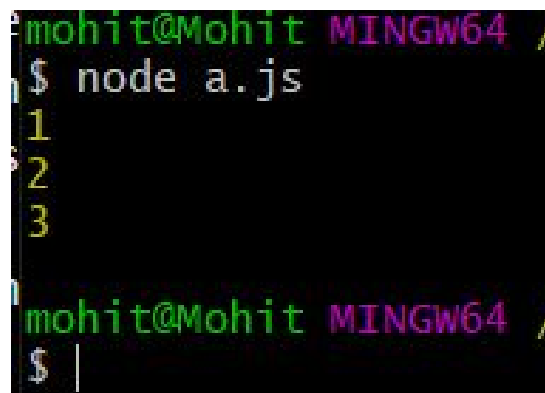


1. Slide 52- Self-Invoking Functions

CODE-

```
var add = (function () {  
    var counter = 0;  
    return function () {  
        return (counter += 1);  
    };  
})();  
  
console.log(add());  
console.log(add());  
console.log(add());
```

OUTPUT-



```
mohit@Mohit MINGW64 /  
$ node a.js  
1  
2  
3  
mohit@Mohit MINGW64 /  
$
```

2. Question 2 - Display usage of All Array Methods

CODE-

```
//_Question 2 - Display usage of All Array Methods  
  
var array = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];  
//_1. concat()  
array = array.concat(11);  
console.log("1: " + array);  
  
//_2. every()  
const lessthan15 = (arrvalue) => arrvalue < 15;  
console.log("2: " + array.every(lessthan15));
```

```
// 3. filter()
array = array.filter(lessthan15);
console.log("3: " + array);

// 4. forEach()
array.forEach((data) => console.log("4: " + data));

// 5. IndexOf()
console.log("5: " + array.indexOf(45));
console.log("5: " + array.indexOf(5));
console.log("5: " + array.indexOf(11));

// 6. join()
console.log("6: " + array.join());

// 7. lastIndexOf()
console.log("7: " + array.lastIndexOf(4));

// 8. map()
array.map((x) => x + 2);

// 9. pop()
console.log("9: " + array.pop());

// 10. push()
console.log("10: " + array.push(12));

// 11. reverse()
console.log("11: " + array.reverse());

// 12. shift()
console.log("12: " + array.shift());

// 13. slice()
console.log("13: " + array.slice(2, 6));

// 14. sort()
console.log("14: " + array.sort());
```

```
// 15. splice()
console.log("15: " + array.splice(3, 5));

// 16. toString()
console.log("16: " + array.toString());

// 17 unShift()
console.log("17: " + array.unshift());
```

OUTPUT-

```
mohit@Mohit MINGW64 /g/Internship
$ node a.js
1: 1,2,3,4,5,6,7,8,9,10,11
2: true
3: 1,2,3,4,5,6,7,8,9,10,11
4: 1
4: 2
4: 3
4: 4
4: 5
4: 6
4: 7
4: 8
4: 9
4: 10
4: 11
5: -1
5: 4
5: 10
6: 1,2,3,4,5,6,7,8,9,10,11
7: 3
9: 11
10: 11
11: 12,10,9,8,7,6,5,4,3,2,1
12: 12
13: 8,7,6,5
14: 1,10,2,3,4,5,6,7,8,9
15: 3,4,5,6,7
16: 1,10,2,8,9
17: 5
```

3. Create a function which takes input as a string and it returns true if
a) String starts with lion
b) String ends with cat
c) String has abc (b can be n>=1 times) anywhere in between the string.
And also print the location of a/b/c if true or else return false.

CODE-

```
var re = new RegExp("abb*c");

const findIndices = (str, char) =>
  str.split("").reduce((indices, letter, index) => {
    letter === char && indices.push(index);
    return indices;
  }, []);

var fun = (string) => {
  if (string.startsWith("lion") && string.endsWith("cat") &&
string.match(re)) {
    console.log(`${string} true`);
    console.log("Postions of a:", findIndices(string, "a"));
    console.log("Postions of b:", findIndices(string, "b"));
    console.log("Postions of c:", findIndices(string, "c"));
  } else console.log(`${string} false`);
  return "";
};

console.log(fun("lionnodeabbccat"));

console.log(fun("cat"));

console.log(fun("lion"));

console.log(fun("lioncat"));
```

OUTPUT-

```
mohit@mohit MINGW64 /g/Internship +  
$ node a.js  
lionnodeabbccat true  
Postions of a: [ 8, 13 ]  
Postions of b: [ 9, 10 ]  
Postions of c: [ 11, 12 ]  
  
cat false  
  
lion false  
  
lioncat false
```

4. Create a function which takes array as an input

a) Sort array in ascending order

b) Multiply each number by 10

c) Return those numbers which are divisible by 3

CODE-

```
var fun = (array) => {  
  console.log(`Initial Array: `, array);  
  console.log(`Sorted Array: `, array.sort());  
  console.log(  
    `X by 10 Array: `,  
    array.map((data) => data * 10)  
  );  
  console.log(  
    `Divisible by 3 Array: `,  
    array.filter((data) => !(data % 3))  
  );  
  return "";  
};  
  
console.log(fun([9, 8, 7, 6, 5, 4, 3, 2, 1]));
```

OUTPUT-

```
mohit@mohit MINGW64 /g/Internship + Pro
$ node a.js
Initial Array: [
  9, 8, 7, 6, 5,
  4, 3, 2, 1
]
Sorted Array: [
  1, 2, 3, 4, 5,
  6, 7, 8, 9
]
X by 10 Array: [
  10, 20, 30, 40, 50,
  60, 70, 80, 90
]
Divisible by 3 Array: [ 3, 6, 9 ]
```

5.Difference between '==' and '==='

CODE-

```
var a = 1;
var b = "1";

console.log("");
console.log("");
console.log(a == b);
console.log(a === b);

console.log("");

var c = 1;
var d = true;

console.log(c == d);
console.log(c === d);
```

OUTPUT-

```
mohit@mohit MINGW64 /
$ node a.js

true
false

true
false
```