30 Days Of JavaScript: Sets and Maps

LinkedIn Follow @asabeneh

Author: Asabeneh Yetayeh

January, 2020

<< Day 9 | Day 11>>

Day 10

Day 10

Set

Set is a collection of elements. Set can only contains unique elements. Let us see how to create a set in the section below.

Creating an empty set

```
const companies = new Set()
console.log(companies)
```

```
Set(0) {}
```

Creating set from array

```
const languages = [
  'English',
  'Finnish',
  'English',
  'French',
  'Spanish',
  'English',
  'French',
]

const setOfLanguages = new Set(languages)
console.log(setOfLanguages)
```

```
Set(4) {"English", "Finnish", "French", "Spanish"}
```

Set is an iterable object and we can iterate through each elements.

```
const languages = [
  'English',
  'Finnish',
  'English',
  'French',
  'Spanish',
  'English',
  'French',
]

const setOfLanguages = new Set(languages)

for (const language of setOfLanguages) {
  console.log(language)
}
```

```
  English
  Finnish
  French
  Spanish
```

Adding an element to a set

```js
const companies = new Set() // creating an empty set
console.log(companies.size) // 0

companies.add('Google') // add element to the set
companies.add('Facebook')
companies.add('Amazon')
companies.add('Oracle')
companies.add('Microsoft')
console.log(companies.size) // 5 elements in the set
console.log(companies)
```

```
Set(5) {"Google", "Facebook", "Amazon", "Oracle", "Microsoft"}
```

We can also use loop to add element to a set.

```js
const companies = ['Google', 'Facebook', 'Amazon', 'Oracle', 'Microsoft']
setOfCompanies = new Set()
for (const company of companies) {
  setOfCompanies.add(company)
}
```

```
Set(5) {"Google", "Facebook", "Amazon", "Oracle", "Microsoft"}
```

Deleting an element a set

We can delete an element from a set using a delete method.

```js
console.log(companies.delete('Google'))
console.log(companies.size) // 4 elements left in the set
```

Checking an element in the set

The has method can help to know if a certain element exists in a set.

```js
console.log(companies.has('Apple')) // false
console.log(companies.has('Facebook')) // true
```

Clearing the set

It removes all the elements from a set.

```
companies.clear()
console.log(companies)
```

```
Set(0) {}
```

See the example below to learn how to use set.

```
const languages = [
  'English',
  'Finnish',
  'English',
  'French',
  'Spanish',
  'English',
  'French',
]
const langSet = new Set(languages)
console.log(langSet) // Set(4) {"English", "Finnish", "French", "Spanish"}
console.log(langSet.size) // 4

const counts = []
const count = {}

for (const l of langSet) {
  const filteredLang = languages.filter((lng) => lng === l)
  console.log(filteredLang) // ["English", "English", "English"]
  counts.push({ lang: l, count: filteredLang.length })
}
console.log(counts)
```

```
[
  { lang: 'English', count: 3 },
  { lang: 'Finnish', count: 1 },
  { lang: 'French', count: 2 },
  { lang: 'Spanish', count: 1 },
]
```

Other use case of set. For instance to count unique item in an array.

```
const numbers = [5, 3, 2, 5, 5, 9, 4, 5]
const setOfNumbers = new Set(numbers)
```

```
console.log(setOfNumbers)
```

```
Set(5) {5, 3, 2, 9, 4}
```

Union of sets

To find a union to two sets can be achieved using spread operator. Lets find the union of set A and set B (A U B)

```
let a = [1, 2, 3, 4, 5]
let b = [3, 4, 5, 6]
let c = [...a, ...b]

let A = new Set(a)
let B = new Set(b)
let C = new Set(c)

console.log(C)
```

```
Set(6) {1, 2, 3, 4, 5,6}
```

Intersection of sets

To find an intersection of two sets can be achieved using filter. Lets find the intersection of set A and set B (A ∩ B)

```
let a = [1, 2, 3, 4, 5]
let b = [3, 4, 5, 6]

let A = new Set(a)
let B = new Set(b)

let c = a.filter((num) => B.has(num))
let C = new Set(c)

console.log(C)
```

```
Set(3) {3, 4, 5}
```

Difference of sets

To find an the difference between two sets can be achieved using filter. Lets find the different of set A and set B (A \ B)

```
let a = [1, 2, 3, 4, 5]
let b = [3, 4, 5, 6]

let A = new Set(a)
let B = new Set(b)

let c = a.filter((num) => !B.has(num))
let C = new Set(c)

console.log(C)
```

```
Set(2) {1, 2}
```

Map

Creating an empty Map

```
const map = new Map()
console.log(map)
```

```
Map(0) {}
```

Creating an Map from array

```
countries = [
  ['Finland', 'Helsinki'],
  ['Sweden', 'Stockholm'],
  ['Norway', 'Oslo'],
]
const map = new Map(countries)
console.log(map)
console.log(map.size)
```

```
Map(3) {"Finland" => "Helsinki", "Sweden" => "Stockholm", "Norway" => "Oslo"}
3
```

Adding values to the Map

```
const countriesMap = new Map()
console.log(countriesMap.size) // 0
countriesMap.set('Finland', 'Helsinki')
countriesMap.set('Sweden', 'Stockholm')
countriesMap.set('Norway', 'Oslo')
console.log(countriesMap)
console.log(countriesMap.size)
```

```
Map(3) {"Finland" => "Helsinki", "Sweden" => "Stockholm", "Norway" => "Oslo"}
3
```

Getting a value from Map

```
console.log(countriesMap.get('Finland'))
```

```
Helsinki
```

Checking key in Map

Check if a key exists in a map using *has* method. It returns *true* or *false*.

```
console.log(countriesMap.has('Finland'))
```

```
true
```

Getting all values from map using loop

```
for (const country of countriesMap) {
  console.log(country)
}
```

```
(2) ["Finland", "Helsinki"]
(2) ["Sweden", "Stockholm"]
(2) ["Norway", "Oslo"]
```

```
for (const [country, city] of countriesMap){
 console.log(country, city)
}
```

```
Finland Helsinki
Sweden Stockholm
Norway Oslo
```

🌕 You established a big milestone, you are unstoppable. Keep going! You have just completed day 10 challenges and you are 10 steps a head in to your way to greatness. Now do some exercises for your brain and for your muscle.

Exercises

Exercises:Level 1

```
const a = [4, 5, 8, 9]
const b = [3, 4, 5, 7]
const countries = ['Finland', 'Sweden', 'Norway']
```

1. create an empty set
2. Create a set containing 0 to 10 using loop
3. Remove an element from a set
4. Clear a set
5. Create a set of 5 string elements from array
6. Create a map of countries and number of characters of a country

Exercises:Level 2

1. Find a union b
2. Find a intersection b
3. Find a with b

Exercises:Level 3

1. How many languages are there in the countries object file.

2. *** Use the countries data to find the 10 most spoken languages:

```
// Your output should look like this
console.log(mostSpokenLanguages(countries, 10))
[
  { English: 91 },
  { French: 45 },
  { Arabic: 25 },
  { Spanish: 24 },
  { Russian: 9 },
   { Portuguese: 9 },
   { Dutch: 8 },
   { German: 7 },
   { Chinese: 5 },
   { Swahili: 4 },
   { Serbian: 4 }
 ]

// Your output should look like this
console.log(mostSpokenLanguages(countries, 3))
[
{English:91},
{French:45},
{Arabic:25}
]
```

🎉 CONGRATULATIONS ! 🎉

&lt;&lt; Day 9 | Day 11 &gt;&gt;