

IP-PRACTICAL NO 5

Date of Performance: 20-09-2022

Software Requirement: Visual Studio Code, Notepad, Notepad++, NodeJS

Aim: To use JavaScript to write codes based on normal, arrow, anonymous and generator functions .

Objectives: The aim of this experiment is that the students will be able to:

- Implement JavaScript codes based on functions.
- Understand different features of JavaScript functions & its usage in code.

Outcomes: After study of this experiment, the students will be able to:

- Write and run functions in JavaScript.
- Implement the usage of different functions in JavaScript.

Prerequisite: Basic knowledge of JavaScript syntax required.

Theory:

JavaScript is the world most popular lightweight, interpreted compiled programming language. It is also known as scripting language for web pages.

Function in any programming language is the basic building block to create and combine the related bits of code. Every programming language provides certain kinds of practices to write any function.

Syntax of regular functions:-

```
let x = function function_name(parameters){  
    // body of the function  
};
```

ES6 introduced a new and shorter way of declaring an anonymous function, which is known as **Arrow Functions**. In an Arrow function, everything remains the same, except here we don't need the *function* keyword also. Here, we define the function by a single parenthesis and then '=>' followed by the function body. The **arrow function** syntax is one of the most used and efficient ones to create a function in JavaScript. To write the arrow

function, simply create any variable it can be const, let, or var but always do prefer with const to avoid unnecessary problems.

Syntax:

```
let myFunction = (arg1, arg2, argN) => {  
  statement(s)  
}
```

Anonymous Function is a function that does not have any name associated with it.

Normally we use the *function* keyword before the function name to define a function in JavaScript, however, in anonymous functions in JavaScript, we use only the *function* keyword without the function name.

An anonymous function is not accessible after its initial creation, it can only be accessed by a variable it is stored in as a *function as a value*. An anonymous function can also have multiple arguments, but only one expression.

Syntax:

```
let x = function () {  
  //statements  
};  
x();
```

A **generator-function** is defined like a normal function, but whenever it needs to generate a value, it does so with the yield keyword rather than return. The yield statement suspends function's execution and sends a value back to caller, but retains enough state to enable function to resume where it is left off. When resumed, the function continues execution immediately after the last yield run.

Syntax :

```
function* gen(){  
  yield 1;  
  yield 2;  
  ...  
}
```

Problem Statement:

Implement functions in JS (Arrow, Normal & Anonymous) & generator function.

Source Code:

```
function add() {
  let a = 7,b = 5;
  console.log(`The addition using normal fuction without paramater is ${a + b}`);
}
add();

function add2(a, b) {
  console.log(`The addition using normal fuction with paramater is ${a + b}`);
}
add2(55, 87);

const arrowAdd = () => {
  let a = 64,b = 36;
  console.log(`The addition using arrow fuction without parameter is ${a + b}`);
};
arrowAdd();
const arrowAdd2 = (a, b) => {
  console.log(`The addition using arrow fuction with parameter is ${a + b}`);
};
arrowAdd2(25, 75);

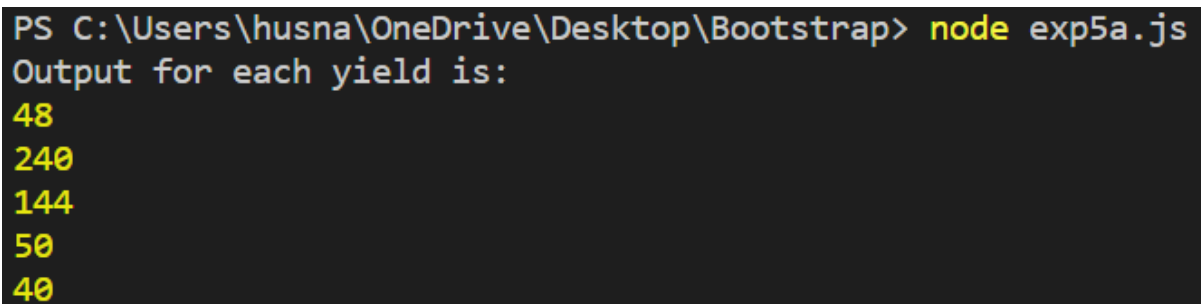
const anonymousAdd = function () {
  let a = 32,b = 88;
  return `The addition using anonymous fuction without parameter is ${a + b}`;
};
let x = anonymousAdd();
console.log(x);
const anonymousAdd2 = function (a, b) {
  return `The addition using anonymous fuction with parameter is ${a + b}`;
};
let y = anonymousAdd2(31, 9);
console.log(y);
```

Output:

```
PS C:\Users\husna\OneDrive\Desktop\Bootstrap> node exp5.js
The addition using normal fuction without paramater is 12
The addition using normal fuction with paramater is 142
The addition using arrow fuction without parameter is 100
The addition using arrow fuction with parameter is 100
The addition using anonymous fuction without parameter is 120
The addition using anonymous fuction with parameter is 40
```

Source Code: Generator function:

```
function* generator(x) {  
    yield x;  
    yield x * 5;  
    yield x * 3;  
    yield x + 2;  
    yield x - 8;  
}  
const gen = generator(48);  
console.log("Output for each yield is:");  
console.log(gen.next().value);  
console.log(gen.next().value);  
console.log(gen.next().value);  
console.log(gen.next().value);  
console.log(gen.next().value);
```

Output:

```
PS C:\Users\husna\OneDrive\Desktop\Bootstrap> node exp5a.js  
Output for each yield is:  
48  
240  
144  
50  
40
```

Conclusion:

From this experiment we understand that JavaScript is a very powerful web language and is used at a large scale. We have learned how to implement JavaScript normal, arrow and anonymous functions/generator functions in programs and understood its syntax.

Performance: 7M	Journal: 3M	Lab Ethics: 2M	Attendance: 3M	Total: 15M	Faculty Sign