

Day 10 - Sets and Maps

Set

A collection of elements. It can only contain unique elements.

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 element

```
for (const language of setOfLanguages) {
  console.log(language)
}
// English Finnish French Spanish
```

Adding an element to a set

```
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.

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

Deleting an element in a set

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

```
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.

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

Clearing the set

Removes all the elements from a set

```
companies.clear()
console.log(companies)
// Set(0) {}
```

Full Example using a 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 },
//]

```

Another use case of set (count a 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

We can unify two sets using the spread operator. Lets find the union of set A and set B ($A \cup B$)

```

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

The intersection of two sets can be achieved using filter. Lets find the intersection of set A and set B ($A \cap B$)

```

let a = [1,2,3,4,5]
let b = [3,4,5,6,7]

```

```
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

We can find the difference between two sets, we can use the filter method. ($A \setminus B$)

```
let a = [1,2,3,4,5]
let b = [4,5,6,7,8]

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(6) {1,2,3,6,7,8}
```

Map

Creating an empty Map

```
const map = new Map()
console.log(map)
// Map(0) {}
```

Creating a Map from an array

```
countries = [
  ['Finland', 'Helsinki'],
  ['Sweden', 'Stockholm'],
  ['Norway', 'Oslo'],
]
const map = new Map(countries)
console.log(map)
// Map(3) {"Finland" => "Helsinki", "Sweden" => "Stockholm", "Norway" => "Oslo"}
console.log(map.size)
// 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) // Map(3) {"Finland" => "Helsinki", "Sweden" => "Stockholm", "Norway" => "Oslo"}
console.log(countriesMap.size) // 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. Returns a Boolean.

```
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
```