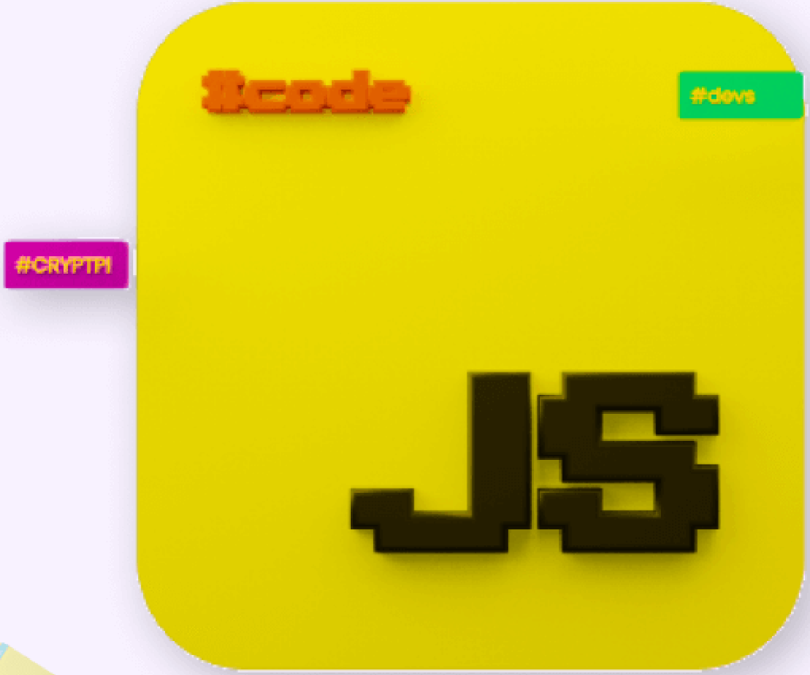
JavaScript

Sets



Next →

Hey Everyones 👋

In this post, you will learn about JavaScript Sets with the help of examples.

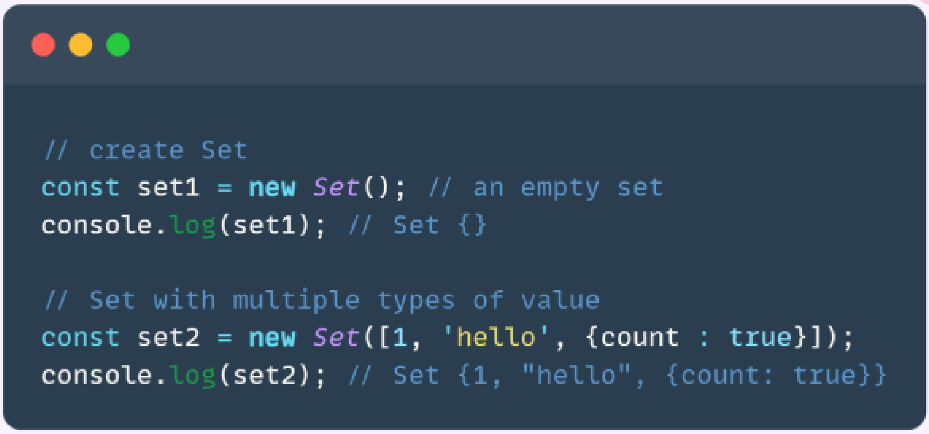
The JavaScript ES6 has introduced two new data structures, i.e Set and WeakSet.

Do Like, save and Share This Post If You Found This Helpful.

NEXT →

Create Set

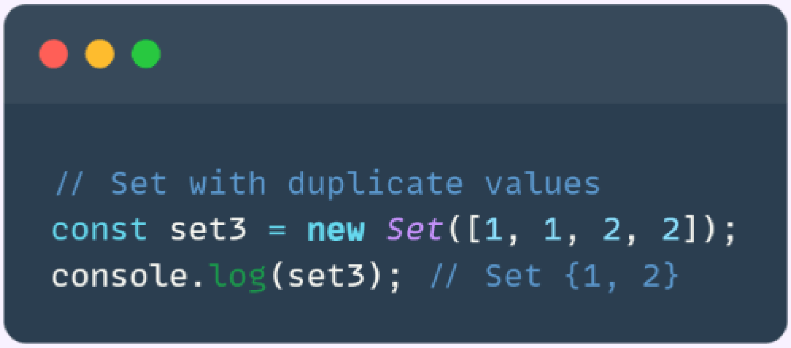
- To create a Set, you need to use new Set() constructor.



```
// create Set
const set1 = new Set(); // an empty set
console.log(set1); // Set {}

// Set with multiple types of value
const set2 = new Set([1, 'hello', {count : true}]);
console.log(set2); // Set {1, "hello", {count: true}}
```

- When duplicate values are passed to a Set object, the duplicate values are excluded.

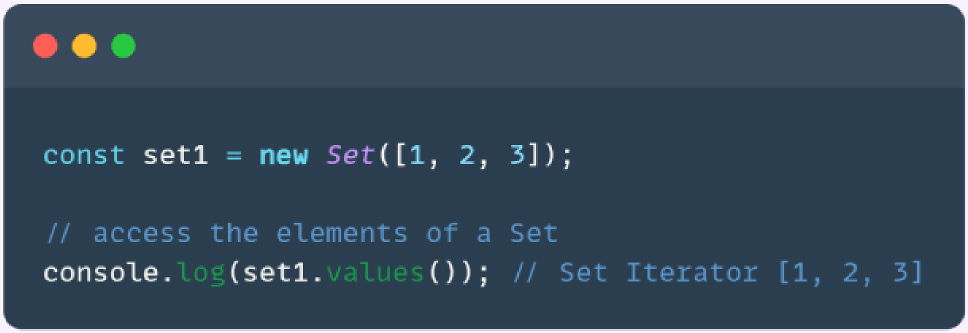


```
// Set with duplicate values
const set3 = new Set([1, 1, 2, 2]);
console.log(set3); // Set {1, 2}
```

Next →

Access Elements

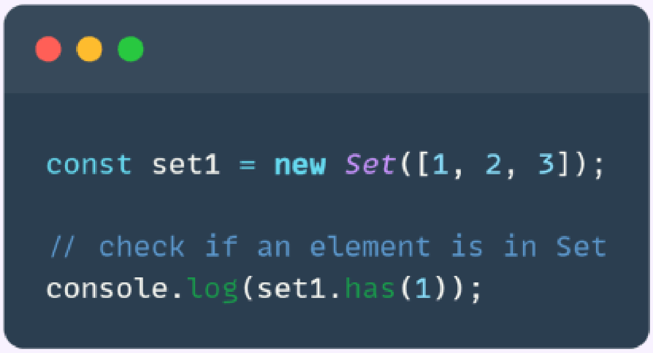
- You can access Set elements using the values() method and check if there is an element inside Set using has() method.



```
const set1 = new Set([1, 2, 3]);

// access the elements of a Set
console.log(set1.values()); // Set Iterator [1, 2, 3]
```

- You can use the has() method to check if the element is in a Set.



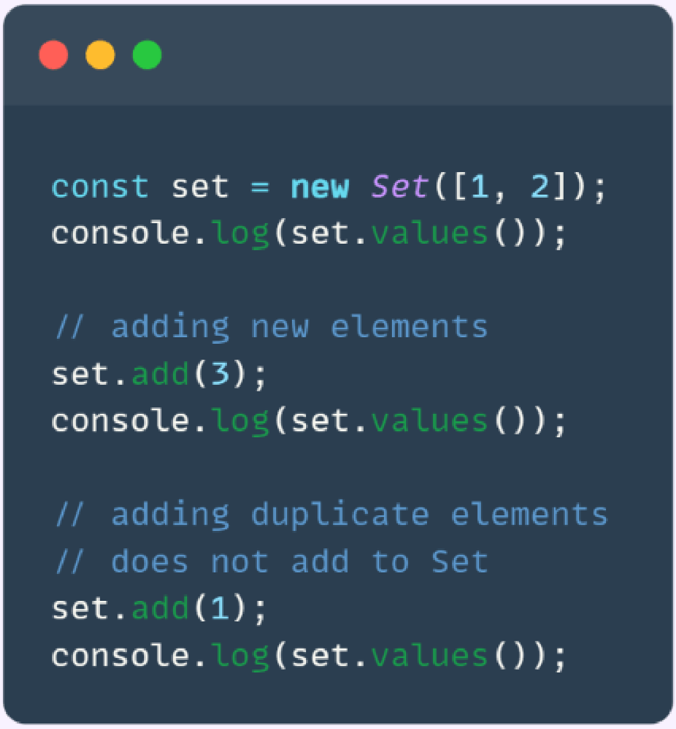
```
const set1 = new Set([1, 2, 3]);

// check if an element is in Set
console.log(set1.has(1));
```

NEXT →

Adding New Elements

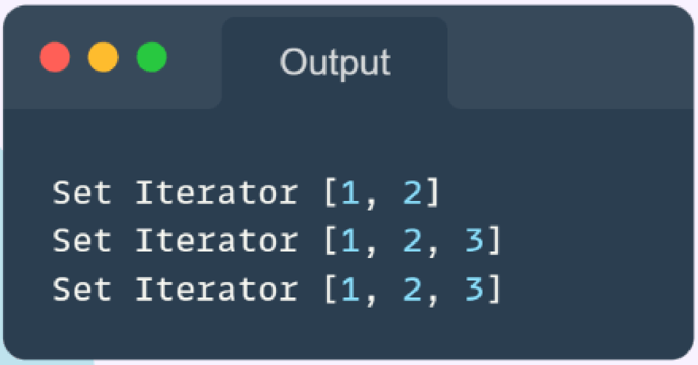
- You can add elements to a Set using the add() method.



```
const set = new Set([1, 2]);
console.log(set.values());

// adding new elements
set.add(3);
console.log(set.values());

// adding duplicate elements
// does not add to Set
set.add(1);
console.log(set.values());
```

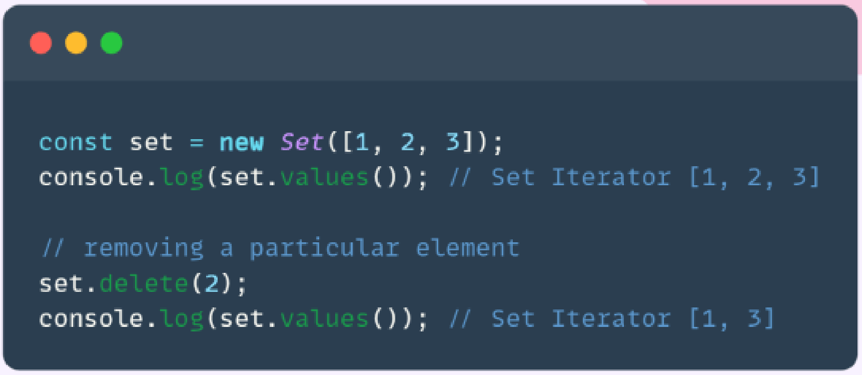


```
Set Iterator [1, 2]
Set Iterator [1, 2, 3]
Set Iterator [1, 2, 3]
```

Next →

Removing Elements

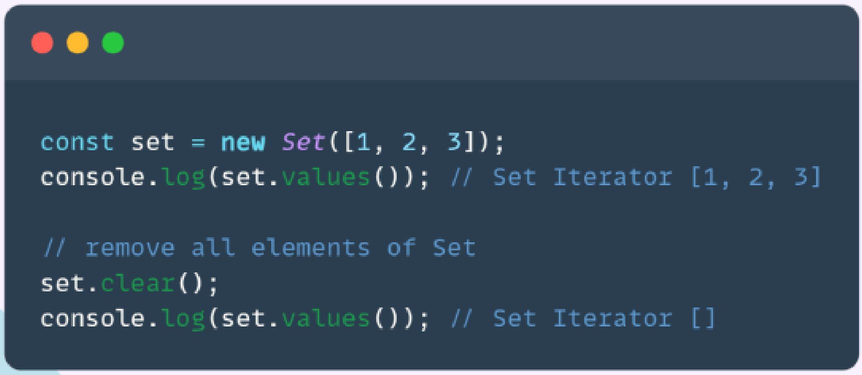
- The delete() method removes a specific element from a Set.



```
const set = new Set([1, 2, 3]);
console.log(set.values()); // Set Iterator [1, 2, 3]

// removing a particular element
set.delete(2);
console.log(set.values()); // Set Iterator [1, 3]
```

- The clear() method removes all elements from a Set.



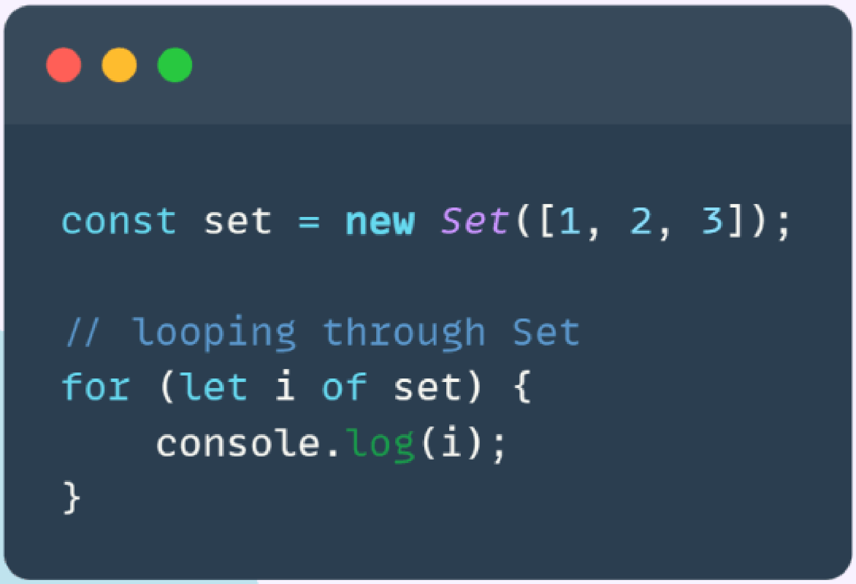
```
const set = new Set([1, 2, 3]);
console.log(set.values()); // Set Iterator [1, 2, 3]

// remove all elements of Set
set.clear();
console.log(set.values()); // Set Iterator []
```

NEXT →

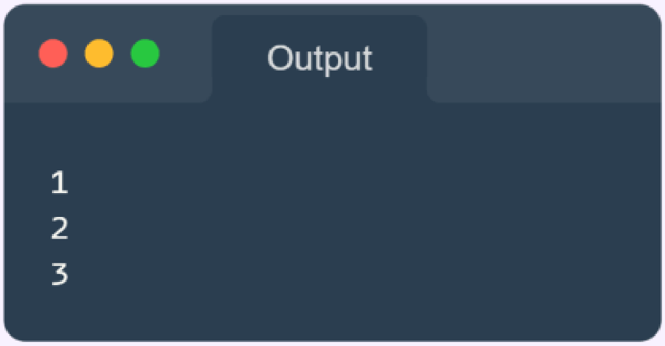
Iterate Sets

- You can iterate through the Set elements using the for...of loop or forEach() method.
- The elements are accessed in the insertion order.



```
const set = new Set([1, 2, 3]);

// looping through Set
for (let i of set) {
    console.log(i);
}
```



```
1
2
3
```

NEXT →

Chech Out Other Javascript Post Also 😊

0

@CODE.CLASH

JavaScript

Arrow

Function

Next →

0

@CODE.CLASH

JavaScript

Exception - Handling

try_catch,finally,throw Statement

Next →

Next →

Best Of Luck :)

Next →

THANKS FOR YOUR ATTENTION

Code.Clash

Imtiyaz Nandasaniya

LIKE AND SAVE IT

FOR LATER