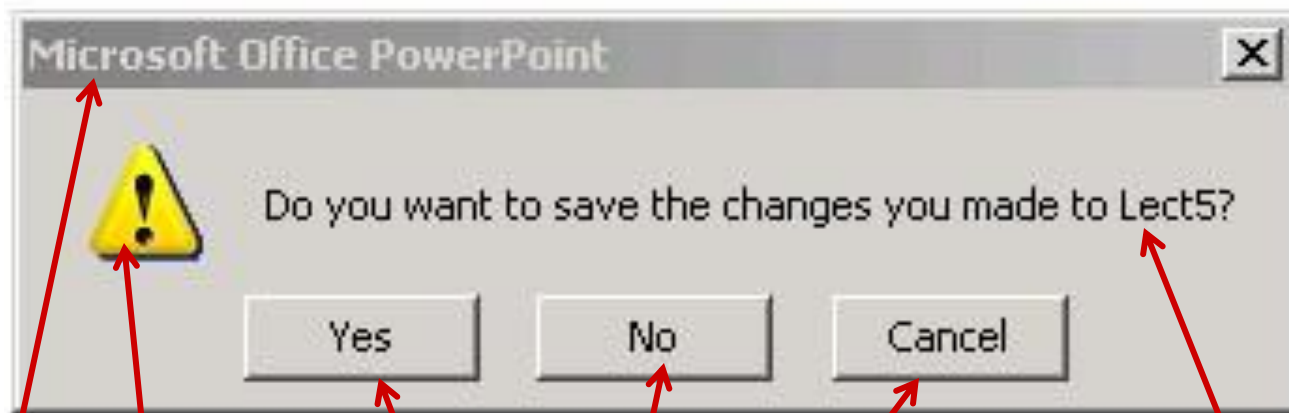


Interactive In and Output

- ▶ We have already seen how to transfer data between the spreadsheet and VBA programs, by writing into cells and reading from cells:
 - VBA program → spreadsheet
`Range("A1").value = 2`
(puts the value 2 into cell A1)
 - spreadsheet → VBA program
`x = Range("A1").value`
(assigns the value of cell A1 to the variable x)
- ▶ Now we look at another useful technique, using message boxes.
 - this is useful when you write a code for a user, who does not know about the VBA code, as you can provide more information

Message boxes are widely used (one can hardly find any kind of software today that does not use them!).

For example, if you try to close an application (e.g. PowerPoint) without saving your work, you will get something like this:



Title

Prompt

Warning message Icon

Buttons

► Message box:

- displays a message in a dialog box and returns an integer value which depends on the answer of the user

syntax:

```
return = MsgBox(prompt [, buttons] [, title] [, helpfile ,context])
```

- parameters in [] are optional, i.e. you don't have to specify them
- when you omit the optional parameters you have to include the ,
- or:

syntax:

```
return = MsgBox(prompt:= "...", title:= "...")
```

- now you do not have to include the commas
- we will not treat here the helpfile and context option (they allow to display some help information)

prompt \equiv string expression, the text displayed in the dialog box
(maximal 1024 characters)

title \equiv string expression, the text displayed in the title bar of
the dialog box. When omitted, it is the application name.

buttons \equiv a sum of several values specifying:

(a) the number and type of buttons:

Constant	Value	Description
vbOKOnly	0	OK button only
vbOKCancel	1	OK and Cancel
vbAbortRetryIgnore	2	Abort , Retry , and Ignore
vbYesNoCancel	3	Yes , No , and Cancel
vbYesNo	4	Yes and No
vbRetryCancel	5	Retry and Cancel

(b) the icon style

vbCritical	16	Display Critical Message icon
vbQuestion	32	Display Warning Query icon
vbExclamation	48	Display Warning Message icon
vbInformation	64	Display Info Message icon



(c) the default button

(this is the button selected when you just press return)

vbDefaultButton1	0	First button is default
vbDefaultButton2	256	Second button is default
vbDefaultButton3	512	Third button is default
vbDefaultButton4	768	Fourth button is default

(d) the modality of display

vbApplicationModal	0	The application stops until the user responds
vbSystemModal	4096	whole system stops until the user responds
vbMsgBoxHelpButton	16384	adds Help button
VbMsgBoxSetForeground	65536	MsgBox is foreground
vbMsgBoxRight	524288	Text is right aligned
vbMsgBoxRtlReading	1048576	text right-to-left

- select maximal one number from each of the groups (a) to (d)
- you can either use the Excel constant name or the number

e.g. buttons := 3 + 32

buttons := 35

buttons := **vbYesNoCancel** + **vbQuestion**

return \equiv a number between 1 and 7 which depends on the answer

- you can either use the Excel constant name or the number

Constant	Return value	Selected button
vbOK	1	OK
vbCancel	2	Cancel
vbAbort	3	Abort
vbRetry	4	Retry
vbIgnore	5	Ignore
vbYes	6	Yes
vbNo	7	No

- e.g. if the OK button is selected return has the value 1 or vbOK

► Examples: (see Excel file!)

Sub message1()

MsgBox ("Do you know how to view pdf files?")

End Sub

or: ret = MsgBox(Prompt:="Do you know how to view pdf files?")

Sub message2()

prompt = "Do you know how to view pdf files?"

title = "Programming Excel/VBA PartII"

ret = MsgBox(prompt, , title)

End Sub

⇒ displays a message box with OK button

prompt: Do you know how to view pdf files?

title: Microsoft Excel (in message1)

title: Programming Excel/VBA PartII (in message2)


```
Sub message3()
```

```
.....
```

```
ret = MsgBox(prompt:=pr, Buttons:=3, Title:=ti)
```

```
End Sub
```

⇒ displays a message box with Yes/No/Cancel button

```
Sub message4()
```

```
.....
```

```
bu = vbYesNoCancel + vbQuestion
```

```
ret = MsgBox(prompt:=pr, Buttons:=bu, Title:=ti)
```

```
End Sub
```

⇒ displays a message box with Yes/No/Cancel button and question mark icon (warning query icon)

Sub message5()

pr = "Do you know how to view pdf files?"

ti = "Programming Excel/VBA PartII"

111:

ret = MsgBox(prompt:=pr, Buttons:=35, Title:=ti)

If ret = vbYes Then (or: ret = 6 then)

ret = MsgBox("Good, you can print the lecture material", 48, ti)

ElseIf ret = vbNo Then (or: ret = 7 then)

ret = MsgBox("By now you should know!", 16, ti)

Else

ret = MsgBox("Either you know or you don't. Decide!", 32, ti)

GoTo 111

End If

End sub

► Goto command:

- forces the program to go to a certain position

syntax:

 position:

 Goto position

or:

syntax:

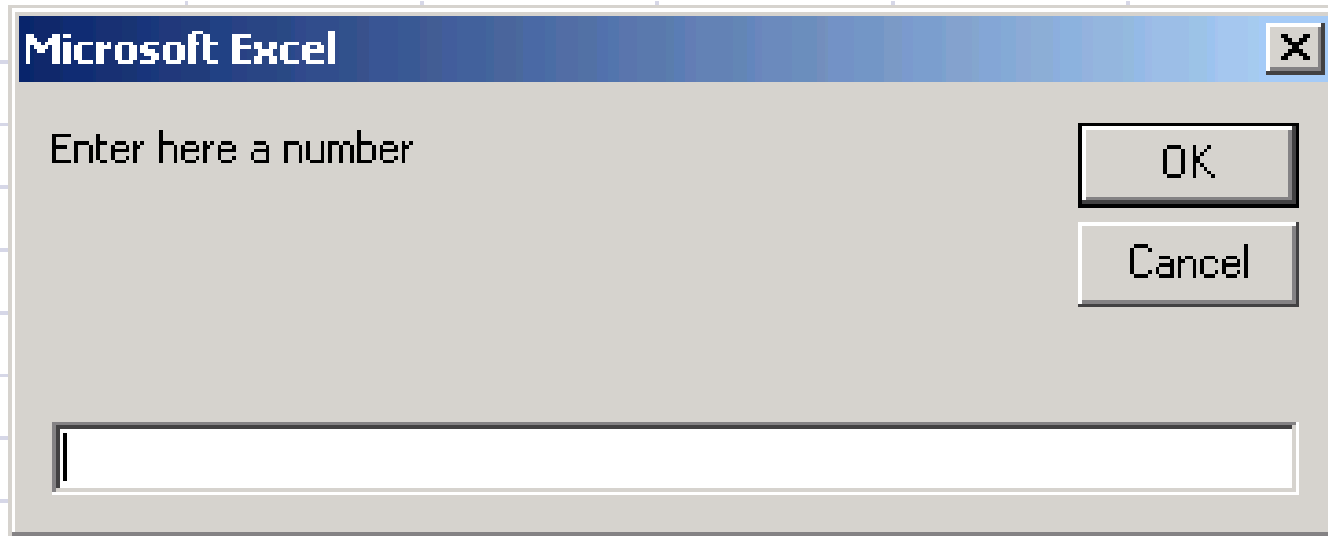
 Goto position

 position

 make sure can get out of this loop!!!!

► Input box:

- displays a prompt in a dialog box, waits for the user to enter a text or click a button, and returns a string containing the content of the text box or the value “” (the empty string)



syntax:

```
return = InputBox(prompt [, title] [, default] [, xpos] [, ypos])
```

default ≡ a default output value

xpos/ypos ≡ horizontal/vertical distance of the left/upper edge of the dialog box from the left/top edge of the screen.

return ≡ an input box always contains an OK and a Cancel button. If you choose the **OK** button, InputBox returns the value entered in the text box. If you click the **Cancel** button, InputBox returns a zero-length string ("").