

# Anatomy of a Procedure

If you are building a procedure and you type in the parameters like this...

```
Sub AddNumbers (num1 as Integer, num2 as Integer)
```

and then you hit the 'enter' key, you get this...

```
Sub AddNumbers (ByVal num1 As Integer, ByVal num2 As Integer)  
End Sub
```

The 'By Val' syntax is added automatically as is 'End Sub'

Keyword 'Sub'

Parameters

```
Sub AddNumbers (ByVal num1 As Integer, ByVal num2 As Integer)
```

```
End Sub
```

Signature

Sub name

The first line of the Procedure or Sub is known as the signature and it includes the syntax 'Sub' followed by the sub name and then any parameters.

# Matching Parameters with Arguments

A procedure containing two integer variables as parameters is called by a procedure call providing 2 integers

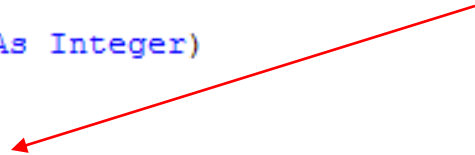
```
Private Sub btnCallSub_Click(ByVal sender As  
    AddNumbers(4, 5)  
End Sub
```

Procedure Call



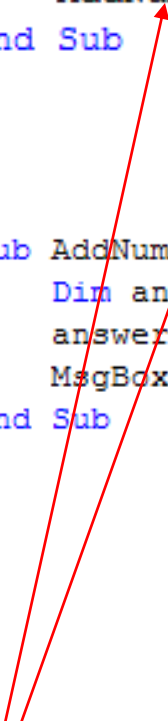
```
Sub AddNumbers(ByVal num1 As Integer, ByVal num2 As Integer)  
    Dim answer As Integer  
    answer = num1 + num2  
    MsgBox(answer)  
End Sub
```

Procedure



Number, type and order of parameters of procedure call must match those of procedure.

Names of procedure must match with procedure call



# Using Variables As Parameters in Procedure Call

In the previous slide, the parameters in the procedure call were 2 integer literals. In this case we prompt the user for the parameters and assign them to variables which are then provided in the procedure call.

```
Sub AddNumbers(ByVal num1 As Integer, ByVal num2 As Integer)
    Dim answer As Integer
    answer = num1 + num2
    MsgBox(answer)
End Sub
```

---

```
Private Sub btnCallSub_Click(ByVal sender As System.Object, ByVal
    Dim n1 As Integer
    Dim n2 As Integer
    n1 = Val(TextBox("Enter 1st number"))
    n2 = Val(TextBox("Enter 2nd number"))
    AddNumbers(n1, n2)
End Sub
```

**Note that the name of the parameters in the Procedure call are not the same as the names in Procedure signature. This is perfectly acceptable since the number, type and order of the parameters match up.**

# Variations in Parameter Types

```
Sub AddNumbers(ByVal num1 As Double, ByVal num2 As Integer)
    Dim answer As Double
    answer = num1 + num2
    MsgBox(answer)
End Sub
```

---

```
Private Sub btnCallSub_Click(ByVal sender As System.Object, ByVal e As System
    Dim n1 As Integer
    Dim n2 As Double
    n1 = Val(TextBox("Enter 1st number, this can be a decimal"))
    n2 = Val(TextBox("Enter 2nd number, this must be an integer"))
    AddNumbers(n1, n2)
End Sub
```

# Functions: Procedures That Return A Value

Many Procedures, not only accept information in the form of parameters but also return values. These special Procedures are called Functions.

Functions require extra components in their signatures.

**The keyword 'Function' tells us that this is a function (wow!)**

```
Function ReturnHighNumber(ByVal num1, ByVal num2) As Integer  
  
    Dim high As Integer  
  
    If (num1 > num2) Then  
        high = num1  
    Else  
        high = num2  
    End If  
  
    Return high  
  
End Function
```

**The type of value that will be returned by the function is set here. It is necessary that the variable type in the signature match the variable type of the return statement. In this example 'high' is indeed an integer variable.**

# Function Call

```
highNum = ReturnHighNumber(num1, num2)
```

Here is the function call for the function ReturnHighNumber that was demonstrated in the previous slide.

Since it is a function and by definition it will return a value, it needs to be assigned to a variable that will hold the value returned by the function.

ReturnHighNumber has a return type of integer. This means the variable that ReturnHighNumber is assigned to, must be an integer variable as well.

```
Dim num1 As Integer
Dim num2 As Integer
Dim highNum As Integer

num1 = Val(InputBox("Enter a number 1-100"))
num2 = Val(InputBox("Enter a 2nd number 1-100"))

highNum = ReturnHighNumber(num1, num2)

MsgBox("The high number is " & highNum)
```

# Tracing the Program Flow

```
Private Sub btnEnterNumbers_Click(ByVal sender As System.Object,  
  
    Dim num1 As Integer  
    Dim num2 As Integer  
    Dim highNum As Integer  
  
    num1 = Val(TextBox("Enter a number 1-100"))  
    num2 = Val(TextBox("Enter a 2nd number 1-100"))  
  
    highNum = ReturnHighNumber(num1, num2)  
  
    MsgBox("The high number is " & highNum)  
  
End Sub
```

```
Function ReturnHighNumber(ByVal num1, ByVal num2) As Integer  
  
    Dim high As Integer  
  
    If (num1 > num2) Then  
        high = num1  
    Else  
        high = num2  
    End If  
  
    Return high  
  
End Function
```

User enters value of 1<sup>st</sup> number into an Input Box which assign the value to num1

User enters value of 2nd number into an Input Box which assign the value to num2

The function 'ReturnHighNumber' is called with the integer variable parameters num1 and num2 provided.

If statement compares two values and returns the higher of the two.

Flow of program returns to the point at which function was called