

SET DATA TYPE

JavaScript has a built-in Set data type, introduced in ECMAScript 2015 (ES6). A Set is a collection of unique values, meaning that it automatically removes duplicates.

Here's a quick overview of how to use the Set data type in JavaScript:

Creating a Set

You can create a new Set instance by using the Set constructor.

```
let mySet = new Set();
```

You can also initialise a Set with an array of values.

```
let mySet = new Set([1, 2, 3, 4, 5]);
```

Adding Values

You can add values to a Set using the add method.

```
mySet.add(6);  
mySet.add(7);
```

Checking for Values

You can check if a value exists in a Set using the has method.

```
console.log(mySet.has(3)); // true  
console.log(mySet.has(10)); // false
```

Removing Values

You can remove values from a Set using the delete method.

```
mySet.delete(4);  
console.log(mySet.has(4)); // false
```

Size of the Set

You can get the number of values in a Set using the size property.

```
console.log(mySet.size); // 6
```

Iterating Over a Set

You can iterate over the values in a Set using a for...of loop or the forEach method.

```
for (let value of mySet) {  
  console.log(value);  
}
```

Clearing a Set

You can remove all values from a Set using the clear method.

```
mySet.clear();  
console.log(mySet.size); // 0
```

The Set data type in JavaScript is useful for managing collections of unique values and performing operations that require uniqueness, such as removing duplicates from arrays or managing a collection of unique items.