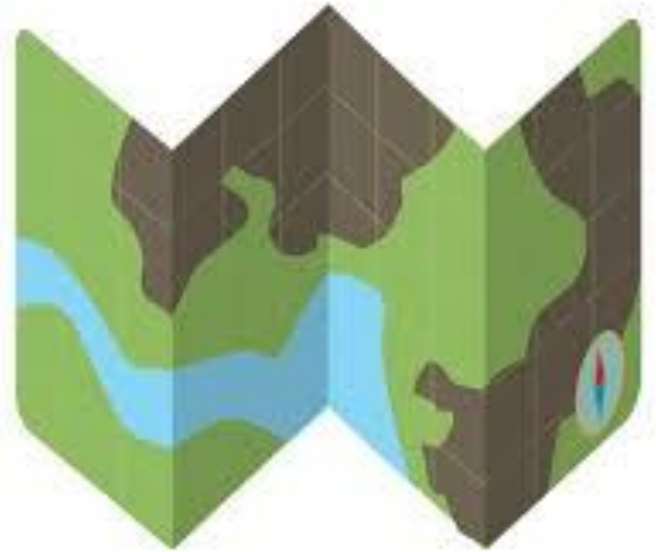
JAVASCRIPT SET

Andrew Sheehan

Boston University
Metropolitan College

WHAT IS
A SET?

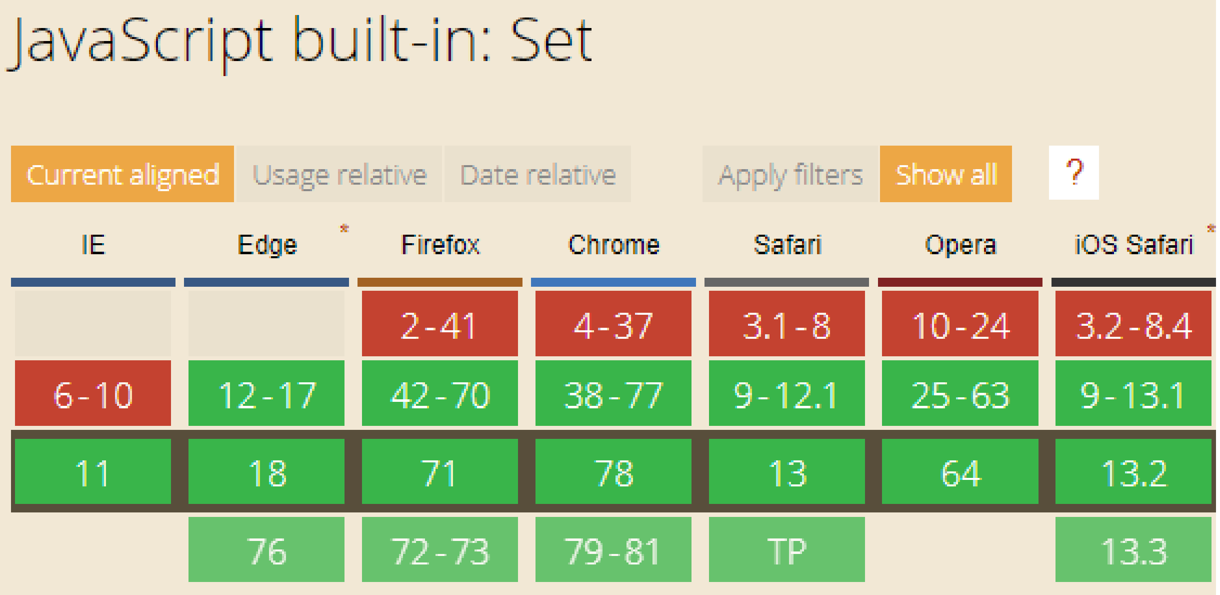


A Set is a built-in data type that stores unique values.

TAKE NOTE
ON THESE TYPES

NaN and undefined can also be stored (placed) in your Set.

CANIUSE?
AS OF 2020



JavaScript built-in: Set

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *
		2-41	4-37	3.1-8	10-24	3.2-8.4
6-10	12-17	42-70	38-77	9-12.1	25-63	9-13.1
11	18	71	78	13	64	13.2
	76	72-73	79-81	TP		13.3

DUPLICATES?
NOT ALLOWED

Only **unique values** are allowed.

ORDERED

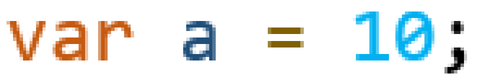
INSERTION ORDER

Insertions will be done in order – as you do it.

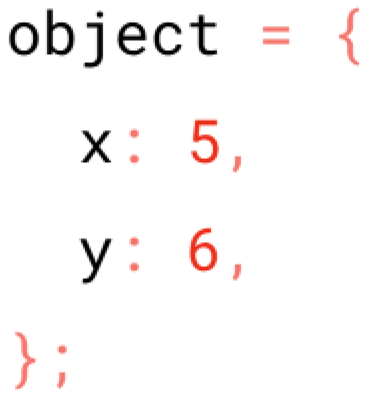
SETS
USE ANYTHING

A Set can store any type:

primitives or objects.



```
var a = 10;
```



```
object = {
  x: 5,
  y: 6,
};
```

USING
INSTANCES

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

USING

EXAMPLE 2

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

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

USING

EXAMPLE (SET OF CHARACTERS)

```
                // any dup char’s removed
const name = “Massachusetts”;

const uniqueChars
                = new Set(name);
```

SETS

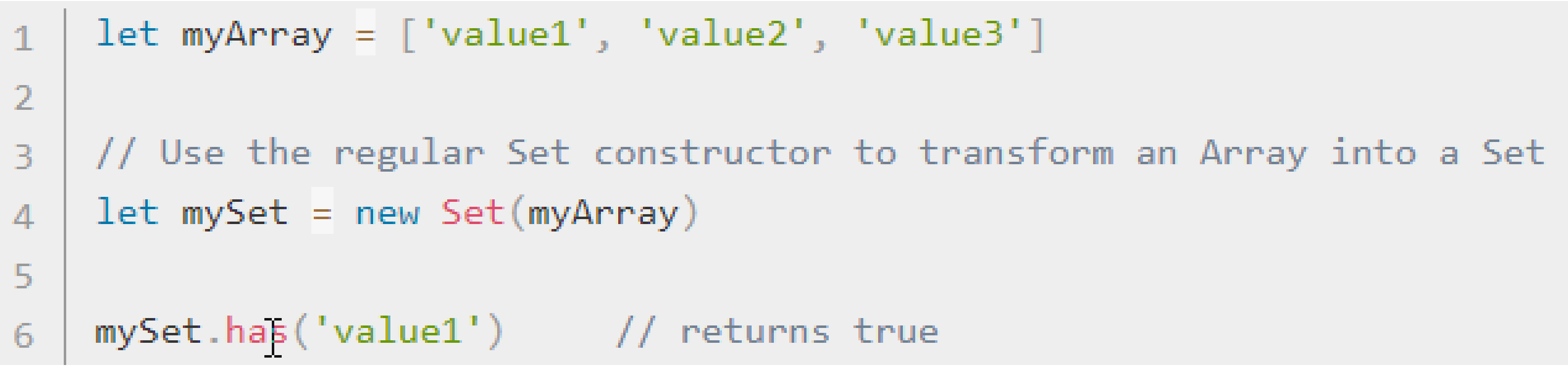
SET.ADD(VALUE)

```
const numbers = new Set();

// one expression
names. add(5);

// chained expressions
names.add(67).add(15).add(405);
```

SETS
USING ARRAYS



```
let myArray = ['value1', 'value2', 'value3']

// Use the regular Set constructor to transform an Array into a Set
let mySet = new Set(myArray)

mySet.has('value1')     // returns true
```

USING

DOES THE VALUE EXIST?

```
                           // key
if ( IdToName.has(“U-13-32”)  ) {
    /* true: Yes, it is within the Set. */
}
```

SETS

REMOVING A VALUE

```
// true:  deleted;  false did not
names.delete(‘Andrew’);
```

CLEAR()

TO REMOVE EVERYTHING

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

USING MAPS

RETRIEVING THE SIZE

```
If ( IdToName.size > 0) {
    // logic goes here
}
```

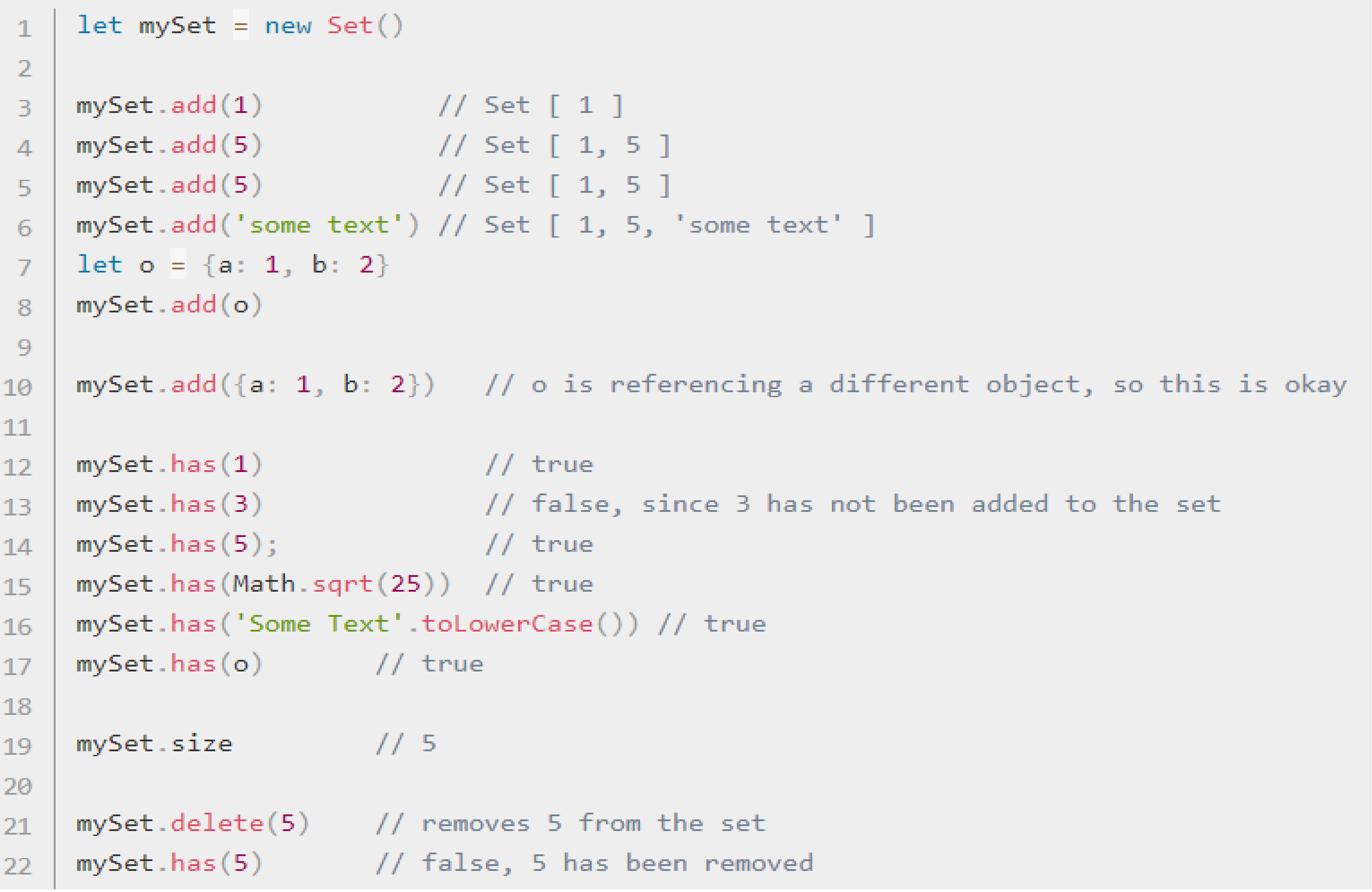
USING SET

ONLY THE VALUES

```
IdToName
    .values()
        .forEach(value => {
                // logic goes here.
        });
```

USING SET

EXAMPLE



```
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
```