

String Functions on Excel Macros

The word "string" is used to describe the combination of one or more characters in an orderly manner. In excel vba, variables can be declared as **String** or the **Variant** data types. However, this section will be used to explain how to use string functions for various assignments in Excel vba.

The following table presents various actions that can be performed using string functions. The corresponding language element shows how they can be declared and manipulated for a specific function.

Table showing String actions and their corresponding declarations

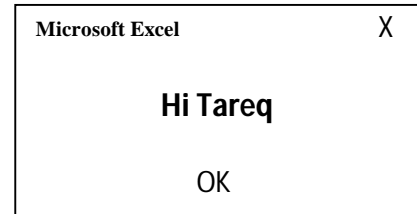
Action	Language element
Compare two strings.	StrComp
Convert strings.	StrConv
Reverse a string.	InStrRev, StrReverse
Convert to lowercase or uppercase.	Format, LCase, UCase
Create a string of repeating characters.	Space, StrDup
Find the length of a string.	Len
Format a string.	Format, FormatCurrency, FormatDate, FormatTime, FormatNumber, FormatPercent
Manipulate strings.	InStr, Left, LTrim, Mid, Right, RTrim, Trim
Set string comparison rules.	Option Compare
Work with ASCII and ANSI values.	Asc, AscW, Chr, ChrW
Replace a specified substring.	Replace
Return a filter-based string array.	Filter
Return a specified number of substrings.	Split, Join

Example (Return specific number of substrings - *Join*)

Suppose two separate text messages "Hi" and "Tareq" is to be displayed together on the message box. This can be done by employing the use of operator **&**.

The use of operator **&** by executing this code below;

```
Sub variable ( )  
    Dim Text1 As String  
    Dim Text2 As String  
    Text1 = "Hi"  
    Text2 = "Tareq"  
    MsgBox Text1 & " " & Text2
```



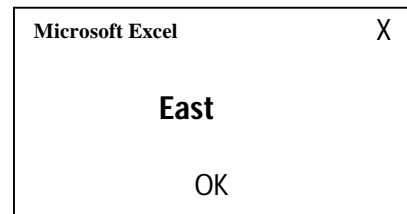
Do not forget that " " is inserted to create space between Text1 and Text2

After running the code, the pop-up message will look like;

Example (Manipulate strings by *Left*)

This can be used to extract the left-part character from a particular string. Here the number of characters to be extracted is included in the code as illustrated below

```
Sub Variables( )  
    Dim Text As String  
    Text = "Eastern Mediterranean"  
    MsgBox Left(Text, 4)  
End Sub
```



The result of the execution of the code will be as shown below.

Similarly, it may be desired to extract the right-part characters from a given string. Right function is used to perform this task. Always remember to indicate the number of the characters to be extracted from the right hand-side.

```
Sub Variables( )  
    Dim Text As String  
    Text = "Eastern Mediterranean"  
    MsgBox Right(Text, 13)  
End Sub
```



Example (Extracting the middle substring from a string)

Another way to manipulate a string could be to extract certain middle substrings out of the whole given string. Function **Mid** can be executed to perform this task. Here there is need to specify the number of the *starting character* and number for *as long*.

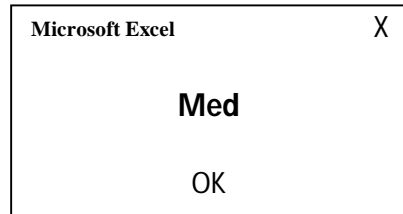
```
Sub Variables ( )
```

```
Dim Text As String
```

```
Text = "Eastern Mediterranean"
```

```
MsgBox Mid(Text, 9, 3)
```

```
End Sub
```



Example (To get the length of a string using Len function)

It should be noted that the length of any two strings joined together include the space between them. *Len* will count the space (s) between the texts as part of their length.

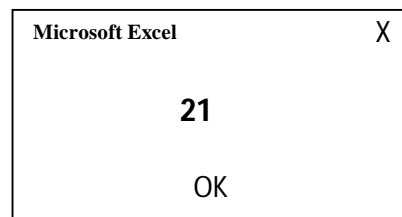
```
Sub Variables ( )
```

```
Dim Text As String
```

```
Text = "Eastern Mediterranean"
```

```
MsgBox Len (Text)
```

```
End Sub
```



Example (To find the position of any substring in a string using Instr function)

The particular substring whose position is sought within the string must be stated in the code. Care need to be taken in case the desired character is more than one in a given string. The position of "M" sought for in the example is 9 because there is only one "M" in the string and it is in uppercase. Imagine what would occur if M is more than one in the string. For instance, if we need to find the position of "e", there are 4 characters with "e" that is in lowercase. You need to create a unique identifier for the character "e" anticipated. If you put "e", the code will return the "e" in the "Eastern" as 5 being the first "e". If you input "er" when the "e" after "t" is desired same e is returned as 5 because there it is the first "er" in the string. So the best *identifier* for the "e" after "t" in the string is "err" which 14th position and for "e" after "n" is "ea" which is of 19th position.

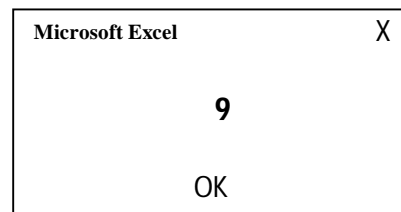
```
Sub Variables ( )
```

```
Dim Text As String
```

```
Text = "Eastern Mediterranean"
```

```
MsgBox Instr (Text, "Med")
```

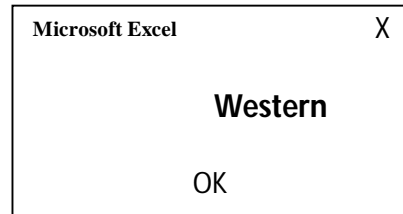
```
End Sub
```



Example (To replace part or substring in the string with another different substring using *Replace function*)

Suppose we want to change "Eastern" in "Eastern Mediterranean" to "Western" without writing the entire string, we need to specify the particular substring and its corresponding replacement in the code. The below example illustrates how it can be done.

```
Sub Variables ( )  
Dim Text As String  
Text = "Eastern Mediterranean"  
MsgBox Replace("Eastern", "Eas", "Wes")  
End Sub
```



Example (To convert uppercase string to lowercase using *LCase function*)

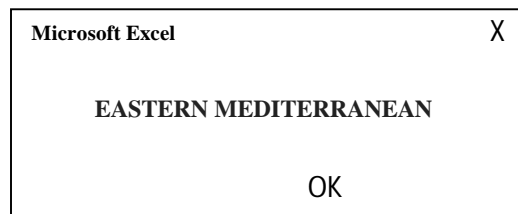
Suppose we have a string in uppercase

```
Sub Variables( )  
Dim UpperCase As String  
Dim LowerCase As String  
UpperCase = "EASTERN MEDITERRANEAN"  
LowerCase = LCase(UpperCase)  
MsgBox LowerCase  
End Sub
```



Similarly, *UCase function* can be declared to convert lowercase string to uppercase

```
Sub Variables( )  
Dim UpperCase As String  
Dim LowerCase As String  
LowerCase = "eastern mediterranean"  
UpperCase = UCase(LowerCase)  
MsgBox LowerCase  
End Sub
```



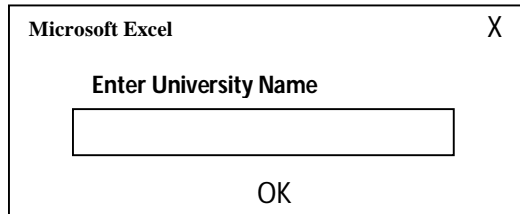
Example (Creating user input box)

InputBox function in Excel vba can be used to prompt **users** to enter a their custom string. Instead of entering "Eastern Mediterranean" as we have in the examples, an inputbox can be created so that any string as desired by any user can be entered. Inputbox function is inserted into the general code for the task. A command prompt window is displayed on the worksheet through which the user can enter their specific string. For example, to display the name of any university with a particular student name, two input boxes are needed. The first box, the user enters the name of their university, and with the second input box the user enters their full name. Using the below code;

Sub Variables ()

```
Dim University As String  
Dim StudentName As String  
University = InputBox("Enter University Name")  
StudentName = InputBox("Enter your Full Name")  
MsgBox (University & " " & vbCrLf & "My Name is" & " " & StudentName)
```

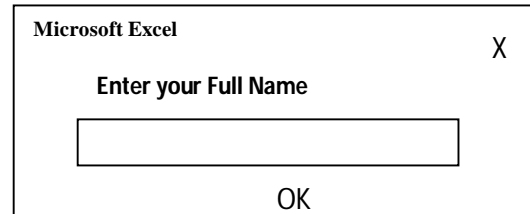
End Sub



Microsoft Excel X

Enter University Name

OK

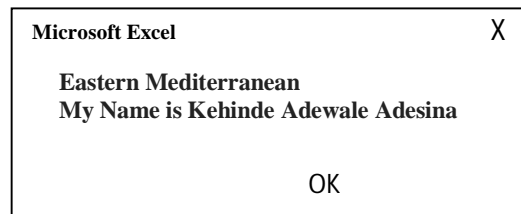


Microsoft Excel X

Enter your Full Name

OK

Let us assume that the user entered University Name as Eastern Mediterranean and Full name as Kehinde Adewale Adesina, the final window that will pop-up will look like this;



Microsoft Excel X

Eastern Mediterranean
My Name is Kehinde Adewale Adesina

OK