Visual Basic for Applications
Programming

Damiano SOMENZI

School of Economics and Management

Advanced Computer Skills

damiano.somenzi@unibz.it

Week 7

Outline

Outline

MsgBox Function

MsgBox

MsgBox function displays a *message* in a dialog box and waits for the user to click a button. We look at the following syntax as reference

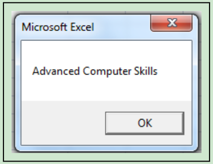
```
MsgBox(prompt)
```

- **prompt** (required), is the string expression displayed as the message in the dialog box
- if prompt consists of more than one line you can separate the lines using a carriage return character **Chr(13)**

MsgBox Function

Examples

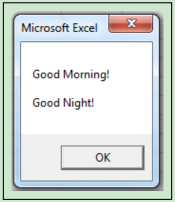
```
MsgBox("Advanced Computer Skills")
```



Microsoft Excel

Advanced Computer Skills

```
MsgBox ("Good Morning!" & Chr(13) & Chr(13) & "Good Night!")
```



Microsoft Excel

Good Morning!

Good Night!

Outline

Input Function

InputBox

InputBox function displays a **prompt** in a dialog box, waits for the user to input **text** or **click a button (OK)**, and returns a **String** containing the contents of the text box. We look at the following syntax as reference

```
InputBox(prompt)
```

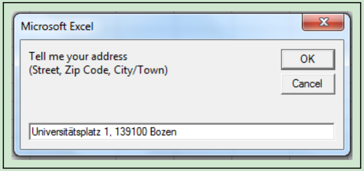
- **prompt** (required), is the string expression displayed as the message in the dialog box
- if prompt consists of more than one line you can separate the lines using a carriage return character **Chr(13)**
- if the user clicks OK or presses ENTER, the InputBox function returns whatever is in the text box, even the empty string ("")
- if text represents a numeric value the **Val** function is able to return the numbers contained in the input **string** as a numeric value of appropriate type. We look at the following syntax as reference

```
Val(InputBox(prompt))
```

InputBox Function

Examples

```
address = InputBox("Tell me your address" & Chr(13) & "(Street,
                    Zip Code, City/Town)")
```

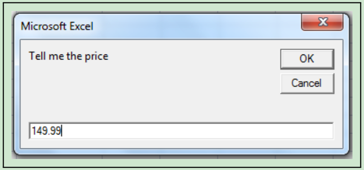


Microsoft Excel

Tell me your address
(Street, Zip Code, City/Town)

Universitätsplatz 1, 139100 Bozen

```
price = Val (InputBox ("Tell me the price"))
```



Microsoft Excel

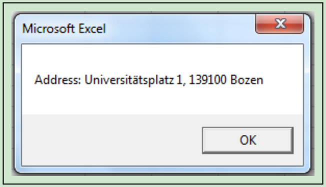
Tell me the price

149.99

Input Box - MsgBox Functions

Examples

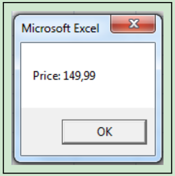
```
MsgBox("Address:  " & address)
```



Microsoft Excel

Address: Universitätsplatz 1, 139100 Bozen

```
MsgBox ("Price:  " & price)
```



Microsoft Excel

Price: 149,99

Outline

Input Validation

Input Validation

The following examples provide some programming solution that could be adopted to validate input data, for example when a subset of data is suitable for the computation

```
Sub validateInput()

    'it asks for a word until the typed word is different from the empty word

    Dim s As String
    Do
        s = InputBox("Tell me a word")
    Loop While s = ""
    MsgBox (s & " -> this is your word!")
End Sub
```

Input Validation

Input Validation

```
Sub sumIntegers()

    'it computes the sum of positive integers (> 0) typed by the user
    'any set of positive integers is suitable for the computation
    'the computation stops when is typed a non positive integer (<= 0)

    Dim n As Integer
    Dim sum As Integer
    sum = 0
    n = Val(InputBox("Type a positive integer"))
    Do While n > 0
        sum = sum + n
        n = Val(InputBox("Type a positive integer"))
    Loop
    MsgBox ("Sum - > " & sum)
End Sub
```

Input Validation

Input Validation

```
Sub wordConc()

    'it creates a new sentence concatenating a set of words typed by the user
    'the words starting with the letter "b" are accepted
    'the computation stops when it is returned an empty word

    Dim w As String
    Dim s As String
    s = ""
    w = InputBox("Type a word")
    Do While w <> ""
        If UCase(Left(w, 1)) = "B" Then
            s = s & w & " "
        End If
        w = InputBox("Type a word")
    Loop
    MsgBox (s)
End Sub
```

Input Validation

Input Validation

```
Sub maxLenWords()

    'it determines the maximum length word among a set of words typed by the user
    'it is accepted the first occurrence of two words of equal length
    'the computation stops when it is returned an empty word

    Dim w As String
    Dim max As String
    max = ""
    w = InputBox("Type the word!")
    Do While w <> ""
        If Len(w) > Len(max) Then
            max = w
        End If
        w = InputBox("Type the word!")
    Loop
    MsgBox ("word -> " & max & ", length -> " & Len(max))
End Sub
```

Input Validation

Input Validation

```
Sub avgLenWords()

    'it determines the average length word among a set of words typed by the user
    'the computation stops when it is returned an empty word

    Dim w As String
    Dim char As Integer, no As Integer
    char = 0
    no = 0
    w = InputBox("Type the word!")
    Do While w <> ""
        char = char + Len(w)
        no = no + 1
        w = InputBox("Type the word!")
    Loop
    MsgBox ("Average length -> " & Round((char / no), 2))
End Sub
```

Input Validation

Input Validation

```
Sub avgIntegers()

    'it determines the average of a set of positive integers (> 0) typed by the user
    'any set of positive integers is suitable for the computation
    'the computation stops when is typed a non positive integer (<= 0)

    Dim n As Integer
    Dim sum As Integer, count As Integer
    sum = 0
    count = 0
    n = Val(InputBox("Type a positive number"))
    Do While n > 0
        sum = sum + n
        count = count + 1
        n = Val(InputBox("Type a positive number"))
    Loop
    If count = 0 Then
        MsgBox ("No Numbers")
    Else
        MsgBox ("Average - > " & sum / count)
    End If
End Sub
```

Exercise

For a set of positive integer numbers (> 0) typed by the user, we need a sub procedure that counts and displays how many of the typed numbers belong to the classes

1. 1–10
2. 11–20
3. 21–30

Numbers outside those classes are not valid, but counted as well. The user could type any set of integer numbers. The sub procedure stops the computation when is typed a non positive integer ($<= 0$)

Exercises

```
Sub classes()
    Dim n As Integer
    Dim c1 As Integer, c2 As Integer, c3 As Integer
    Dim nv As Integer
    c1 = 0
    c2 = 0
    c3 = 0
    nv = 0
    n = Val(InputBox("Type a positive integer"))
    Do While n > 0
        Select Case n
        Case 1 To 10
            c1 = c1 + 1
        Case 11 To 20
            c2 = c2 + 1
        Case 21 To 30
            c3 = c3 + 1
        Case Else
            nv = nv + 1
        End Select
        n = Val(InputBox("Type a positive integer"))
    Loop
    MsgBox ("[1-10]: " & c1 & "[11-20]: " & c2 & "[21-30]: " & c3 & "Not Valid: " & nv)
End Sub
```