Associative Arrays and Maps

SoftUni Team

Technical Trainers

Software University

http://softuni.bg

Table of Content

1. Associative Arrays
   - Definition
   - Declaration
   - Attributes
   - Looping
2. Maps
   - Definition
   - Declaration
   - Functions
   - Sorting

Have a Question?

SoftUni Foundation

sli.do

#tech-fund

Associative Arrays

Storing keys and values

What is an Associative Array ?

- Associative arrays are arrays indexed by **keys**
- Hold a set of pairs `[key => value]`
  - The key can either be an integer or a string
  - The value can be of any type

Key	Value
`John Smith`	`+1-555-8976`
`Lisa Smith`	`+1-555-1234`
`Sam Doe`	`+1-555-5030`

Declaration

- We can declare associative array **dynamically**

```
let arr = [];
arr["one"] = 1;
arr["two"] = 2;
```

- Keep in mind that if you use **named** indexes, JavaScript will **redefine** the array to a **standard object**.

```
arr[0]; // will return undefined
arr.length; // will return 0
```

Attributes

- The syntax for accessing the value of a key is:

```
arrayName["key"] // person["age"]
```

or

```
arrayName[key] // key = "age"; person[key]
```

- Assigning the value to a variable

```
let value = arrayName[key];
```

Looping

- Using for-in loop

```
let arr = [];
arr["one"] = 1;
arr["two"] = 2;
arr["three"] = 3;
for(let key in arr) {
   console.log(key + " = " + arr[key]);
}
```

```
one = 1
two = 2
three = 3
```

Problem Phone book

- Write a function that reads **names** and **numbers**, store them in an array and print them. If same name occurs, safe the **latest** number

Input	Output
Tim 0834212554	Tim -> 0876566344
Peter 0877547887	Peter -> 0877547887
Bill 0896543112	Bill -> 0896543112
Tim 0876566344	

Solution Phone book

```
function solve(input){
    let arr = [];
    for(let string of input){
        let tokens = string.split(" ");
        let name = tokens[0];
        let number = tokens[1];
        arr[name] = number;
    }
    for(let key in arr){
        console.log(`${key} -> ${arr[key]}`);
    }
}
```

Check your solution here: https://judge.softuni.bg/Contests/1231/

Map()

Maps

Storing key-value pairs

What is a Map ?

SoftUni Foundation

- A Map object iterates its elements in **insertion order**
- A for-of loop returns an array of **[key, value]** for each iteration
- Pure **JavaScript objects** are similar to **Maps** in that both let you set keys to values, delete keys and detect whether something is stored in a key
- A Map may **perform better** in scenarios involving **frequent addition and removal** of key pairs

Adding/Accessing Elements

- **.set**(key, value) – adds a new key-value pair

```
let map = new Map()
map.set(1, "one") // key – 1, value – one
map.set(2, "two") // key – 2, value – two
```

- **.get**(key) – returns the value of the given key

```
map.get(2) // two
map.get(1) // one
```

Contain/Delete

- **.has(key)** – checks if the map has the given key

```
map.has(2) // true
map.has(4) // false
```

- **.delete(key)** – returns true if it exists and has been removed.

```
map.delete(1) // removes 1 from the map
```

- **.clear**() – removes all key-value pairs

Iterators

- **.entries**() – returns Iterator – array of [key, value]
- **.keys**() – returns Iterator with all the keys
- **.values**() – returns Iterator with all the values

```
let entries = Array.from(map.entries())
// [[2, two], [3, three]]
let keys = Array.from(map.keys()) // [2, 3]
let vals = Array.from(map.values()) // [two, three]
```

Entries, keys and values returns an Iterator, so we transform it into an Array

Problem Storage

- Write a function that **stores products** and their **quantity**. If the same product appears **more than once**, add the **new quantity** to the old one.

Input	Output
tomatoes 10 coffee 5 olives 100 coffee 40	tomatoes -> 10 coffee -> 45 olives -> 100

Solution Storage

```
function solve(arr){
   let map = new Map();
   for(let string of arr){
       let tokens = string.split(" ");
       let product = tokens[0];
       let quantity = Number(tokens[1]);
       if(!map.has(product)){
           map.set(product, quantity);
       } else {
           let currQuantity = map.get(product);
           let newQuantity = currQuantity += quantity;
           map.set(product, newQuantity);
        }
    }
    // TODO: Print Map
}
```

Check your solution here: https://judge.softuni.bg/Contests/1231/

Map sorting

- To sort a map, we just make it into an array and use the sort array function

```
let map = new Map();
map.set("one", 1);
map.set("eight", 8);
map.set("two", 2);
let sorted = […map.entries()]
.sort((a, b) => a[1] - b[1]);
for (let [key, value] of sorted){
    console.log(`${key} -> ${value}`);
}
```

[["one", 1], ["eight", 8], ["two", 2]]

returns array of arrays of 2 elements

sort by the values (idx 1 of each array)

sorted -> [["one", 1], ["two", 2]], ["eight", 8]]

Problem School grades

- Write a function to **store students** with all of their **grades**. If a student appears **more than once**, add the **new grades**. At the end print the students sorted by **average grade**.

Input	Output

`Lilly 4 6 6 5`
`Tim 5 6`
`Tammy 2 4 3`
`Tim 6 6`
	`Tammy: 2, 4, 3`
	`Lilly: 4, 6, 6, 5`
	`Tim: 5, 6, 6, 6`

Solution School grades

```
function solve(arr){
    let map = new Map();
    for(let string of arr){
       let tokens = string.split(" ");
       let name = tokens[0];
       let grades = tokens.splice(1, tokens.length).map(Number);
       //TODO: fill the map
    }
    let sorted = [...map].sort((a, b) => average(a, b));
     //TODO: print each key and joined values
} //TODO: implement the average function
```

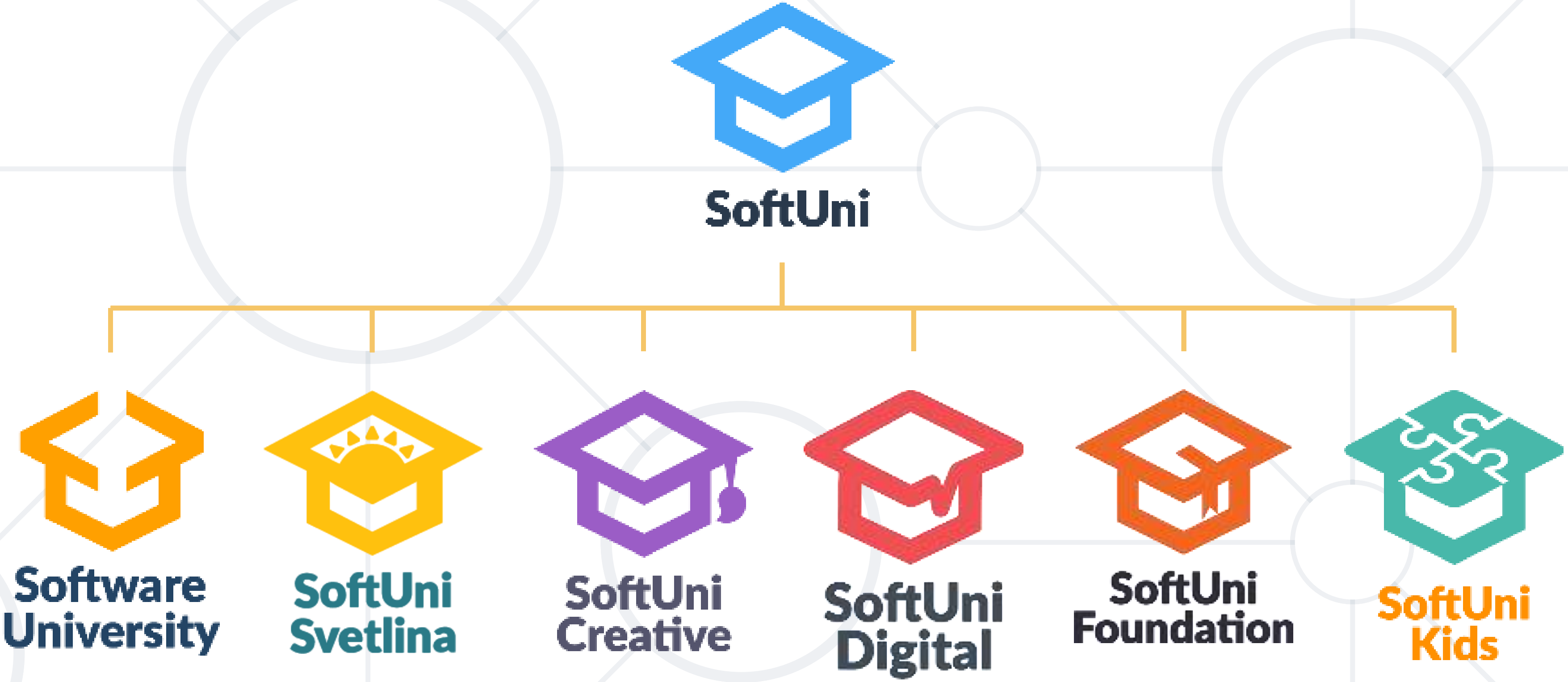
Check your solution here: https://judge.softuni.bg/Contests/1231/

Live Exercises

Summary

- We can use both **Arrays** and **Maps** to store key-value pairs
- Maps are a better way to do it because:
  - They are **iterable**
  - They have **size property**
  - They are better for **adding** and **deleting** many key-value pairs

Questions?


SoftUni

Software University

SoftUni Svetlina

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

SoftUni Diamond Partners

XSsoftware

SBTech
we know sports

telenor

SoftwareGroup
doing it right

NETPEAK

SmartIT

Postbank
Решения за твоето утре

SUPER HOSTING .BG

INDEAVR
Serving the high achievers

INFRAGISTICS

LIEBHERR

æternity

codexio

SoftUni Organizational Partners

IS ИНФОРМАЦИОННО ОБСЛУЖВАНЕ

OneBit SOFTWARE

WORLD OF MYTHS

Lukanet.com

codexio

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education and Employment Opportunities
  - softuni.bg
- Software University Foundation
  - http://softuni.foundation/
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg

License

- This course (slides, examples, demos, videos, homework, etc.) is licensed under the "Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International" license