

Roll No :- 48

Name:- Manav Jain

Date:-12/08/2023

ASSIGNMENT NO. - 4a

AIM:- Write a program in JavaScript to study conditional statements , Loops and functions

LO MAPPED:- LO4

THEORY:-

Conditional statements in JavaScript are used to make decisions in your code based on certain conditions. They allow your program to execute different blocks of code depending on whether a given condition is true or false. The main types of conditional statements in JavaScript are if, else if, else, and the switch statement.

1)if statement:The basic if statement allows you to execute a block of code if a specified condition is true.

```
if (condition) {  
    // Code to be executed if the condition is true  
}
```

2)if-else statement:The if-else statement allows you to execute one block of code if a condition is true, and another block of code if the condition is false.

```
if (condition) {  
    // Code to be executed if the condition is true  
} else {  
    // Code to be executed if the condition is false
```

```
}
```

3)else if statement:The else if statement allows you to test multiple conditions and execute different blocks of code depending on which condition is true. You can have multiple else if blocks.

```
if (condition1) {  
    // Code to be executed if condition1 is true  
} else if (condition2) {  
    // Code to be executed if condition2 is true  
} else {  
    // Code to be executed if none of the conditions are true  
}
```

4)switch statement:The switch statement allows you to compare a single value against multiple possible cases and execute code based on which case matches the value.

```
switch (expression) {  
    case value1:  
        // Code to be executed if expression === value1  
        break;  
    case value2:  
        // Code to be executed if expression === value2  
        break;  
    // ... more cases ...  
    default:  
        // Code to be executed if none of the cases match  
}
```

Here's an example of using these conditional statements:

```
var num = 10;  
if (num > 0) {  
    console.log("Positive number");
```

```
} else if (num < 0) {  
    console.log("Negative number");  
} else {  
    console.log("Zero");  
}
```

In this example, the if and else if statements are used to determine whether num is positive, negative, or zero, and the appropriate message is logged to the console based on the value of num.

Loops in JavaScript are used to execute a block of code repeatedly as long as a specified condition is true. There are three main types of loops in JavaScript: for loop, while loop, and do-while loop.

5)for loop:The for loop is used when you know the number of iterations you want to perform. It consists of an initialization, a condition, and an iteration statement.

```
for (initialization; condition; iteration) {  
    // Code to be executed in each iteration  
}
```

6)while loop:The while loop is used when you want to execute a block of code as long as a specific condition remains true.

```
while (condition) {  
    // Code to be executed as long as the condition is true  
}
```

7)do-while loop:The do-while loop is similar to the while loop, but it ensures that the code inside the loop is executed at least once, even if the condition is initially false.

```
do {  
    // Code to be executed at least once  
} while (condition);
```

8)forEach loop:The forEach loop is specifically designed to iterate over each element of an array and execute a provided function for each element.

```
array.forEach(function(element, index, array) {  
    // Code to be executed for each element  
});
```

9)for-in loop:The for-in loop is used to iterate over the enumerable properties of an object. It is often used for iterating over the keys (properties) of an object.

```
for (var key in object) {  
    // Code to be executed for each key  
}
```

10)for-of loop:The for-of loop is used to iterate over the values of iterable objects such as arrays, strings, maps, sets, etc.

```
for (var value of iterable) {  
    // Code to be executed for each value  
}
```

11)Return function:Return function contains return statement that contains return statement that returns value to console

```
function add(a, b) {  
    return a + b;  
}
```

```
var sum = add(3, 5);  
console.log(sum);
```

12)Parameterized Function:A parameterized function is a function that accepts parameters (arguments) when it is called. These parameters allow you to pass values into the function to be used in its logic.

Example:

```
function greet(name) {  
    console.log("Hello, " + name + "!");  
}  
greet("Alice"); // Output: Hello, Alice!
```

13)Arrow Function:Arrow functions are a concise way to write functions in JavaScript. They provide a more compact syntax and automatically bind the this value to the surrounding context.

Example:

```
const multiply = (a, b) => a * b;  
console.log(multiply(2, 3)); // Output: 6
```

14)Anonymous Function:An anonymous function is a function without a specified name. They are often used as arguments to higher-order functions or assigned to variables.

Example:

```
var calculate = function(a, b) {  
    return a * b;  
};  
console.log(calculate(4, 5));
```

15)Callback Function:A callback function is a function that is passed as an argument to another function and is executed later, typically after an asynchronous operation or an event occurs.

Example:

```
function processInput(input, callback) {  
    console.log("Processing input: " + input);  
    callback();  
}
```

```
function complete() {  
    console.log("Processing complete.");  
}  
processInput("data", complete)
```

1)Code for loops in javascript :-

```
<html>  
<head>  
    <title>various types of loops</title>  
</head>  
<body>  
<script>  
    loopdemo()  
    function loopdemo()  
    {  
        //for loop  
        console.log("using for loop:")  
        for(let i=1;i<10;i++)  
        {  
            console.log(i);  
        }  
  
        //while loop  
        console.log("Using while loop:");  
        let count = 0;  
        while (count < 3) {  
            console.log(count);  
            count++;  
        }  
  
        //do while loop  
        console.log("using do while loop:")  
        let x=4;
```

```
do{
  console.log("x= "+x)
  x--;
}while(x<4 && x>0);

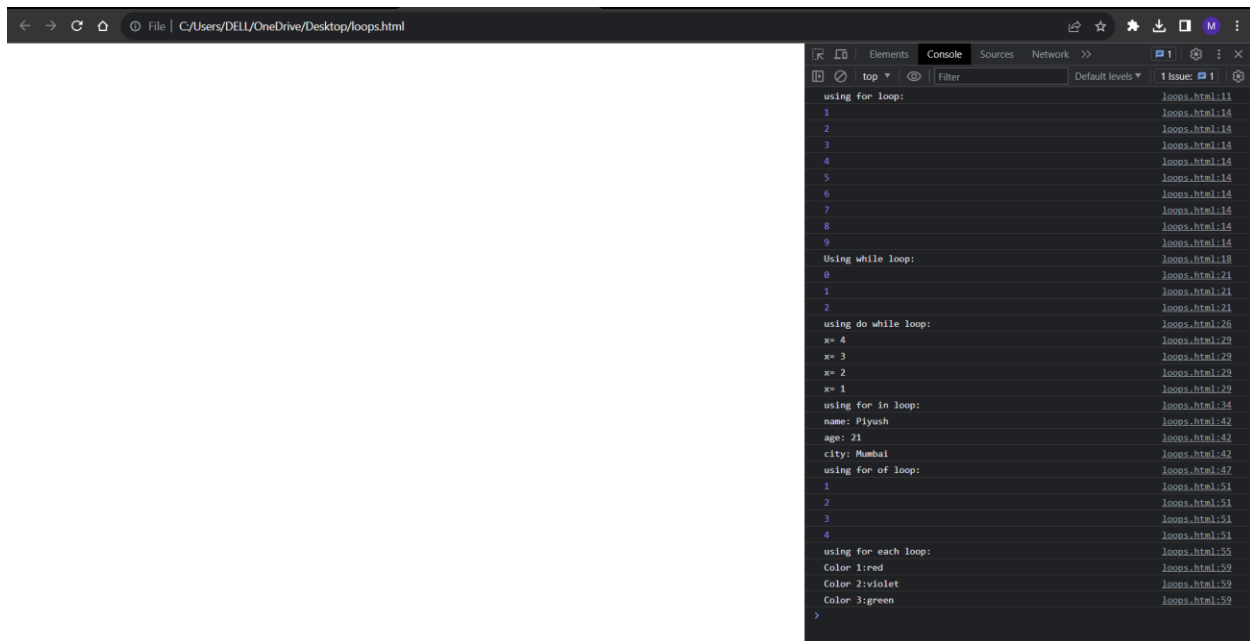
//for in loop
console.log("using for in loop:")
const identity={
  name:"Piyush",
  age:21,
  city:"Mumbai"
}
for (const key in identity){
  console.log(`${key}: ${identity[key]}`);
}

//for of loop
console.log("using for of loop:")
const arr=[1,2,3,4]
for (const x of arr)
{
  console.log(x)
}

//for each loop
console.log("using for each loop:")
const clr=["red","violet","green"]
clr.forEach((x,index)=>
{
  console.log('Color '+(index+1)+':'+x);
})
}
</script>
```

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

Output:-



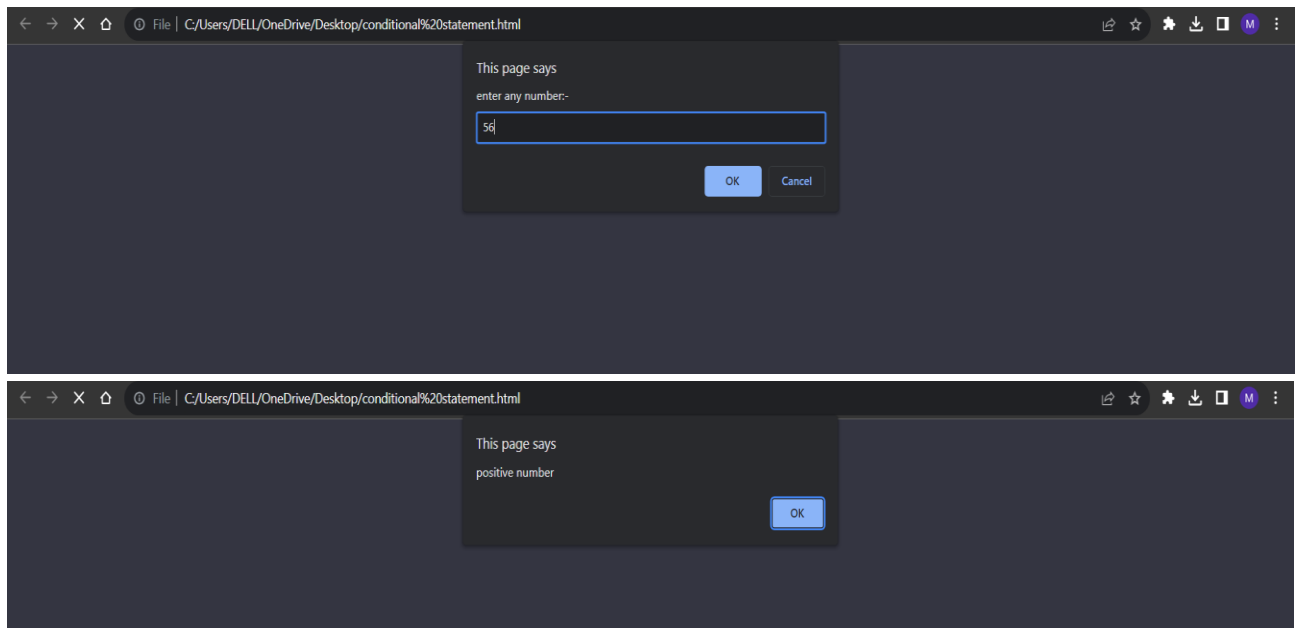
2)Code for conditional statements in JavaScript :-

```
<html>
<head>
  <title>consitional</title>
</head>
<body>
<script>
  let num=prompt("enter any number:-");
  console.log("Conditional statements:-");
  demo(num);
  function demo(num){
    if(num>0)
    {
      alert("positive number");
```



```
}  
else if(num==0)  
{  
    alert("zero is entered");  
}  
else  
{  
    alert("negative number");  
}  
}  
</script>  
</body>  
</html>
```

Output:-



3)Code for functions in javascript :-

```
<html>  
<head>  
    <title>functions</title>  
</head>
```

```
<body>
<script>
  //returning function and parameterized and multiline arrow function
  function multip(a,b)
  {
    console.log("return and parameterized and multiline arrow function")
    return a*b;
  }
  multip(2,3);

  //arrow function
  const myfun = () => {
    console.log("arrow function with no argument");
  }
  myfun();
  const myfun2 = (a) => {
    console.log("arrow function with one argument");
  }
  myfun2();

  //arrow function as expression
  let age=5;
  let welcome = (age<20) ?
  () => console.log("less than 20"):
  () => console.log("more than 20");

  //anonymous function
  console.log("Using anonymous function :")
  const greet = function(name)
  { console.log(`Hello, ${name}!`); };
  greet("PH");

  //function constructor
```

```
console.log("function constructor: ")
const multiple=new Function('a','b','return a*b');
console.log(multiple(3,4));
```

```
// callback function
console.log("Using call back function: ")
function calculate(num1, num2, operation)
{ return operation(num1, num2); }
const sum = (a, b) => a + b;
console.log(calculate(5, 3, sum));
function* fun() {
  yield 10;
  yield 20;
  yield 30;
}
```

```
let gen = fun();
console.log(gen.next().value);
console.log(gen.next().value);
console.log(gen.next().value);
console.log(gen.next().value);
```

</script>

</body>

</html>

Output:-



CONCLUSION:-

Learnt and Implemented various types of functions , loops , conditional statements in javascript and used them in various programs and observed output for each case. Also learnt about types of loops such as for..of , for...each , for...in in JavaScript .