

JavaScript Map and Set

Map

- A collection of key-value pairs where keys can be of any type.
- Maintains the order of elements.
- Map does not allow duplicate keys

Creating a Map

```
let map = new Map();
map.set('name', 'Alice');
map.set(1, 'one');
console.log(map.get('name')); // Alice
```

Map Methods

- `set(key, value)` : Adds a key-value pair.
- `get(key)` : Retrieves value by key.
- `delete(key)` : Removes a key.
- `has(key)` : Checks if key exists.
- `size` : Returns the number of elements.
- `clear()` : Removes all elements.

Iterating Over a Map

```
map.forEach((value, key) => console.log(key, value));
```

```
for (let key of map.keys()) {
  console.log("Key:", key);
}
```

```
for (let value of map.values()) {
  console.log("Value:", value);
}
```

```
for (let [key, value] of map.entries()) {
  console.log(`Key: ${key}, Value: ${value}`);
}
```

Set

- A collection of unique values.
- Does not allow duplicate elements.

Creating a Set

```
let set = new Set([1, 2, 3, 3]);  
console.log(set); // Set { 1, 2, 3 }
```

Set Methods

- `add(value)` : Adds a value.
- `delete(value)` : Removes a value.
- `has(value)` : Checks if a value exists.
- `size` : Returns the number of elements.
- `clear()` : Removes all elements.

Iterating Over a Set

```
set.forEach(value => console.log(value));  
for (let value of set) console.log(value);
```

Key Differences

Feature	Map	Set
Stores Key-Value Pairs?	Yes	No
Maintains Order?	Yes	Yes
Allows Duplicates?	No (unique keys)	No

Use Map when you need key-value pairs.

Use Set when you need unique values.