in a Set.

Essential Set Methods

|  |  |
| --- | --- |
| **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 in the Set |
| forEach() | Invokes a callback for each element in the Set |
| values() | Returns an iterator with all the values in a Set |
| **Property** | **Description** |
| size | Returns the number of elements in a Set |

How to Create a Set

You can create a JavaScript Set by:

* Passing an Array to new Set()
* Create a new Set and use add() to add values
* Create a new Set and use add() to add variables

The new Set() Method

Pass an Array to the new Set() constructor:

Example

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

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_create_array)

Create a Set and add values:

Example

// Create a Set  
const letters = new Set();  
  
// Add Values to the Set  
letters.add("a");  
letters.add("b");  
letters.add("c");

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_create_values)

Create a Set and add variables:

Example

// Create a Set  
const letters = new Set();  
  
// Create Variables  
const a = "a";  
const b = "b";  
const c = "c";  
  
// Add Variables to the Set  
letters.add(a);  
letters.add(b);  
letters.add(c);

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_create)

The add() Method

Example

letters.add("d");  
letters.add("e");

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_add)

If you add equal elements, only the first will be saved:

Example

letters.add("a");  
letters.add("b");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");  
letters.add("c");

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_add_equals)

ADVERTISEMENT

The forEach() Method

The forEach() method invokes (calls) a function for each Set element:

Example

// Create a Set  
const letters = new Set(["a","b","c"]);  
  
// List all Elements  
let text = "";  
letters.forEach (function(value) {  
  text += value;  
})

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_foreach)

The values() Method

The values() method returns a new iterator object containing all the values in a Set:

Example

letters.values()   // Returns [object Set Iterator]

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_values)

Now you can use the Iterator object to access the elements:

Example

// List all Elements  
let text = "";  
for (const x of letters.values()) {  
  text += x;  
}

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_set_iterator)

MAPS

# JavaScript Maps

[❮ Previous](https://www.w3schools.com/js/js_sets.asp)[Next ❯](https://www.w3schools.com/js/js_typeof.asp)

A Map holds key-value pairs where the keys can be any datatype.

A Map remembers the original insertion order of the keys.

## Essential Map Methods

|  |  |
| --- | --- |
| **Method** | **Description** |
| new Map() | Creates a new Map |
| set() | Sets the value for a key in a Map |
| get() | Gets the value for a key in a Map |
| delete() | Removes a Map element specified by the key |
| has() | Returns true if a key exists in a Map |
| forEach() | Calls a function for each key/value pair in a Map |
| entries() | Returns an iterator with the [key, value] pairs in a Map |
| **Property** | **Description** |
| size | Returns the number of elements in a Map |

## How to Create a Map

You can create a JavaScript Map by:

* Passing an Array to new Map()
* Create a Map and use Map.set()

## The new Map() Method

You can create a Map by passing an Array to the new Map() constructor:

### Example

// Create a Map  
const fruits = new Map([  
  ["apples", 500],  
  ["bananas", 300],  
  ["oranges", 200]  
]);

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_create_array)

## The set() Method

You can add elements to a Map with the set() method:

### Example

// Create a Map  
const fruits = new Map();  
  
// Set Map Values  
fruits.set("apples", 500);  
fruits.set("bananas", 300);  
fruits.set("oranges", 200);

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_set_new)

The set() method can also be used to change existing Map values:

### Example

fruits.set("apples", 200);

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_set)

## The get() Method

The get() method gets the value of a key in a Map:

### Example

fruits.get("apples");    // Returns 500

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_get)

ADVERTISEMENT

## The size Property

The size property returns the number of elements in a Map:

### Example

fruits.size;

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_size)

## The delete() Method

The delete() method removes a Map element:

### Example

fruits.delete("apples");

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_delete)

## The has() Method

The has() method returns true if a key exists in a Map:

### Example

fruits.has("apples");

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_has)

### Try This:

fruits.delete("apples");  
fruits.has("apples");

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_has_deleted)

## JavaScript Objects vs Maps

#### **Differences between JavaScript Objects and Maps:**

|  |  |  |
| --- | --- | --- |
|  | **Object** | **Map** |
| **Iterable** | Not directly iterable | Directly iterable |
| **Size** | Do not have a size property | Have a size property |
| **Key Types** | Keys must be Strings (or Symbols) | Keys can be any datatype |
| **Key Order** | Keys are not well ordered | Keys are ordered by insertion |
| **Defaults** | Have default keys | Do not have default keys |

## The forEach() Method

The forEach() method calls a function for each key/value pair in a Map:

### Example

// List all entries  
let text = "";  
fruits.forEach (function(value, key) {  
  text += key + ' = ' + value;  
})

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_foreach)

## The entries() Method

The entries() method returns an iterator object with the [key, values] in a Map:

### Example

// List all entries  
let text = "";  
for (const x of fruits.entries()) {  
  text += x;  
}

[Try it Yourself »](https://www.w3schools.com/js/tryit.asp?filename=tryjs_map_entries)