DATA STRUCTURES: SET

Andrew Sheehan

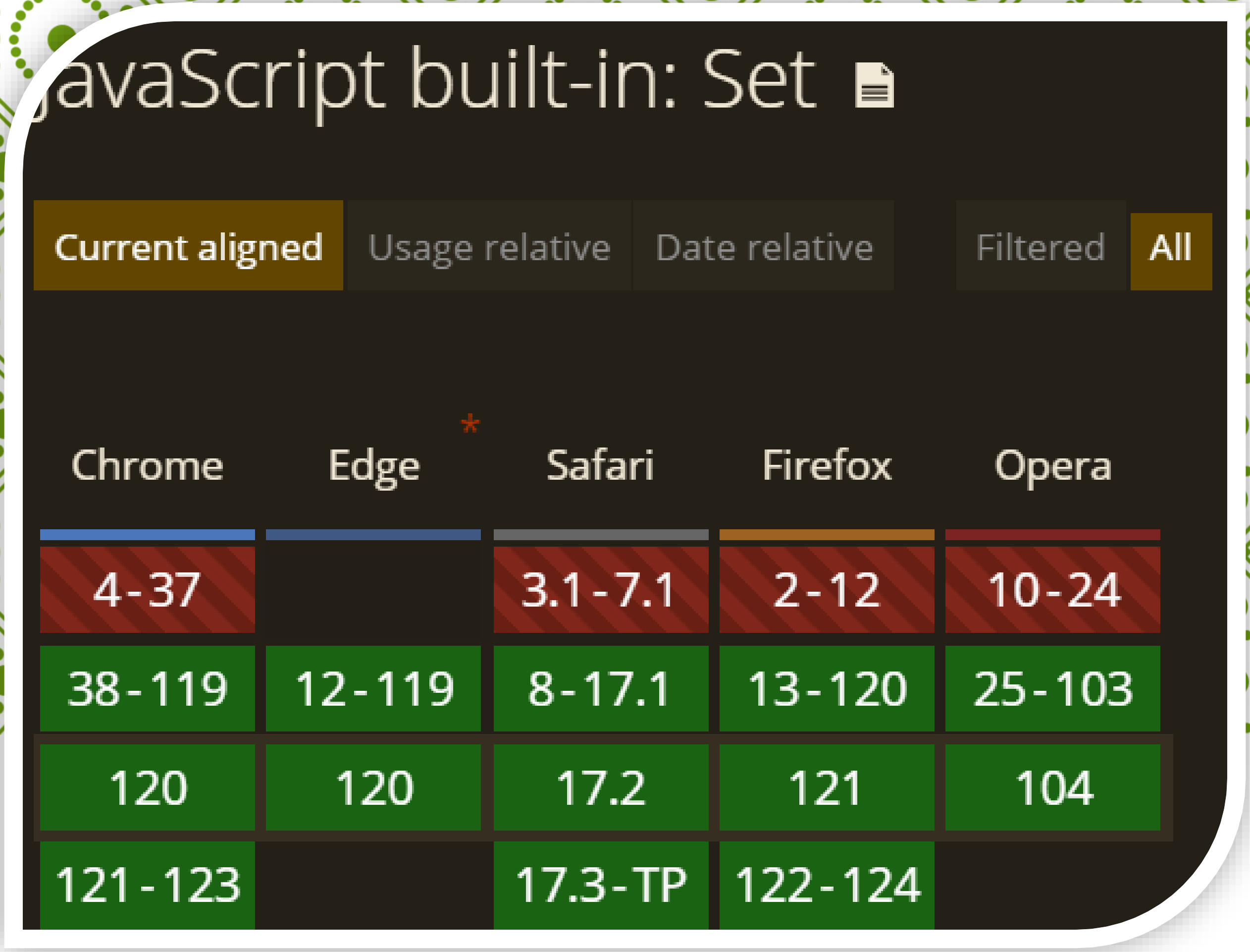
Boston University
Computer Science/MET

The Set object lets you store unique values, whether primitive values or objects.

NAN AND UNDEFINED ALSO CAN PLACED INTO A SET

12/ “Nice Dog!” // NaN

let whatAmI; // undefined



JavaScript built-in: Set

Chrome	Edge*	Safari	Firefox	Opera
4-37		3.1-7.1	2-12	10-24
38-119	12-119	8-17.1	13-120	25-103
120	120	17.2	121	104
121-123		17.3-TP	122-124	

CANIUSE.COM

(AS OF JAN/2024)

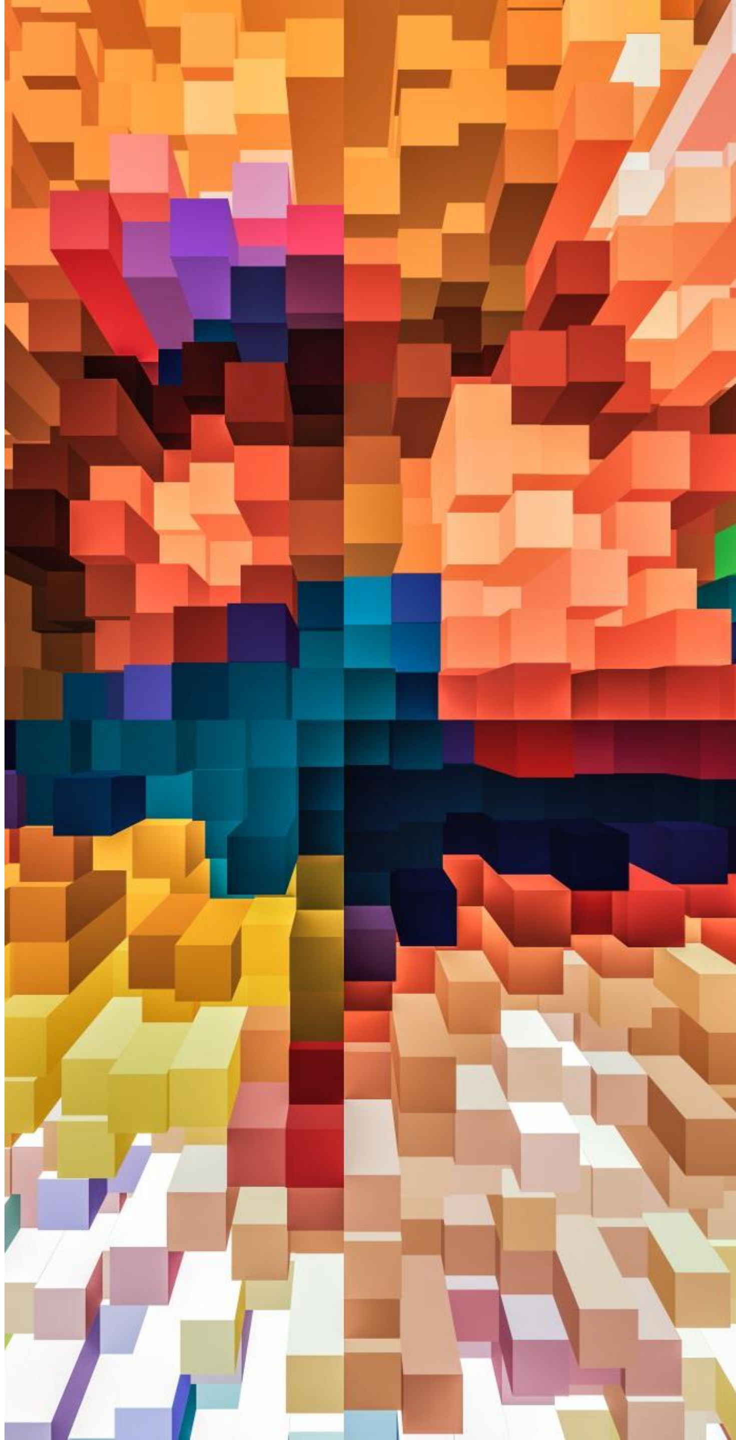
DUPLICATES? *NOT ALLOWED*

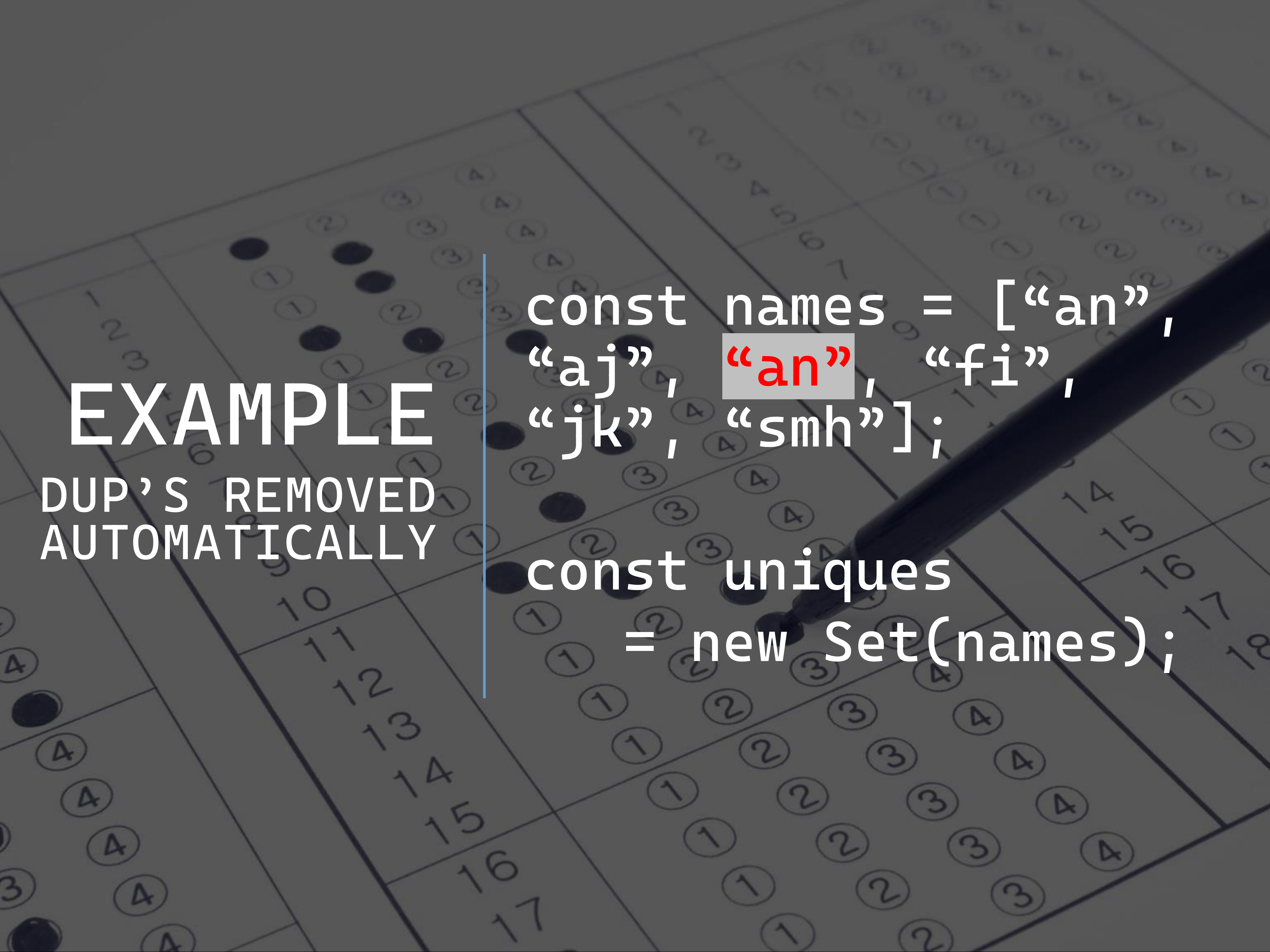
Only unique values are allowed

INSERTIONS WILL MAINTAIN IN THE ORDER YOU STARTED WITH

CREATING INSTANCES

```
const ages = new Set();
```





EXAMPLE

DUP’S REMOVED AUTOMATICALLY

```
const names = [“an”,
“aj”, “an”, “fi”,
“jk”, “smh”];

const uniques
    = new Set(names);
```

Using a string? All duplicate characters will be removed

```
const state = “Massachusetts”;
const uniqueChars = new Set(state);
// Maschuet
```

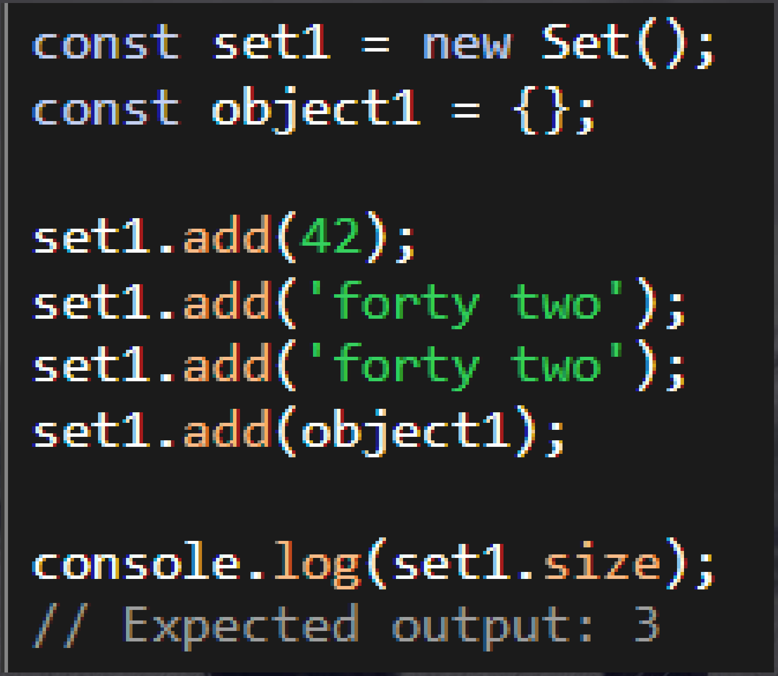
```
const vals = new Set();

// one expression
vals.add(5);

// chained expressions
vals.add(7).add(5).add(5);
```

ARRAYS

```
const salts =
[‘38x’, ‘2x1x0’];

let saltSet = new
Set(salts);

saltSet.has(‘2x0x’)
// false
// true: deleted;
// false - did not
let names = [‘joe’, ‘Dan’];
let nameSet = new Set(names);

let result =
names.delete(‘joe’
);
```

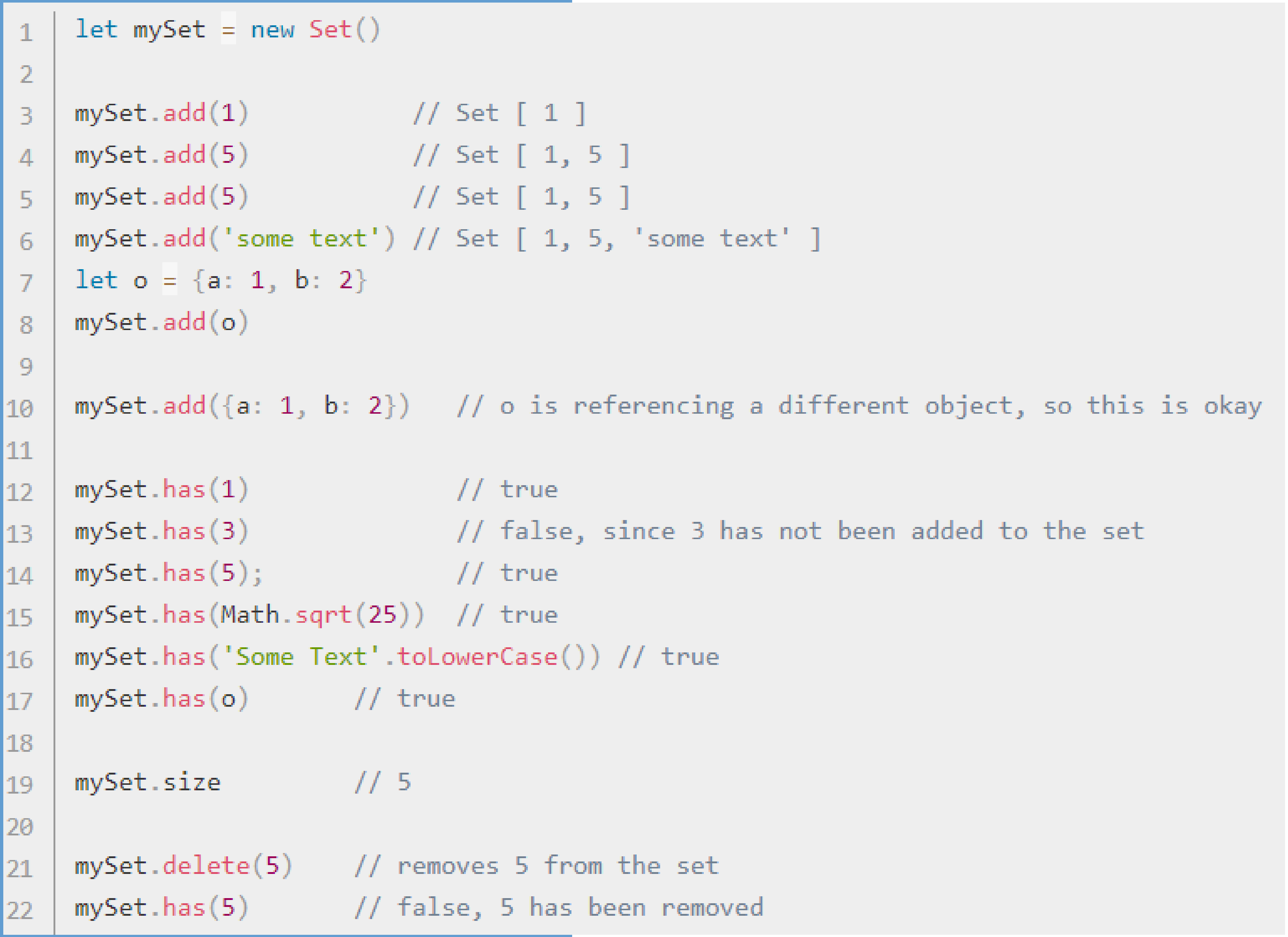
```
// All key/values are deleted
values.clear();
```

if (values.size > 0)



```
const set1 = new Set();
const object1 = {};

set1.add(42);
set1.add('forty two');
set1.add('forty two');
set1.add(object1);

console.log(set1.size);
// Expected output: 3
```

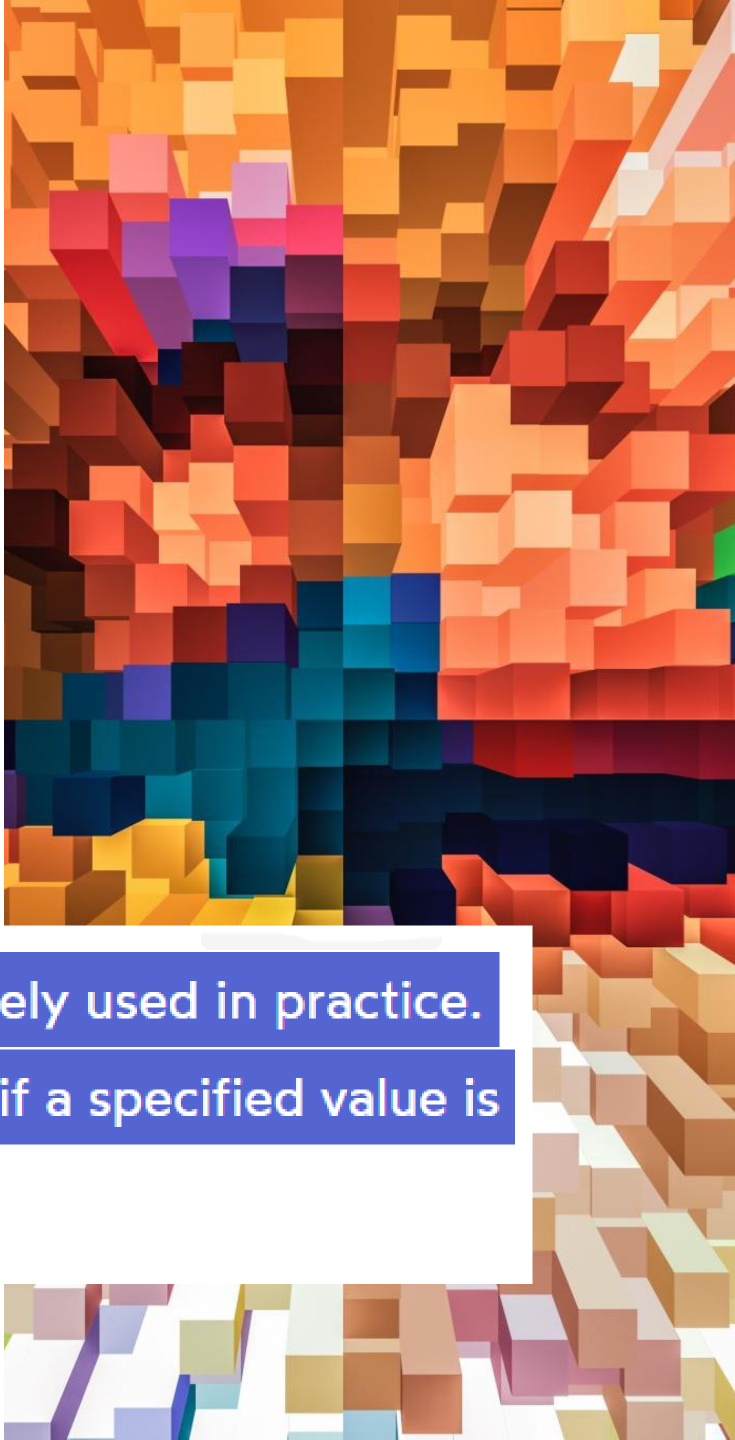
```
ids.values().forEach(id => {
    // logic goes here.
});
```

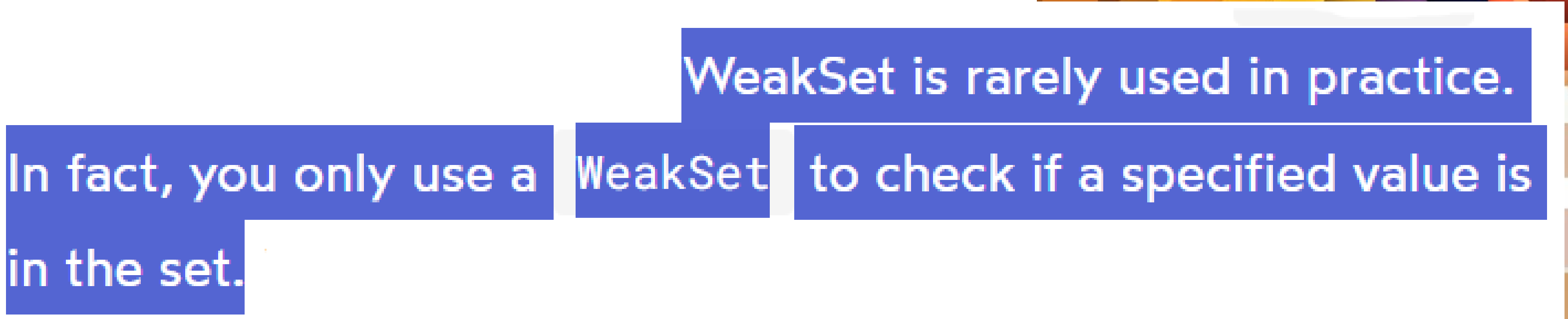


```javascript
let mySet = new Set()

mySet.add(1)           // Set [ 1 ]
mySet.add(5)           // Set [ 1, 5 ]
mySet.add(5)           // Set [ 1, 5 ]
mySet.add('some text') // Set [ 1, 5, 'some text' ]
let o = {a: 1, b: 2}
mySet.add(o)

mySet.add({a: 1, b: 2})   // o is referencing a different object, so this is okay

mySet.has(1)              // true
mySet.has(3)              // false, since 3 has not been added to the set
mySet.has(5);             // true
mySet.has(Math.sqrt(25))  // true
mySet.has('Some Text'.toLowerCase()) // true
mySet.has(o)       // true

mySet.size         // 5

mySet.delete(5)    // removes 5 from the set
mySet.has(5)       // false, 5 has been removed
```

WEAKSET
VARIANT OF SET

- Only objects.
- Cannot iterate over it
- .size – cannot use




WeakSet is rarely used in practice.
In fact, you only use a `WeakSet` to check if a specified value is in the set.