How to Create a Set

You can create a JavaScript Set by:

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

Example 1

Pass an Array to the new Set() constructor:

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

Example 2

Create a Set and add values:

```
// Create a Set
const letters = new Set();

// Add Values to the Set
letters.add("a");
letters.add("b");
letters.add("c");
```

Listing Set Elements

You can list all Set elements (values) with a **for..of** loop:

Example

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

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

The add() Method

Example

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

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");
```

Description

The add() method inserts a new element in the set.

Syntax

set.add(*value*)

Parameters

Parameter	Description
value	Required. The value to add.

Return Value

Type	Description
Set	A set object with the added value.

JavaScript Set clear()

Example

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

// Clear Set
letters.clear()
```

Description

The clear() method removes all values from a set.

Syntax

set.clear()

Parameters

NONE

Return Value

NONE

JavaScript Set delete()

Example

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

// Remove one Element
letters.delete("a");
```

Description

The delete() method removes a specified value from a set.

Syntax

set.delete(*value*)

Parameters

Parameter	Description
value	Required. The value to remove.

Return Value

Type	Description
Boolean	true if the value existed, otherwise false.

JavaScript Set entries()

Example 1

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

// Get all Entries
const myIterator = letters.entries();

// List all Entries
let text = "";
for (const entry of myIterator) {
  text += entry;
}
```

More Examples Below !

Description

The entries() method returns an Iterator with [value,value] pairs from a set.

Note

The entries() method is supposed to return a [key,value] pair from an object.

Since a set has no keys, the entries() method returns [value,value].

This makes Sets compatible with Maps.

Syntax

set.entries()

Parameters

NONE

Return Value

Type	Description
Iterator	An iterable object with the values of the set.

JavaScript Set forEach()

Example

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

// List all entries
let text = "";
letters.forEach (function(value) {
  text += value;
})
```

Description

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

The forEach() method does not change the original set.

Syntax

set.forEach(*callback*)

Parameters

Parameter	Description
callback	Required. A function to execute for each element.

Return Value

NONE

JavaScript Set has()

The has() Method

Example

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

// Does the Set contain "d"?
answer = letters.has("d");
```

Description

The has() method returns true if a specified value exists in a set.

Syntax

set.has(*value*)

Parameters

Parameter	Description
value	Required. The value to test for.

Return Value

Type	Description
Boolean	true if the value exists, otherwise false.

JavaScript Set keys()

Example

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

```
// Get the Values
const myIterator = letters.keys();

// List the Values
let text = "";
for (const x of myIterator) {
  text += x;
}
```

More Examples Below !

Description

The keys() method returns an Iterator object with the values in a set.

The keys() method does not change the original set.

Note

Since a set has no keys, the keys() method returns the same as values().

This makes JavaScript sets compatible with JavaScript maps.

Syntax

set.keys()

Parameters

NONE

Return Value

Type	Description
Iterator	An iterable object with the values of the set.

JavaScript Set values()

Example

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

```
// Get all Values
const myIterator = letters.values();

// List all Values
let text = "";
for (const entry of myIterator) {
  text += entry;
}
```

More Examples Below !

Description

The values() method returns an Iterator object with the values in a set.

The values() method does not change the original set.

Syntax

set.values()

Parameters

NONE

Return Value

Type	Description
Iterator	An iterable object with the values of the set.