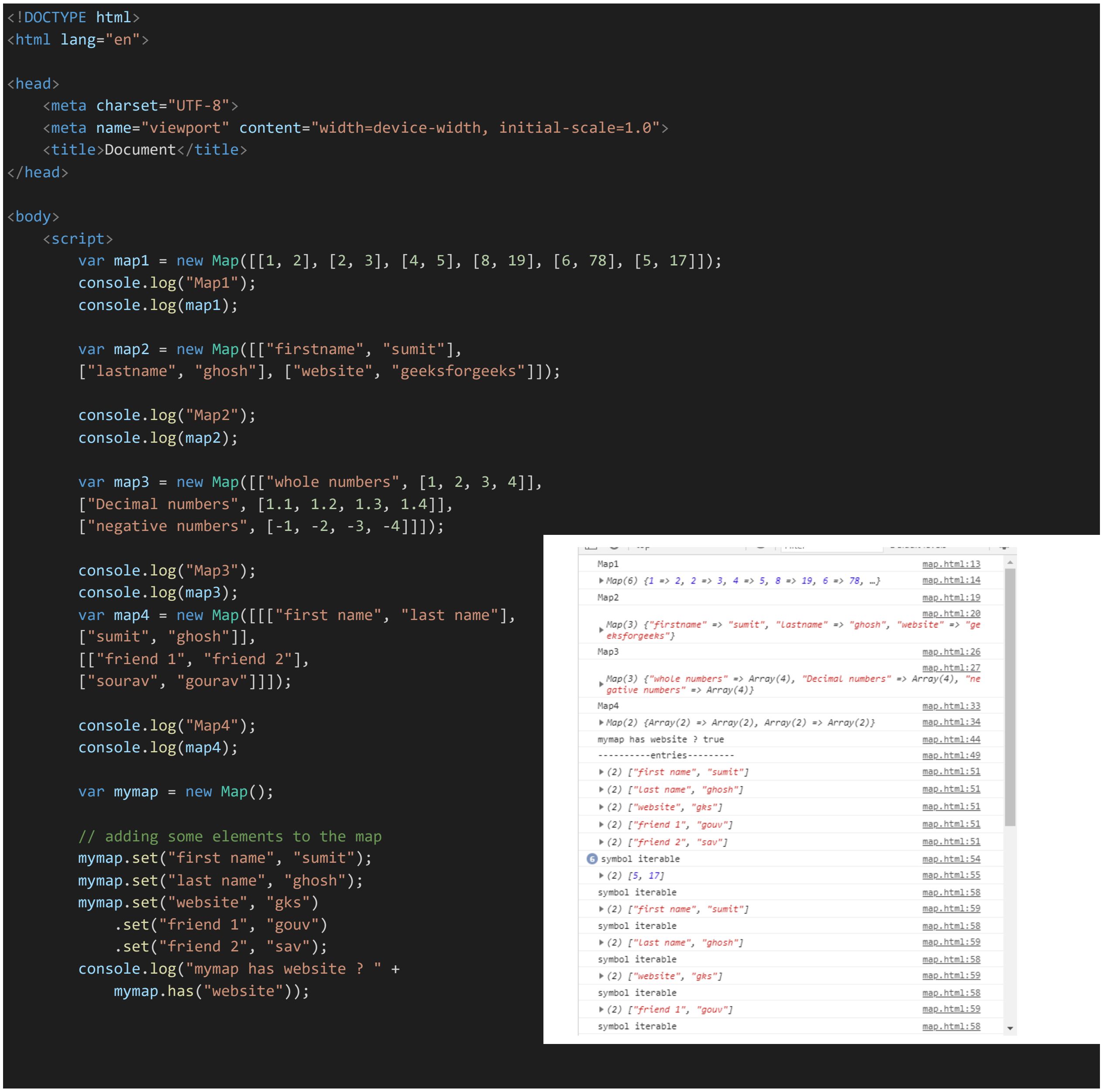
Name : Mariyam Mahnoor

Course : Blockchain Programming

Section:“B”

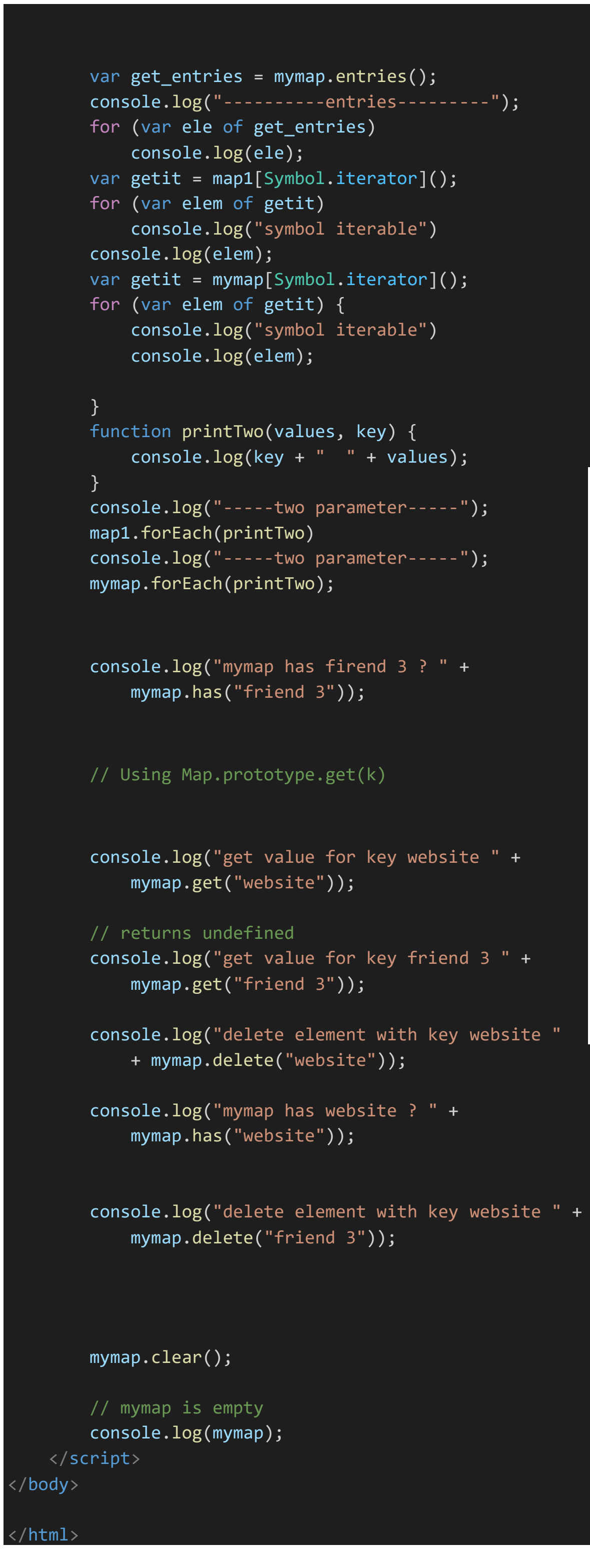
Map :

Map is a collection of elements where each element is stored as a *Key, value* pair. *Map* object can hold both *objects and primitive* values as either key or value.

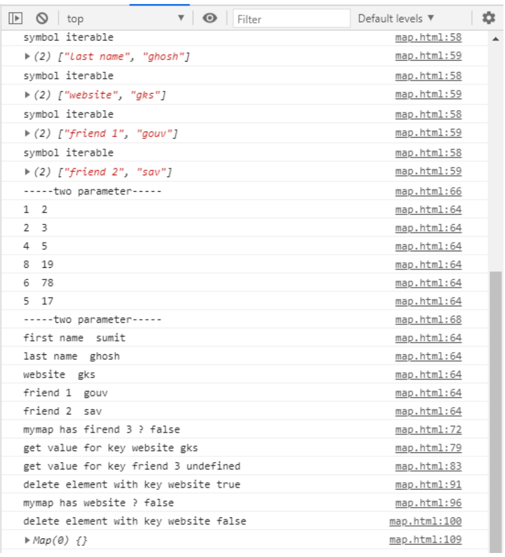


```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>

<body>
    <script>
        var map1 = new Map([[1, 2], [2, 3], [4, 5], [8, 19], [6, 78], [5, 17]]);
        console.log("Map1");
        console.log(map1);

        var map2 = new Map([["firstname", "sumit"],
        ["lastname", "ghosh"], ["website", "geeksforgeeks"]]);

        console.log("Map2");
        console.log(map2);

        var map3 = new Map([["whole numbers", [1, 2, 3, 4]],
        ["Decimal numbers", [1.1, 1.2, 1.3, 1.4]],
        ["negative numbers", [-1, -2, -3, -4]]]);

        console.log("Map3");
        console.log(map3);
        var map4 = new Map([[["first name", "last name"],
        ["sumit", "ghosh"]],
        [["friend 1", "friend 2"],
        ["sourav", "gourav"]]]);

        console.log("Map4");
        console.log(map4);

        var mymap = new Map();

        // adding some elements to the map
        mymap.set("first name", "sumit");
        mymap.set("last name", "ghosh");
        mymap.set("website", "gks")
            .set("friend 1", "gouv")
            .set("friend 2", "sav");
        console.log("mymap has website ? " +
            mymap.has("website"));
```

```
Map1                                                              map.html:13
▶Map(6) {1 => 2, 2 => 3, 4 => 5, 8 => 19, 6 => 78, …}             map.html:14
Map2                                                              map.html:19
                                                                  map.html:20
▶Map(3) {"firstname" => "sumit", "lastname" => "ghosh", "website" => "ge
 eksforgeeks"}
Map3                                                              map.html:26
                                                                  map.html:27
▶Map(3) {"whole numbers" => Array(4), "Decimal numbers" => Array(4), "ne
 gative numbers" => Array(4)}
Map4                                                              map.html:33
▶Map(2) {Array(2) => Array(2), Array(2) => Array(2)}              map.html:34
mymap has website ? true                                          map.html:44
----------entries---------                                        map.html:49
▶(2) ["first name", "sumit"]                                      map.html:51
▶(2) ["last name", "ghosh"]                                       map.html:51
▶(2) ["website", "gks"]                                           map.html:51
▶(2) ["friend 1", "gouv"]                                         map.html:51
▶(2) ["friend 2", "sav"]                                          map.html:51
6 symbol iterable                                                 map.html:54
▶(2) [5, 17]                                                      map.html:55
symbol iterable                                                   map.html:58
▶(2) ["first name", "sumit"]                                      map.html:59
symbol iterable                                                   map.html:58
▶(2) ["last name", "ghosh"]                                       map.html:59
symbol iterable                                                   map.html:58
▶(2) ["website", "gks"]                                           map.html:59
symbol iterable                                                   map.html:58
▶(2) ["friend 1", "gouv"]                                         map.html:59
symbol iterable                                                   map.html:58
```



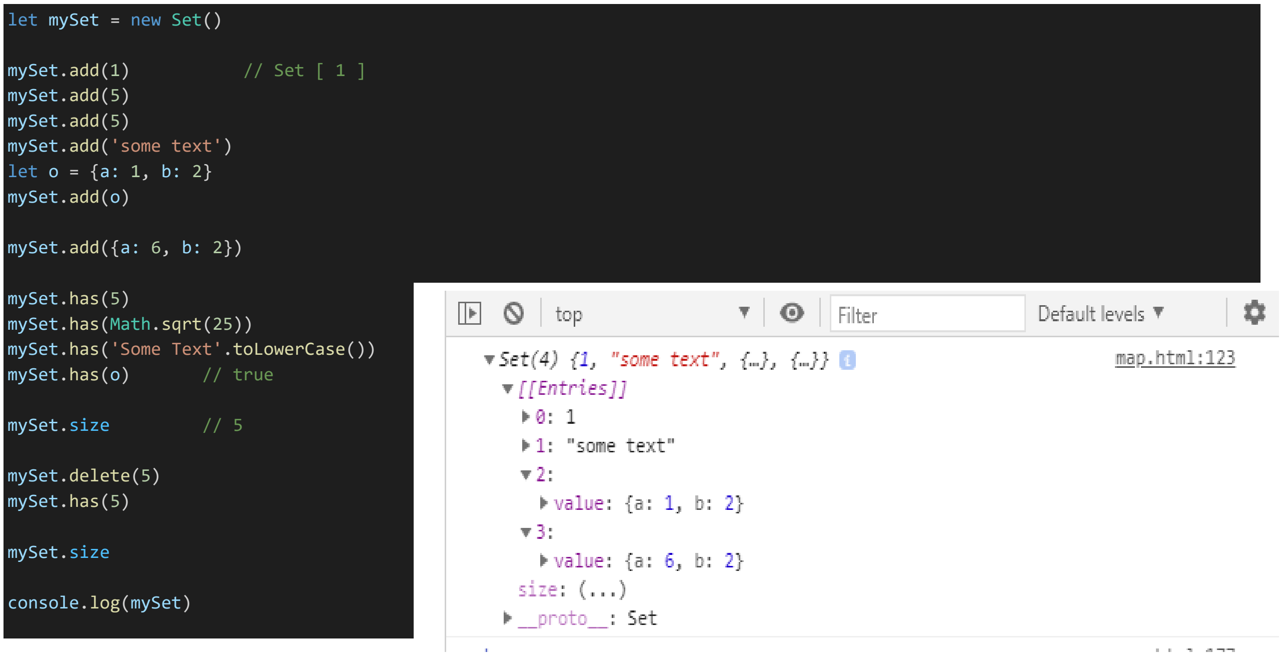
```
        var get_entries = mymap.entries();
        console.log("----------entries---------");
        for (var ele of get_entries)
            console.log(ele);
        var getit = map1[Symbol.iterator]();
        for (var elem of getit)
            console.log("symbol iterable")
        console.log(elem);
        var getit = mymap[Symbol.iterator]();
        for (var elem of getit) {
            console.log("symbol iterable")
            console.log(elem);

        }
        function printTwo(values, key) {
            console.log(key + "  " + values);
        }
        console.log("-----two parameter-----");
        map1.forEach(printTwo)
        console.log("-----two parameter-----");
        mymap.forEach(printTwo);


        console.log("mymap has firend 3 ? " +
            mymap.has("friend 3"));


        // Using Map.prototype.get(k)


        console.log("get value for key website " +
            mymap.get("website"));

        // returns undefined
        console.log("get value for key friend 3 " +
            mymap.get("friend 3"));

        console.log("delete element with key website "
            + mymap.delete("website"));

        console.log("mymap has website ? " +
            mymap.has("website"));


        console.log("delete element with key website " +
            mymap.delete("friend 3"));




        mymap.clear();

        // mymap is empty
        console.log(mymap);
    </script>
</body>

</html>
```



```
symbol iterable                                  map.html:58
▸(2) ["last name", "ghosh"]                      map.html:59
symbol iterable                                  map.html:58
▸(2) ["website", "gks"]                          map.html:59
symbol iterable                                  map.html:58
▸(2) ["friend 1", "gouv"]                        map.html:59
symbol iterable                                  map.html:58
▸(2) ["friend 2", "sav"]                         map.html:59
-----two parameter-----                          map.html:66
1  2                                             map.html:64
2  3                                             map.html:64
4  5                                             map.html:64
8  19                                            map.html:64
6  78                                            map.html:64
5  17                                            map.html:64
-----two parameter-----                          map.html:68
first name  sumit                                map.html:64
last name  ghosh                                 map.html:64
website  gks                                     map.html:64
friend 1  gouv                                   map.html:64
friend 2  sav                                    map.html:64
mymap has firend 3 ? false                       map.html:72
get value for key website gks                    map.html:79
get value for key friend 3 undefined             map.html:83
delete element with key website true             map.html:91
mymap has website ? false                        map.html:96
delete element with key website false           map.html:100
▸Map(0) {}                                      map.html:109
```

Set:

Set objects are collections of values. You can iterate through the elements of a set in insertion order. A value in the Set **may only occur once**; it is unique in the Set's collection.



```
let mySet = new Set()

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

mySet.add({a: 6, b: 2})

mySet.has(5)
mySet.has(Math.sqrt(25))
mySet.has('Some Text'.toLowerCase())
mySet.has(o)       // true

mySet.size         // 5

mySet.delete(5)
mySet.has(5)

mySet.size

console.log(mySet)
```

```
▼Set(4) {1, "some text", {…}, {…}}                    map.html:123
  ▼[[Entries]]
    ▶0: 1
    ▶1: "some text"
    ▼2:
      ▶value: {a: 1, b: 2}
    ▼3:
      ▶value: {a: 6, b: 2}
   size: (...)
  ▶__proto__: Set
```