# JavaScript Set Reference

Back to JS page

Table of Contents


- JavaScript Set Reference
  - Set Properties
    - Example 1
  - Set Add/Delete Methods
    - Example 2
  - Set Iteration Methods
    - Example 3
  - Set Logic Methods
    - Example 4
  - Document
  - Reference


## Set Properties

The complete `Set` object reference:

| Property | Description |
| --- | --- |
| `size` | Number of elements in the Set |
| `constructor` | The Set constructor function |

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Set Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Set Properties</h4>
<p id="demo"></p>

<script>
const letters = new Set(["a", "b", "c", "d"]);

document.getElementById("demo").innerHTML =
  "Set: " + [...letters] + "<br>" +
  "size: " + letters.size + "<br>" +
  "constructor: " + letters.constructor;
</script>

</body>
</html>
```

### Example 1

**Result** View Example

## Set Add/Delete Methods

Methods for adding and removing elements:

| Method | Description |
|---|---|
| `add(value)` | Add a new element |
| `delete(value)` | Remove an element |
| `has(value)` | Check if element exists |
| `clear()` | Remove all elements |

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Set Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Set Add/Delete Methods</h4>
<p id="demo"></p>

<script>
const letters = new Set(["a", "b"]);

let text = "Start: " + [...letters] + "<br><br>";

letters.add("c");
text += "add('c'): " + [...letters] + "<br>";
text += "has('c'): " + letters.has("c") + "<br><br>";

letters.delete("a");
text += "delete('a'): " + [...letters] + "<br><br>";

letters.clear();
text += "clear(): " + [...letters] + " (empty)<br>";
text += "size: " + letters.size;

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

### Example 2

**Result** View Example

## Set Iteration Methods

Methods for iterating over Set elements:

| Method | Description |
| --- | --- |
| `forEach(fn)` | Call function for each element |
| `values()` | Iterator of all values |
| `keys()` | Same as values() for Set |
| `entries()` | Iterator of [value, value] pairs |

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Set Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Set Iteration Methods</h4>
<p id="demo"></p>

<script>
const letters = new Set(["a", "b", "c"]);

let text = "<b>forEach():</b> ";
letters.forEach(v => text += v + " ");
text += "<br>";

text += "<b>values():</b> ";
for (let v of letters.values()) text += v + " ";
text += "<br>";

text += "<b>keys():</b> ";
for (let k of letters.keys()) text += k + " ";
text += "<br>";

text += "<b>entries():</b> ";
for (let e of letters.entries()) text += "[" + e[0] + "," + e[1] + "] ";

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

## Example 3

**Result** View Example

## Set Logic Methods

Set logic methods (newer Set methods):

| Method | Description |
| --- | --- |
| `union(set)` | All elements from both Sets |
| `intersection(set)` | Elements in both Sets |
| `difference(set)` | Elements in this but not other |
| `symmetricDifference(set)` | Elements in either, not both |
| `isSubsetOf(set)` | Every element in this also in other |
| `isSupersetOf(set)` | Every element in other also in this |
| `isDisjointFrom(set)` | No common elements |

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Set Reference</h2>
<p>From w3schools.com, Experiment by Teeratus_R</p>

<h4>Set Logic Methods</h4>
<p id="demo"></p>

<script>
const setA = new Set(["a", "b", "c"]);
const setB = new Set(["c", "d"]);

let text = "Set A: " + [...setA] + "<br>";
text += "Set B: " + [...setB] + "<br><br>";

if (setA.union) {
  text += "A.union(B): " + [...setA.union(setB)] + "<br>";
  text += "A.intersection(B): " + [...setA.intersection(setB)] + "<br>";
  text += "A.difference(B): " + [...setA.difference(setB)] + "<br>";
  text += "A.symmetricDifference(B): " + [...setA.symmetricDifference(setB)] + "<br>";
  text += "A.isSubsetOf(B): " + setA.isSubsetOf(setB) + "<br>";
  text += "A.isDisjointFrom(new Set(['x'])): " + setA.isDisjointFrom(new Set(["x"]));
} else {
  text += "Logic methods are not supported in this browser yet.";
}

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

## Example 4

**Result** View Example

# Document

Document in project

You can Download PDF file.

## Reference

- W3Schools JavaScript Set Reference