# Maps

Maps hold key-value pair where key and value can be of any type. Maps also allow to loop through them using loops, but normal objects don’t.

Maps also supports multiple other methods: set(),get(),clear(),delete(),has(),forEach(),entries(),keys(),values()

// map can store key-value pair of any type of data

//create new map instance

const myMap= new Map();

//initial keys by storing different type of data in them

const key1 = "string key 1",

    key2 = {},

    key3 = function() {console.log("this is my function");}

// set key-value in map. we are storing different types of objects

// in value against each key

myMap.set(key1, ()=> console.log("function sored in key1"));

myMap.set(key2, { subkey1 : "val1", subkey2 : "val2"});

myMap.set(key3, "string value stored in key2");

myMap.get(key1)();

// above thing will return lambda function stored in key1 in map and since we are

// executing it using (); it will log 'function sored in key1'

//if we loop through map using for..of lool then we can access values

// from individual keys and theu map values.

for(let [key, value] of myMap){

    console.log(`key : ${key}`)

    console.log(`value : ${value}`)

}

In normal objects keys value are mainly string values like this. But map can have keys of any types

const obj = {

    'key1' : "value 1",

    'key2' : "value 2",

    key3 : "value 3",

}