**K. J. Somaiya Polytechnic, Mumbai-77**

**Batch No: C1**

**Enrollment No.: FCOW19118**

**Experiment No: 09**

**Experiment Name: Implement program based on user defined functions, subroutine and built-in functions in vbscript.**

**K. J. Somaiya Polytechnic, Mumbai-77**

**Experiment No.: 09**

**Experiment Name: Implement program based on user defined functions, subroutine and built-in functions in vbscript.**

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

---

**Theory:**

**New concept-**

In VBScript there are two kinds of procedures; the Sub procedure and the Function procedure.

**A] Sub Procedures**

- A Sub procedure is a series of VBScript statements, enclosed by Sub and End Sub statements, that perform actions but don't return a value.
- A Sub procedure can take arguments (constants, variables, or expressions that are passed by a calling procedure).
- If a Sub procedure has no arguments, its Sub statement must include an empty set of parentheses ().
- To invoke a Procedure somewhere later in the script, you would simply need to write the name of that procedure with or without the **Call** keyword.
  sub_name
- **Syntax**

```
Sub sub_name()
Block of code
End Sub
```

## B] Function

- If you want to execute a series of statements and return a value, then you need to use function procedures, commonly known as function.
- Function procedures start and end with **Function** and **End Function** statements respectively.
- In the body of the function, describe what it is supposed to do. To return the right value, assign the desired value to the name of the function.
- To call a function, we have two main alternatives. If we want to use the return value of a function in an event or another function, assign the name of the function to the appropriate local variable.

```
Function function_name(parameter)

    Code block

    function_name = return_value
End Function
```

## A] Conversion Function

| Function | Description |
|---|---|
| **Asc** | Converts the first letter in a string to ANSI code |
| **CBool** | Converts an expression to a variant of subtype Boolean |
| **CByte** | Converts an expression to a variant of subtype Byte |
| **CCur** | Converts an expression to a variant of subtype Currency |
| **CDate** | Converts a valid date and time expression to the variant of subtype Date |
| **CDbl** | Converts an expression to a variant of subtype Double |
| **Chr** | Converts the specified ANSI code to a character |
| **CInt** | Converts an expression to a variant of subtype Integer |
| **CLng** | Converts an expression to a variant of subtype Long |
| **CSng** | Converts an expression to a variant of subtype Single |
| **CStr** | Converts an expression to a variant of subtype String |
| **Hex** | Returns the hexadecimal value of a specified number |
| **Oct** | Returns the octal value of a specified number |

**B] String Functions**

| Function | Description |
|---|---|
| **InStr** | Returns the position of the first occurrence of one string within another. The search begins at the first character of the string |
| **InStrRev** | Returns the position of the first occurrence of one string within another. The search begins at the last character of the string |
| **LCase** | Converts a specified string to lowercase |
| **Left** | Returns a specified number of characters from the left side of a string |
| **Len** | Returns the number of characters in a string |
| **LTrim** | Removes spaces on the left side of a string |
| **RTrim** | Removes spaces on the right side of a string |
| **Trim** | Removes spaces on both the left and the right side of a string |
| **Mid** | Returns a specified number of characters from a string |
| **Replace** | Replaces a specified part of a string with another string a specified number of times |
| **Right** | Returns a specified number of characters from the right side of a string |
| **Space** | Returns a string that consists of a specified number of spaces |
| **StrComp** | Compares two strings and returns a value that represents the result of the comparison |
| **String** | Returns a string that contains a repeating character of a specified length |
| **StrReverse** | Reverses a string |
| **UCase** | Converts a specified string to uppercase |

## C] Arithmetic function

| Function | Description |
|---|---|
| **Abs** | Returns the absolute value of a specified number |
| **Atn** | Returns the arctangent of a specified number |
| **Cos** | Returns the cosine of a specified number (angle) |
| **Exp** | Returns *e* raised to a power |
| **Hex** | Returns the hexadecimal value of a specified number |
| **Int** | Returns the integer part of a specified number |
| **Fix** | Returns the integer part of a specified number |
| **Log** | Returns the natural logarithm of a specified number |
| **Oct** | Returns the octal value of a specified number |
| **Rnd** | Returns a random number less than 1 but greater or equal to 0 |
| **Sgn** | Returns an integer that indicates the sign of a specified number |
| **Sin** | Returns the sine of a specified number (angle) |
| **Sqr** | Returns the square root of a specified number |
| **Tan** | Returns the tangent of a specified number (angle) |

**Questions and answers:**

1. **Give the difference between function and sub procedure?**

| Function | Sub procedure |
|---|---|
| A function is a procedure that enclosed by the Function and End Function statements | A Sub procedure is a block of Visual Basic statements enclosed by a declaration statement and a matching End declaration. |
| A function helps to perform a contain task | Sub procedure helps to make the code readable, easy to modify and debug. |
| A function is a specific type of procedure | A Sub procedure is a generalized type of function. |
| Syntax:<br>Sub subname(argument1,argument2)<br>some statements<br>End Sub | Syntax:<br>Function Functionname(parameter-list)<br>statement 1<br>statement 2<br>statement n<br>End Function |

## 2. How to return value in function?

A function can only return one value. In the function definition you assign the value to function name. You can use "Exit Function" statements to make the function return with different values at different points.

For example:

```
Function Example (strValue)
  If (strValue = "A") Then
    Example = 1
    Exit Function
  End If
  If (strValue = "B") Then
    Example = 2
    Exit Function
  End If
  Example = 3
End Function
```

Although there is only one function value, the function can alter any of the parameters passed to the function (if they are passed ByRef rather than ByVal), and the function could also modify global variables.

**Implementation**

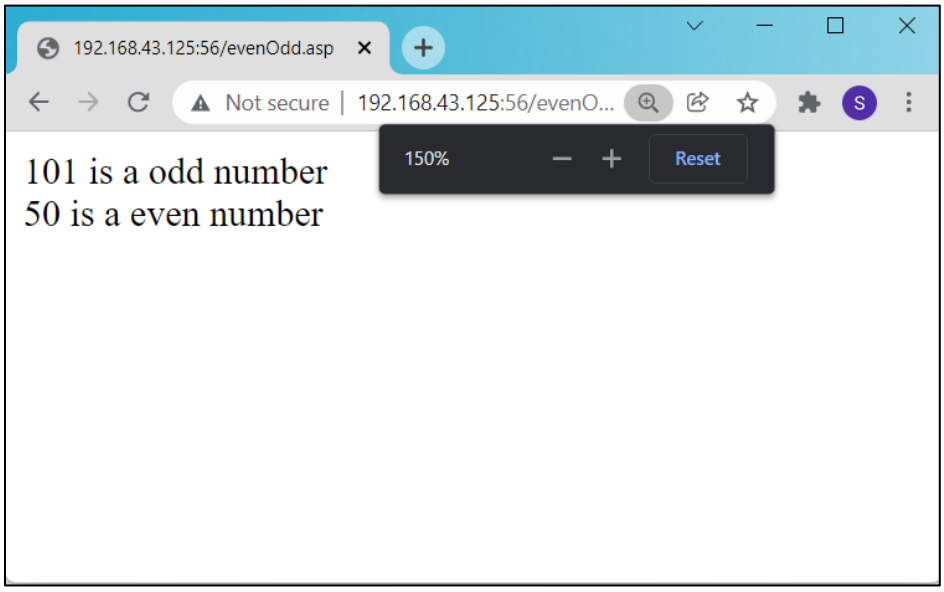
**1. Write a program using sub procedure to check whether number is even or odd?**

**Program:**

```
<!DOCTYPE html>
<html>
<body>
<%
 Sub printMessage(num)
  If (num Mod 2 = 0) Then
   Response.Write(num& " is a even number" &"<br \>")
  Else
   Response.Write(num& " is a odd number" &"<br \>")
  End If
 End Sub
 Call printMessage(101)
 Call printMessage(50)
%>
```

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

**Output:**

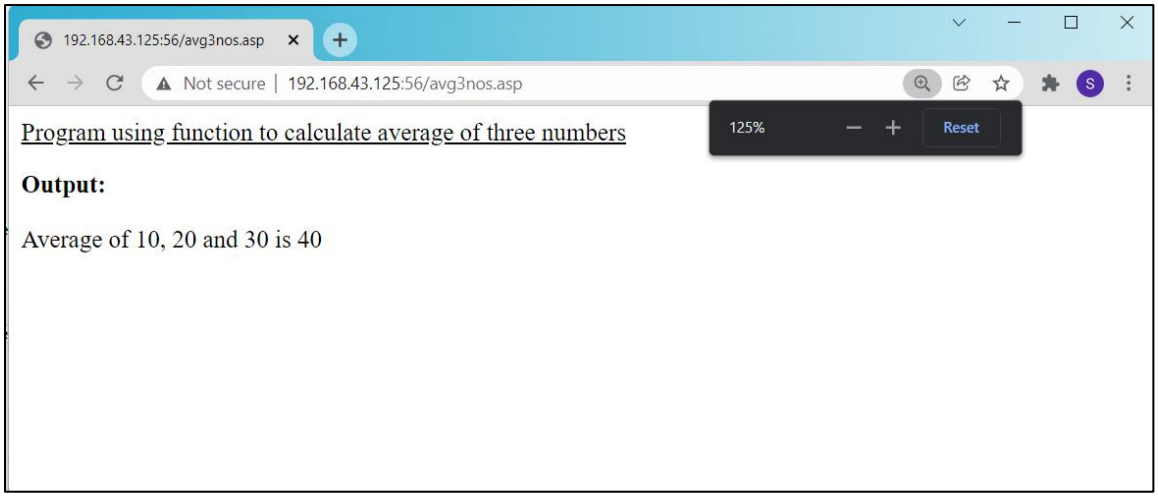



2. **Write a program using function to calculate average of three numbers?**

**Program:**

```
<%@ Language=VBScript %>
<% Option Explicit %>
<html>
<head>
<%
 Function Avg(a,b,c)
  Dim d
  d=a+b+c/3
  Avg=d
 End Function
%>
</head>
<body>
<%
 Function Message()
  Response.Write("<b>Output:</b><br> <br>")
 End Function
%>
<p><u>Program using function to calculate average of three numbers</u></p>
<%
 Message
 Dim res
 res=Avg(10,20,30)
 Response.Write("Average of 10, 20 and 30 is " &res)
%>
</body>
</html>
```

## Output:




## 3. Write a program to compare two strings and search particular string in string.

## Program:

```
<!DOCTYPE html>
<html>
<body>
<%
 Function compareString(a,b)
  IF (a = b) Then
   Response.Write("Both strings are equal")
  Else
   Response.Write("Both strings are unequal")
  End IF
  Response.Write("<br></br>")
 End Function
 Function searchText(statement, target)
  value = InStr(statement, target)
  IF (value = 0) Then
   Response.Write("String " &statement& " doesn't contain value " &target)
  Else
   Response.Write("String " &statement& " contain value " &target& " at position "
&value)
  End IF
  Response.Write("<br></br>")
 End Function
 Call compareString("Hiteshi","Hit")
 Call searchText("Apple","App")
%>
</body>
</html>
```

**Output:**


**Conclusion- Thus we have implemented program based on user defined functions, subroutine and built-in functions in vbscript.**