**VB SCRIPT FUNCTIONS**

VBScript functions are self contained, named blocks of code that perform some function or task.

VBScript functions allow us to reuse our code again and again. For example we can write a function that performs the concatenation of two strings. After writing the function whenever we need the functionality of the concatenation of two strings, we can **CALL** our string concatenation function.

VBScript functions are defined using the **Function**and **End Function**keywords. The code that makes up the function is written between these two keywords.

Here are the steps to write VBScript functions:

1. Mark the start of the function with the Function keyword.
2. Give your function a meaningful name.
3. Follow the function name with a pair of round brackets.
4. If the function is expecting external information to perform its task, add the name(s) for the external information, also called arguments. This is optional.
5. Starting on a new line, write the code that makes up the function.
6. Finish with the End Function keywords.

Here is a simple example that requires two numbers as arguments and sums them both:

Function add(num1, num2)  
add = num1 + num2  
End Function

**Calling VBScript Functions**  
A function doesn't do anything until it is **called**. Calling a function in VBScript is very easy. We call a function by writing its name followed by a pair of brackets. Optionally, if the function requires external information, we have to pass it between the pair of brackets.

Here is a simple Example of calling our add function, above: add (10, 5)

Here is an example:

Function add(num1, num2)  
add = num1 + num2  
End Function

Dim total  
total = add(10, 5)  
Msgbox "The total of 10 and 5 is: " & total

**Vb script Message box**

Often in our VBScript code we need to share some information with the visitor of our website. Sometimes we need to greet or welcome the visitor.   
  
VBScript MsgBox (Message Box) allows us to pupup a message box, which we can use to greet our visitor or share some information with him.

**Creation of a Message Box** in VBScript is very easy; we only need to supply the MsgBox function with a string.   
  
Here is an example that pops a Message Box with the words "Hello World!":

Dim h  
Dim w  
h = "Hello "  
w = "World!"  
MsgBox h & w

**Function Parameters:**

Till now, we have seen function without a parameter, but there is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. The Functions are called using the **Call** Keyword.

Function sayHello(name, age)

msgbox( name & " is " & age & " years old.")

End Function

Call sayHello("Talentsprint", 7)

**Sub Procedures :**

Sub Procedures are similar to functions but there are few differences.

* Sub procedures DONOT Return a value while functions may or may not return a value.
* Sub procedures Can be called without call keyword.
* Sub procedures are always enclosed within **Sub** and **End Sub** statements.

Eg

Sub sayHello()

msgbox ("Hello there")

End Sub

## Calling Procedures:

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 sayHello()

msgbox("Hello there")

End Sub

sayHello()

## Advanced Concepts for Functions

1. Byval Parameters
2. BYRef Parameters

**ByVal:**

If ByVal is specified, then the arguments are sent as by value when the function or procedure is called.

Function fnadd(Byval num1, Byval num2)

num1 = 4

num2 = 5

End Function

Dim x ,y

x=6

y=4

res= fnadd(x,y)

Msgbox “The value of x is " & x

Msgbox "The value of y is " & y

**By Ref**

If **ByRef** is specified, then the arguments are sent as a reference when the function or procedure is called.

Function fnadd(ByRef num1, ByRef num2)

num1 = 4

num2 = 5

End Function

Dim x,y

x=6

y=4

res= fnadd(x,y)

Msgbox “The value of x is " & x

Msgbox "The value of y is " & y

These are built-in VBScript functions. All you have to do is pass proper values to these function and they will respond as they built for.