K. J. Somaiya Polytechnic, Mumbai-77

Batch No: C1

Enrollment No.: FCOW19118

Experiment No: 8

Experiment Name: Implement program based on decision and looping control structure in vbscript

K. J. Somaiya Polytechnic, Mumbai-77

Experiment No.: 08

Experiment Name: Implement program based on decision and looping control structure in vbscript

Objective: **O18RA63.5 Create web pages using different constructs ,cookies and form processing in ASP**

Theory:

New concept-

A] Decision control structure:

- Decision making allows programmers to control the execution flow of a script or one of its sections. The execution is governed by one or more conditional statements.
- VBScript provides following types of decision making statements.

1. <u>If ... Then statement</u>

An **If** statement consists of a Boolean expression followed by one or more statements. If the condition is said to be True, the statements under **If** condition(s) are Executed. If the Condition is said to be False, the statements after the **If** loop are executed.

Syntax:

The syntax of an **If** statement in VBScript is:

```
If (boolean_expression)  Then
   Statement 1
  Statement n
End If
```

2. If ... Then Else statement

- An **If** statement consists of a Boolean expression followed by one or more statements.
- If the condition is said to be True, the statements under **If** condition(s) are Executed. If the Condition is said to be False, the statements under **Else** Part would be executed.

Syntax:

The syntax of an if statement in VBScript is:

```
If (boolean_expression) Then
    Statement 1
        ...
    Statement n
Else
    Statement 1
        ....
    Statement n
End If
```

3.If ... Elseif ... Else statement

- An **If** statement followed by one or more **ElseIf** Statements that consists of Boolean expressions and then followed by a default **else** statement, which executes when all the condition becomes false.

Syntax:

The syntax of an If-ElseIf-Else statement in VBScript is:

```
If (boolean_expression) Then
  Statement 1
  .....
  Statement n
ElseIf (boolean_expression) Then
  Statement 1
   ....
  Statement n
ElseIf (boolean_expression) Then
  Statement 1
   ....
```

```
   Statement n
Else
   Statement 1
    ....
   Statement n
End If
```

3. Switch statement in vbscript

- When a User want to execute a group of statements depending upon a value of an Expression, then Switch Case is used. Each value is called a **Case**, and the variable being switched **ON** based on each case.
- **Case Else** statement is executed if test expression doesn't match any of the Case specified by the user.
- **Case Else** is an optional statement within Select Case, however, it is a good programming practice to always have a Case Else statement.

Syntax:

The syntax of a Switch Statement in VBScript is:

```
Select Case expression
  Case expressionlist1
     statement1
     statement2
     statement1n
  Case expressionlist2
     statement1
     statement2
 Case expressionlistn
     statement1
     statement2
 Case Else
     elsestatement1
     elsestatement2
     ....
End Select
```

B] Looping construct

- There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.
- Programming languages provide various control structures that allow for more complicated execution paths.
- A loop statement allows us to execute a statement or group of statements multiple times

1. For next loop

- A **for** loop is a repetition control structure that allows a developer to efficiently write a loop that needs to execute a specific number of times.

Syntax:

The syntax of a **for** loop in VBScript is:

```
For counter = start To end [Step stepcount]
  [statement 1]
  [statement 2]
  ....
  [statement n]
  [Exit For]
  [statement 11]
  [statement 22]
  ....
  [statement n]
Next
```

Here is the flow of control in a For Loop:

- The **For** step is executed first. This step allows you to initialize any loop control variables and increment the step counter variable.

- Secondly, the **condition** is evaluated. If it is true, the body of the loop is executed. If it is false, the body of the loop does not execute and flow of control jumps to the next statement just after the For Loop.
- After the body of the for loop executes, the flow of control jumps to the **Next** statement. This statement allows you to update any loop control variables. It is updated based on the step counter value.
- The condition is now evaluated again. If it is true, the loop executes and the process repeats itself (body of loop, then increment step, and then again condition). After the condition becomes false, the For Loop terminates.

2. For each next loop

- A **For Each** loop is used when we want to execute a statement or a group of statements for each element in an array or collection.
- A **For Each** loop is similar to For Loop; however, the loop is executed for each element in an array or group. Hence, the step counter won't exist in this type of loop and it is mostly used with arrays or used in context of File system objects in order to operate recursively.

Syntax:

The syntax of a **For Each** loop in VBScript is:

```
For Each element In Group
  [statement 1]
  [statement 2]
  ....
  [statement n]
  [Exit For]
  [statement 11]
  [statement 22]
Next
```

1. While wend loop

- In a **While..Wend** loop, if the condition is True, all statements are executed until **Wend** keyword is encountered.

- If the condition is false, the loop is exited and the control jumps to very next statement after **Wend** keyword.

Syntax

The syntax of a **While..Wend** loop in VBScript is:

```
While condition(s)
   [statements 1]
   [statements 2]
   ...
   [statements n]
Wend
```

2. Do while Loop-

- A **Do..While** loop is used when we want to repeat a set of statements as long as the condition is true. The Condition may be checked at the beginning of the loop or at the end of the loop.
- Syntax

The syntax of a **Do..While** loop in VBScript is:

```
Do While condition
   [statement 1]
   [statement 2]
   ...
   [statement n]
   [Exit Do]
   [statement 1]
   [statement 2]
   ...
   [statement n]
Loop
```

Alternate Syntax

There is also an alternate Syntax for **Do..while** loop which checks the condition at the end of the loop. The Major difference between these two syntax is explained below with an example.

```
Do
   [statement 1]
   [statement 2]
   ...
   [statement n]
   [Exit Do]
   [statement 1]
   [statement 2]
   ...
   [statement n]
Loop While condition
```

3. Do until Loop

- A **Do..Until** loop is used when we want to repeat a set of statements as long as the condition is false. The Condition may be checked at the beginning of the loop or at the end of loop.

Syntax:

The syntax of a **Do..Until** loop in VBScript is:

```
Do Until condition
   [statement 1]
   [statement 2]
   ...
   [statement n]
   [Exit Do]
   [statement 1]
   [statement 2]
   ...
   [statement n]
Loop
```

Implementation and Output

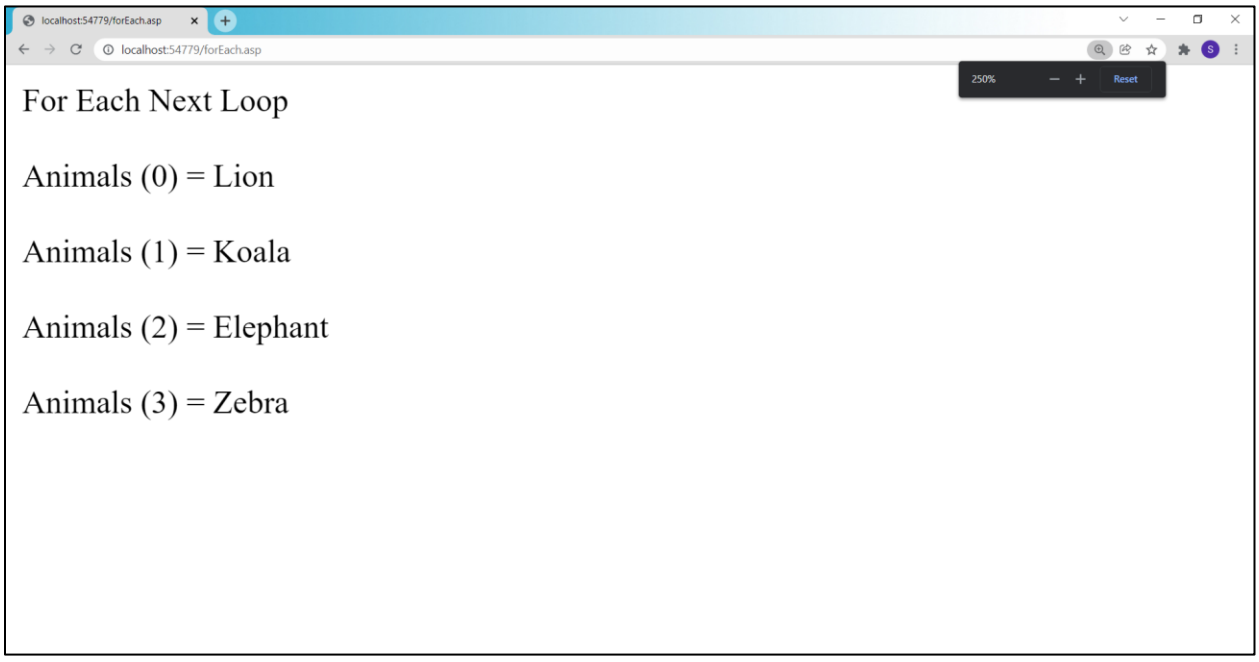
1. Write a program to display array elements using for each next loop.

Implementation:

```
<%@Language = VBScript%>
<% Option Explicit %>
<html>
<body>
   For Each Next Loop<br><br>
<%
Dim animals(3)
animals(0) = "Lion"
animals(1) = "Koala"
animals(2) = "Elephant"
animals(3) = "Zebra"

Dim x,a
x = 0
For Each a in animals
  Response.Write("Animals (" &x &") = " &a &"<br><br>")
  x = x+1
Next
%>
</body>
</html>
```

Output:

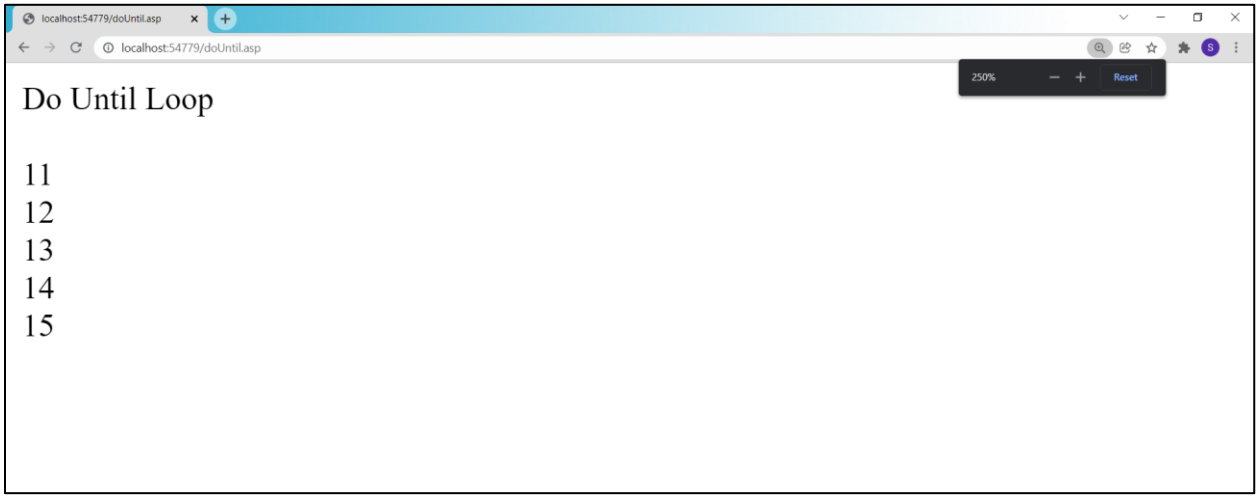


2. Write a program to display numbers from 11 to 15 using do until loop.

Implementation:

```
<%@Language = VBScript%>
<% Option Explicit %>
<html>
<body>
   Do Until Loop<br><br>
<%
Dim n
n = 11
Do Until (n > 15)
Response.Write(n &"<br>")
n = n+1
Loop
%>
</body>
</html>
```

Output:



3. Write a program to give grades to students-
Marks above 75 Distinction
Marks between 61 and 74 First class
Marks between 60 and 51 Second class
Marks between 50 and 40 Pass class
Else fail

Implementation:

```
<%@Language = VBScript%>
<% Option Explicit %>
<html>
<body>
<%
Dim marks
marks = 67
Response.Write("Marks: " &marks &"<br><br>")
If(marks >= 75) Then
Response.Write("Grade is Distinction")
ElseIf (marks >=61 AND marks<=74) Then
Response.Write("Grade is First Class")
ElseIf (marks >=51 AND marks<=60) Then
Response.Write("Grade is Second Class")
ElseIf (marks >=40 AND marks<=50) Then
Response.Write("Grade is Pass Class")
Else
Response.Write("Grade is Fail")
End If
%>
</body>
</html>
```

Output:

Conclusion: Thus, we have implemented program on decision control and looping control structure in vbscript.