NAME: MAYURI SHRIDATTA YERANDE
ROLL NO: 70
CLASS: D15B
IP EXPERIMENT

EXPERIMENT - 5

AIM: Experiment to study the basics of Java Script.

THEORY:

Basics of Javascript :-
JavaScript is a programming language that adds interactivity to your website. This happens in games, in the behavior of responses when buttons are pressed or with data entry on forms; with dynamic styling; with animation, etc.

Different ways of including Java script code in HTML document :-
There are 3 ways to include Javascript in HTML:

External Javascript, load a Javascript file –
<script src="FILE.JS"></script>

Internal Javascript, add a block of code in the HTML document itself –
<script>DO SOMETHING</script>

Inline Javascript, directly add Javascript to an HTML element –
<input type="button" value="Test" onclick="FUNCTION()"/>

Input-output in Java script :-
JavaScript Output defines the ways to display the output of a given code. The output can be display by using four different ways which are listed below:
innerHTML: It is used to access an element. It defines the HTML content.
Syntax:
document.getElementById(id)
document.write(): It is used for testing purpose.
Syntax:
document.write()

window.alert():It displays the content using an alert box.

Syntax:
window.alert()
console.log(): It is used for debugging purposes.
Syntax:
console.log()

Control structures :-
The control structures within JavaScript allow the program flow to change within a unit of code or function. These statements can determine whether or not given statements are executed - and provide the basis for the repeated execution of a block of code.

Functions :-
A JavaScript function is a block of code designed to perform a particular task.
A JavaScript function is executed when "something" invokes it (calls it).
A JavaScript function is defined with the function keyword, followed by a name, followed by parentheses ().
Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
The parentheses may include parameter names separated by commas:
(parameter1, parameter2, ...)
The code to be executed, by the function, is placed inside curly brackets: {}

Arrow Functions :-
An arrow function expression is a compact alternative to a traditional function expression, but is limited and can't be used in all situations.
This creates a function func that accepts arguments arg1..argN, then evaluates the expression on the right side with their use and returns its result.

E6 features(spread, Rest operator,Template Literals etc) :-
Spread syntax (...) allows an iterable, such as an array or string, to be expanded in places where zero or more arguments (for function calls) or elements (for array literals) are expected. In an object literal, the spread syntax enumerates the properties of an object and adds the key-value pairs to the object being created.
Spread syntax looks exactly like rest syntax. In a way, spread syntax is the opposite of rest syntax. Spread syntax "expands" an array into its elements, while rest syntax collects multiple elements and "condenses" them into a single element.
ES6 or the ECMAScript 2015 is the 6th and major edition of the ECMAScript language specification standard. It defines the standard for the implementation of JavaScript and it has become much more popular than the previous edition ES5.

ES6 comes with significant changes to the JavaScript language. It brought several new features like, let and const keyword, rest and spread operators, template literals, classes, modules and many other enhancements to make JavaScript programming easier and more fun. In this article, we will discuss some of the best and most popular ES6 features that we can use in your everyday JavaScript coding.

- let and const Keywords
- Arrow Functions
- Multi-line Strings
- Default Parameters
- Template Literals
- Destructuring Assignment
- Enhanced Object Literals
- Promises
- Classes
- Modules

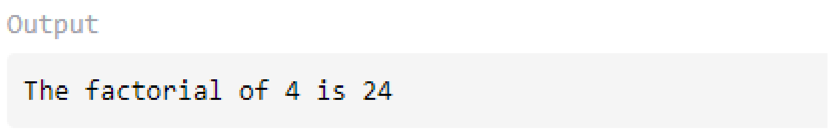
IMPLEMENTATION:

Case 1. To find the factorial of a number.

Program:

```
function factorial(n){
  let answer = 1;
  if (n == 0 || n == 1){
    return answer;
  }else{
    for(var i = n; i >= 1; i--){
      answer = answer * i;
    }
    return answer;
  }
}
let n = 4;
answer = factorial(n)
console.log("The factorial of " + n + " is " + answer);
```

Output:



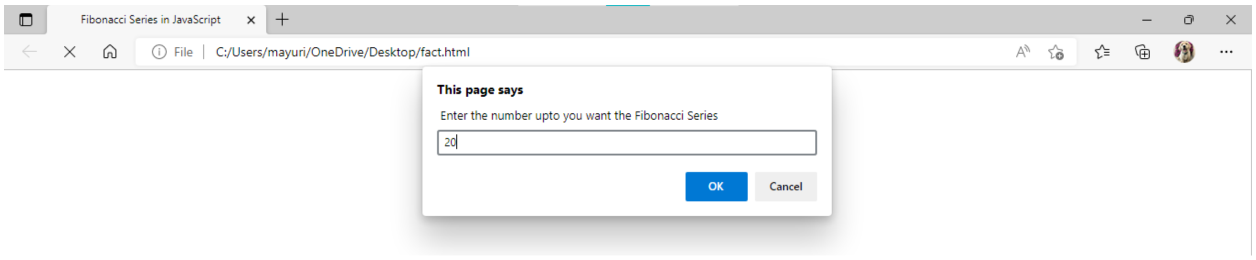
```
Output

The factorial of 4 is 24
```

Case 2. To display the first 20 terms of the Fibonacci series.

Program:

```
<html>
<head>
<title> Fibonacci Series in JavaScript </title>
</head>
<body>
<script>
var n1 = 0,  n2 = 1, next_num, i;
var num = parseInt (prompt (" Enter the number upto you want the Fibonacci Series "));
document.write( "Fibonacci Series: ");
for ( i = 1; i <= num; i++)
{  document.write (" <br> " +  n1);
   next_num = n1 + n2;
   n1 = n2;
   n2 = next_num;
}

</script>
</body>
</html>
```

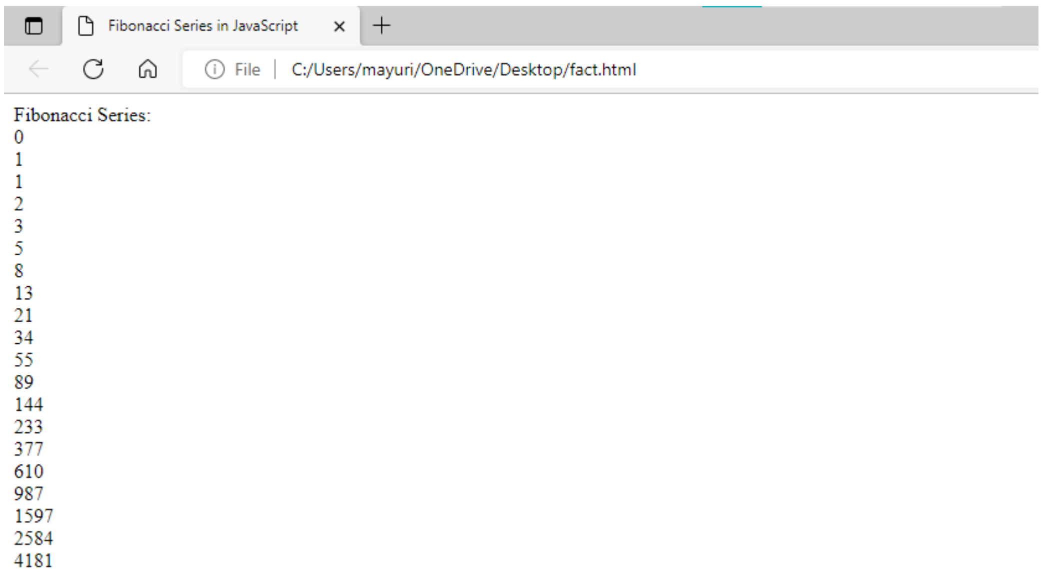
Output:


Fibonacci Series in JavaScript
File
C:/Users/mayuri/OneDrive/Desktop/fact.html

This page says

Enter the number upto you want the Fibonacci Series

20


Fibonacci Series in JavaScript
File
C:/Users/mayuri/OneDrive/Desktop/fact.html

```
Fibonacci Series:
0
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
```

2. Program to implement different arithmetic operations using function and Arrow function.

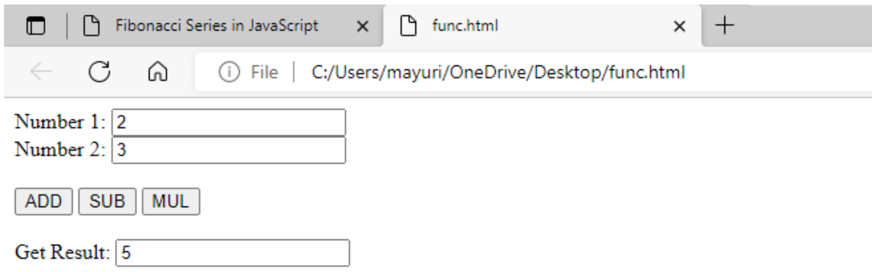
Using Functions:-

Program:

```
<!DOCTYPE html>
<html>
<body>
  <script type="text/javascript">
    function multiply(){
      a=Number(document.my_cal.first.value);
      b=Number(document.my_cal.second.value);
      c=a*b;
      document.my_cal.total.value=c;
    }

    function addition(){
      a=Number(document.my_cal.first.value);
      b=Number(document.my_cal.second.value);
      c=a+b;
      document.my_cal.total.value=c;
    }


    function subtraction(){
      a=Number(document.my_cal.first.value);
      b=Number(document.my_cal.second.value);
      c=a-b;
      document.my_cal.total.value=c;
    }

  </script>

  <form name="my_cal">

    Number 1: <input type="text" name="first">

    <br>

    Number 2: <input type="text" name="second">

    <br><br>

    <input type="button" value="ADD" onclick="javascript:addition();">
    <input type="button" value="SUB" onclick="javascript:subtraction();">
    <input type="button" value="MUL" onclick="javascript:multiply();">

    <br><br>

    Get Result: <input type="text" name="total">

  </body>
```

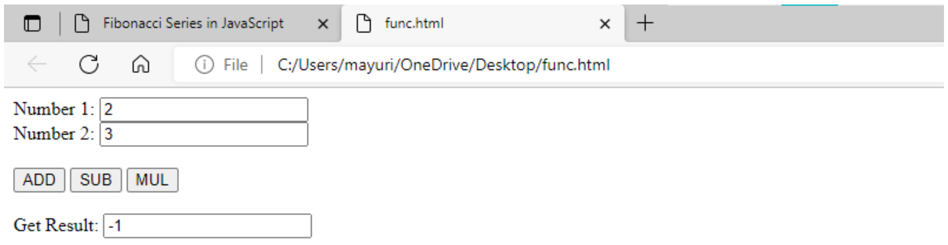
```
</html>
```

Output:


Fibonacci Series in JavaScript
func.html
File
C:/Users/mayuri/OneDrive/Desktop/func.html
ADD
SUB
MUL

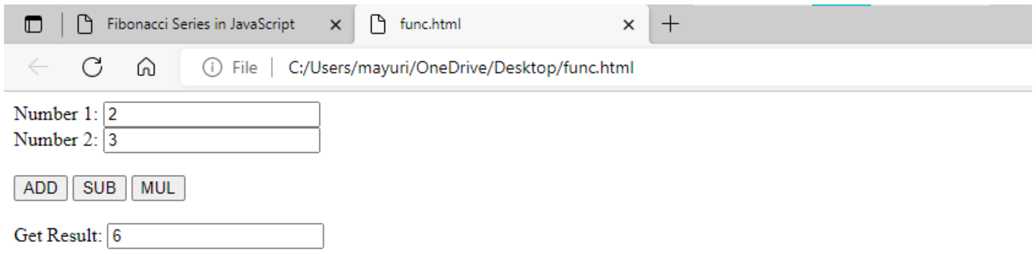
Number 1: 2
Number 2: 3

Get Result: 5


Fibonacci Series in JavaScript
func.html
File
C:/Users/mayuri/OneDrive/Desktop/func.html
ADD
SUB
MUL

Number 1: 2
Number 2: 3

Get Result: -1


Fibonacci Series in JavaScript
func.html
File
C:/Users/mayuri/OneDrive/Desktop/func.html
ADD
SUB
MUL

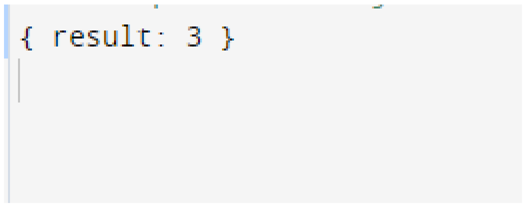
Number 1: 2
Number 2: 3

Get Result: 6

Using Arrow Functions:-

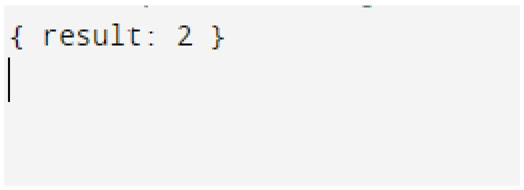
Program:

```
const sum = (a, b) => ({result: a + b})
console.log(sum(1, 2))
```



```
{ result: 3 }
```

```
const sub = (a, b) => ({result: a - b})
console.log(sub(5, 3))
```



```
{ result: 2 }
```

CONCLUSION: We have studied the basics of Java Script and successfully implemented programs on JavaScript.