EXPERIMENT NO. 06

DATE OF PERFORMANCE: **GRADE:**

DATE OF ASSESSMENT: **SIGNATURE OF LECTURER/ TTA:**

AIM: Implementation of Java Script Branching and Looping.

THEORY:

CONDITIONAL STATEMENTS:

Conditional statements are used to perform different actions based on different conditions. Very often when you write code, you want to perform different actions for different decisions. You can use conditional statements in your code to do this.

In JavaScript we have the following conditional statements:

- **Use if to specify a block of code to be executed, if a specified condition is true**
- **Use else to specify a block of code to be executed, if the same condition is false**
- **Use else if to specify a new condition to test, if the first condition is false**
- **Use switch to specify many alternative blocks of code to be executed**

THE IF STATEMENT:

Use the if statement to specify a block of JavaScript code to be executed if a condition is true.

SYNTAX:

```
if (condition) {
   block of code to be executed if the condition is true
}
```

EXAMPLE:

```
if (hour < 18) {
   greeting = "Good day";
}
```

THE ELSE STATEMENT:

Use the else statement to specify a block of code to be executed if the condition is false.

SYNTAX:

```
if (condition) {
   block of code to be executed if the condition is true
} else {
   block of code to be executed if the condition is false
}
```

EXAMPLE:

```
if (hour < 18) {
   greeting = "Good day";
} else {
   greeting = "Good evening";
}
```

THE ELSE IF STATEMENT:

Use the else if statement to specify a new condition if the first condition is false.

SYNTAX:

```
if (condition1) {
   block of code to be executed if condition1 is true
} else if (condition2) {
   block of code to be executed if the condition1 is false and condition2 is true
} else {
   block of code to be executed if the condition1 is false and condition2 is false
}
```

EXAMPLE:

```
if (time < 10) {
   greeting = "Good morning";
} else if (time < 20) {
   greeting = "Good day";
} else {
   greeting = "Good evening";
}
```

JAVASCRIPT SWITCH STATEMENT:

The switch statement is used to perform different actions based on different conditions. Use the switch statement to select one of many blocks of code to be executed.

SYNTAX:

```
switch(expression) {
  case n:
    code block
    break;
  case n:
    code block
    break;
  default:
    default code block
}
```

EXAMPLE:

```
switch (new Date().getDay()) {
  case 6:
    text = "Today is Saturday";
    break;
  case 0:
    text = "Today is Sunday";
    break;
  default:
    text = "Looking forward to the Weekend";
}
```

JAVASCRIPT FOR LOOP:

Loops are handy, if you want to run the same code over and over again, each time with a different value.

Different Kinds of Loops
JavaScript supports different kinds of loops:

- **for - loops through a block of code a number of times**
- **for/in - loops through the properties of an object**
- **while - loops through a block of code while a specified condition is true**
- **do/while - also loops through a block of code while a specified condition is true**

THE FOR LOOP:

The for loop is often the tool you will use when you want to create a loop.

The for loop has the following syntax:

```
for (statement 1; statement 2; statement 3) {
   code block to be executed
}
```

Statement 1 is executed before the loop (the code block) starts.
Statement 2 defines the condition for running the loop (the code block).
Statement 3 is executed each time after the loop (the code block) has been executed.

EXAMPLE:

```
for (i = 0; i < 5; i++) {
   text += "The number is " + i + "<br>";
}
```

THE FOR/IN LOOP:

The JavaScript for/in statement loops through the properties of an object:

EXAMPLE:

```
var person = {fname:"John", lname:"Doe", age:25};
var text = "";
var x;
for (x in person) {
   text += person[x];
}
```

THE WHILE LOOP:

The while loop loops through a block of code as long as a specified condition is true.

SYNTAX:

```
while (condition) {
   code block to be executed
}
```

EXAMPLE:

```
while (i < 10) {
   text += "The number is " + i;
   i++;
}
```

THE DO/WHILE LOOP:

The do/while loop is a variant of the while loop. This loop will execute the code block once, before checking if the condition is true, then it will repeat the loop as long as the condition is true.

SYNTAX:

```
do {
   code block to be executed
}
while (condition);
```

JAVASCRIPT BREAK AND CONTINUE:

The break statement "jumps out" of a loop.
The continue statement "jumps over" one iteration in the loop.

THE BREAK STATEMENT:
You have already seen the break statement used in an earlier chapter of this tutorial. It was used to "jump out" of a switch() statement.
The break statement can also be used to jump out of a loop.
The break statement breaks the loop and continues executing the code after the loop (if any)

EXAMPLE:

```
for (i = 0; i < 10; i++) {
   if (i === 3) { break; }
   text += "The number is " + i + "<br>";
}
```

THE CONTINUE STATEMENT:

The continue statement breaks one iteration (in the loop), if a specified condition occurs, and continues with the next iteration in the loop.

This example skips the value of 3:

EXAMPLE:

```
for (i = 0; i < 10; i++) {
   if (i === 3) { continue; }
   text += "The number is " + i + "<br>";
}
```

JAVASCRIPT LABELS:

To label JavaScript statements you precede the statements with a label name and a colon:

label:
statements

The break and the continue statements are the only JavaScript statements that can "jump out of" a code block.

SYNTAX:

break labelname;

continue labelname;

The continue statement (with or without a label reference) can only be used to skip one loop iteration.

The break statement, without a label reference, can only be used to jump out of a loop or a switch.

With a label reference, the break statement can be used to jump out of any code block.

EXAMPLE:

```
var cars = ["BMW", "Volvo", "Saab", "Ford"];
list: {
   text += cars[0] + "<br>";
   text += cars[1] + "<br>";
   text += cars[2] + "<br>";
   break list;
   text += cars[3] + "<br>";
   text += cars[4] + "<br>";
   text += cars[5] + "<br>";
}
```

PROGRAM 1: DISPLAY A TIME-BASED GREETING USING ELSE IF.

```
<!DOCTYPE html>
<html>
<body>
<p>Click the button to get a time-based greeting:</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
```

```
<script>
function myFunction() {
   var greeting;
   var time = new Date().getHours();
   if (time < 10) {
      greeting = "Good morning";
   } else if (time < 20) {
      greeting = "Good day";
   } else {
      greeting = "Good evening";
   }
document.getElementById("demo").innerHTML = greeting;
}
</script>
</body>
```

OUTPUT:

PROGRAM 2: USE OF SWITCH STATEMENT.

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>

<script>
var day;
switch (new Date().getDay()) {
   case 0:
      day = "Sunday";
      break;
   case 1:
      day = "Monday";
      break;
   case 2:
      day = "Tuesday";
      break;
case 3:
        day = "Wednesday";
        break;
    case 4:
        day = "Thursday";
        break;
    case 5:
        day = "Friday";
        break;
    default:
        day = "Saturday";
}
document.getElementById("demo").innerHTML = "Today is " + day;
</script>
</body>
```

OUTPUT:

PROGRAM 3: USE OF SWITCH STATEMENT.

```
<html>
<body>
<p id="demo"></p>

<script>
var text;
switch (new Date().getDay()) {
    case 1:
    case 2:
    case 3:
        text = "Looking forward to the Weekend";
        break;
    case 4:
    case 5:
        text = "Soon it is Weekend";
        break;
    case 0:
    case 6:
```

```
    text = "It is Weekend";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
```

OUTPUT:

PROGRAM 4: USE OF FOR LOOP.

```
<!DOCTYPE html>
<html>
<body>
<p>Click the button to loop through a block of code five times.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
  var text = "";
  var i;
  for (i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
  }
  document.getElementById("demo").innerHTML = text;
}
</script>
</body>
```

OUTPUT:

PROGRAM 5: USE OF WHILE LOOP.

```
<!DOCTYPE html>
<html>
<body>
<p>Click the button to loop through a block of code as long as i is less than 10.</p>

<button onclick="myFunction()">Try it</button>

<p id="demo"></p>

<script>
function myFunction() {
    var text = "";
    var i = 0;
    while (i < 10) {
        text += "<br>The number is " + i;
        i++;
    }
    document.getElementById("demo").innerHTML = text;
}
</script>
</body>
```

OUTPUT:

PROGRAM 6: USE OF DO WHILE LOOP.

```
<!DOCTYPE html>
<html>
<body>

<p>Click the button to loop through a block of code as long as i is less than 10.</p>

<button onclick="myFunction()">Try it</button>
```

```
<p id="demo"></p>

<script>
function myFunction() {
   var text = ""
   var i = 0;
   do {
      text += "<br>The number is " + i;
      i++;
   }
   while (i < 10)
   document.getElementById("demo").innerHTML = text;
}
</script>
</body>
```

OUTPUT:

PROGRAM 7: USE OF BREAK STATEMENT.

```
<!DOCTYPE html>
<html>
<body>

<p>A loop with a break.</p>

<p id="demo"></p>

<script>
var text = "";
var i;
for (i = 0; i < 10; i++) {
   if (i === 3) { break; }
   text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
```

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

OUTPUT:

PROGRAM 8: USE OF CONTINUE STATEMENT.

```
<!DOCTYPE html>
<html>
<body>

<p>A loop which will skip the step where i = 3.</p>

<p id="demo"></p>

<script>
var text = "";
var i;
for (i = 0; i < 10; i++) {
   if (i === 3) { continue; }
   text += "The number is " + i + "<br>";
}
document.getElementById("demo").innerHTML = text;
</script>
</body>
```

OUTPUT: