COMP1022Q
Introduction to Computing with Excel VBA

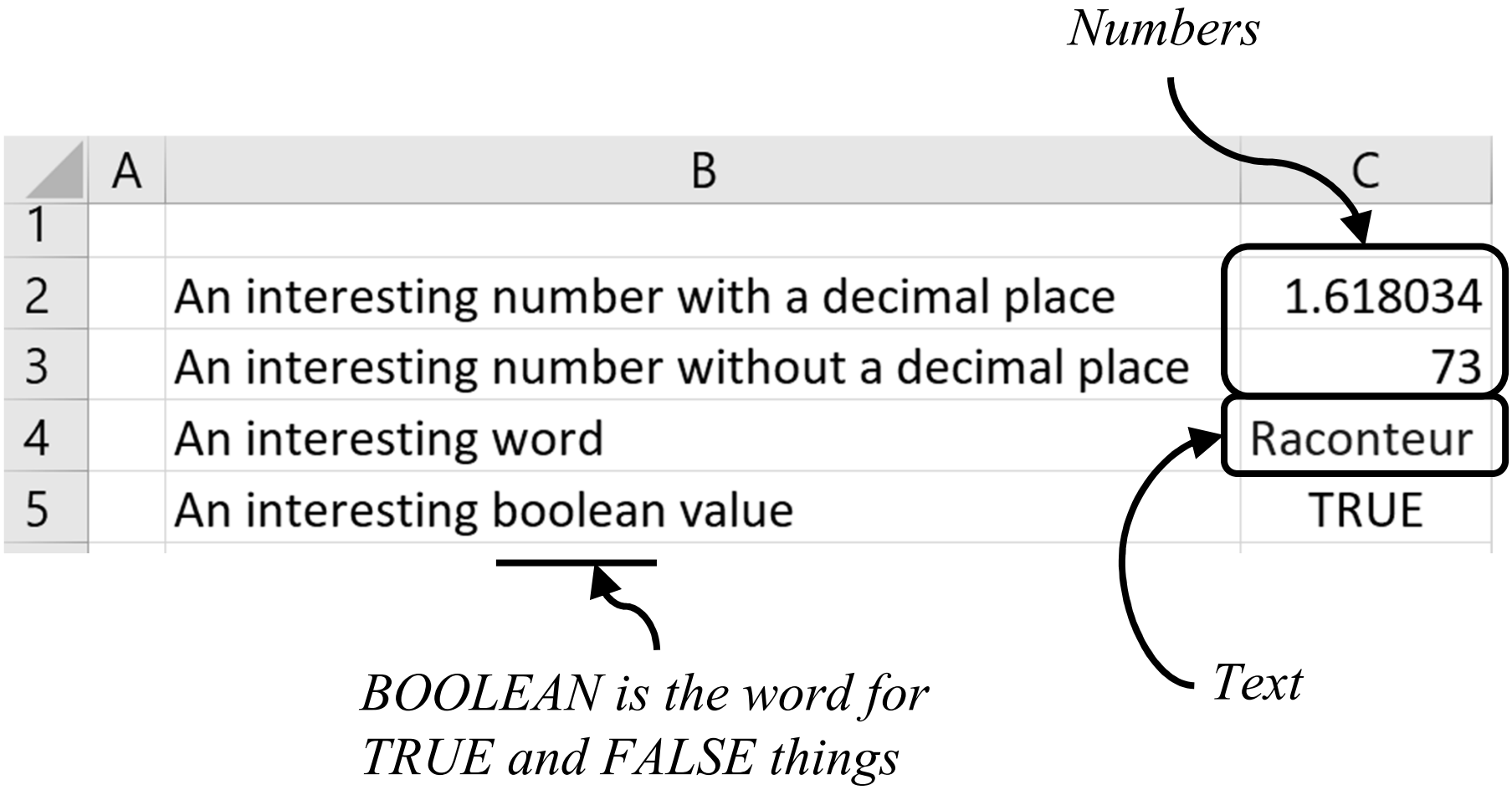
Different Types of Variable in VBA

David Rossiter, Gibson Lam and Oz Lam

Outcomes

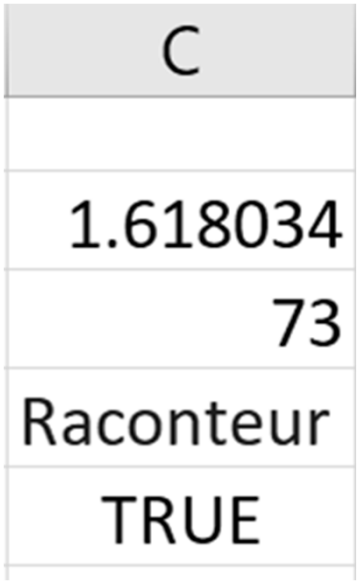
- After completing this presentation, you are expected to be able to:
  1. Understand the use of five common types of VBA variables and their limitations

Things in Spreadsheet Cells



How About Variables in VBA?

- You can simply enter whatever you like into a cell, such as a number with a decimal place, a number without a decimal place, text, and TRUE or FALSE



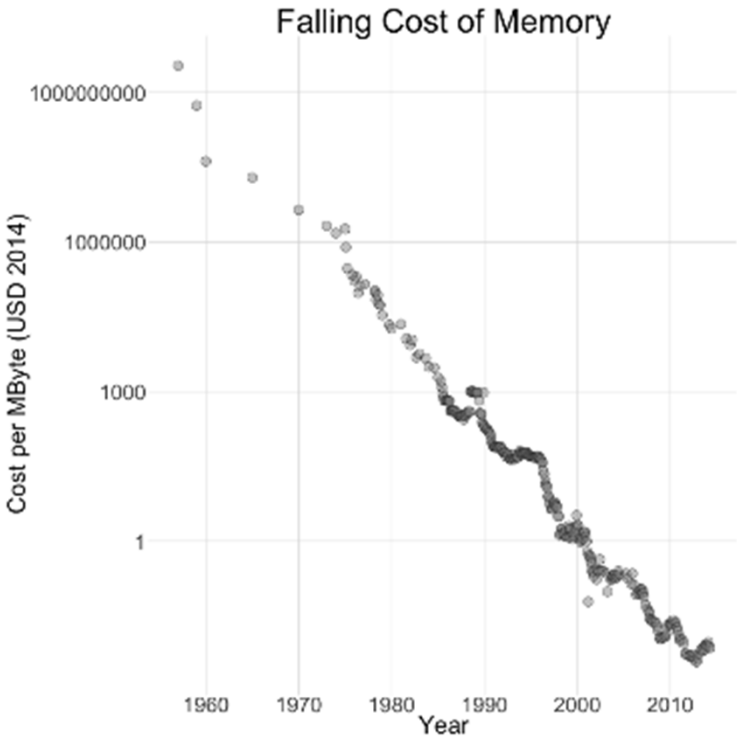
C
1.618034
73
Raconteur
TRUE

- But when you do proper programming with variables, e.g.:

```
Dim Name As String
```

you have to be more precise and careful

The Early Days



- In the early days of computers, memory was very expensive
- So a programmer would use the smallest amount of memory that does the job
- Different types of variable use up different amounts of memory
- Even today, it is still good to use the most suitable type of memory e.g. if you see code which has a *String*, you know some kind of text will go in it

Making and Using a Variable

We have seen this before

- For example, you can make a variable called *Name* which holds a string (a piece of text) using this code:

```
Dim Name As String
```

- You can use the variable to hold some text using this code:

```
Name = "David"
```

Name

- You can then show the content of the variable using a message box:

```
MsgBox Name
```

Different Types of Variable

- In addition to the *String* variable which stores text, there are many other types of variable for storing other things
- You have already seen the use of *Integer* variables and *Long* variables
- In this presentation, together with Integer and Long, we will also look at three other types of variables, which are shown on the next page

Variables in This Presentation

- We will look at these types of variable

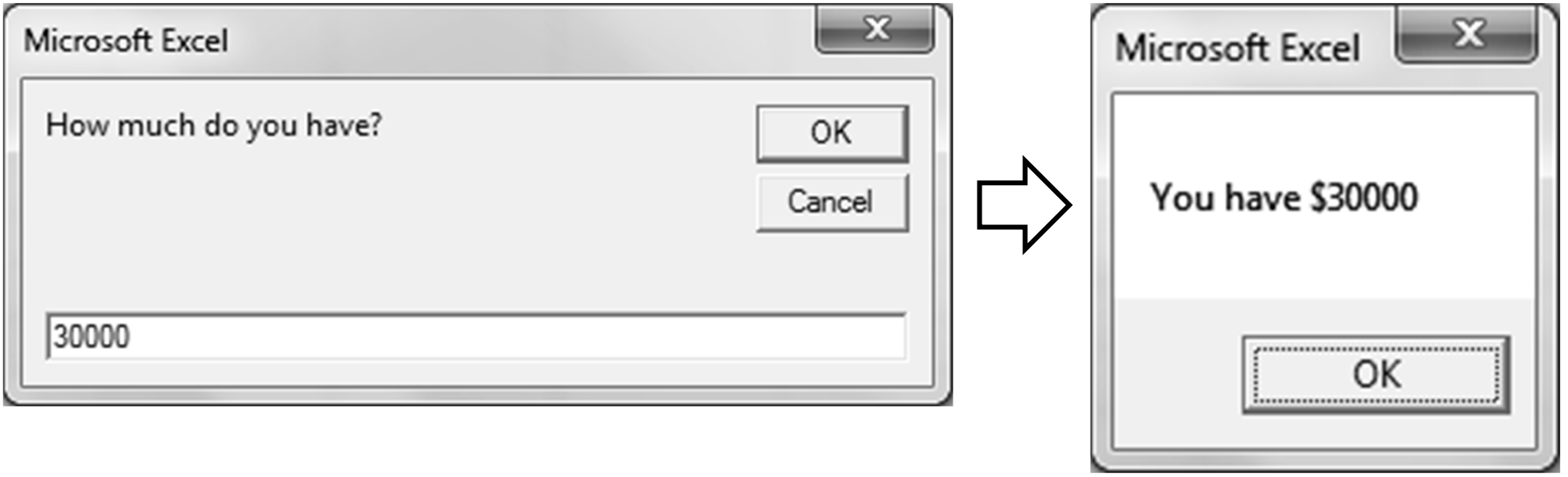
  1. *Integer* for storing (small) integer numbers
  2. *Long* for storing (large) integer numbers
  3. *Single* for storing (less accurate) decimal numbers
  4. *Double* for storing (more accurate) decimal numbers
  5. *Boolean* for storing the *True* or *False* values

- These are not the only types of VBA variable, we will encounter other types later

Integer Variables

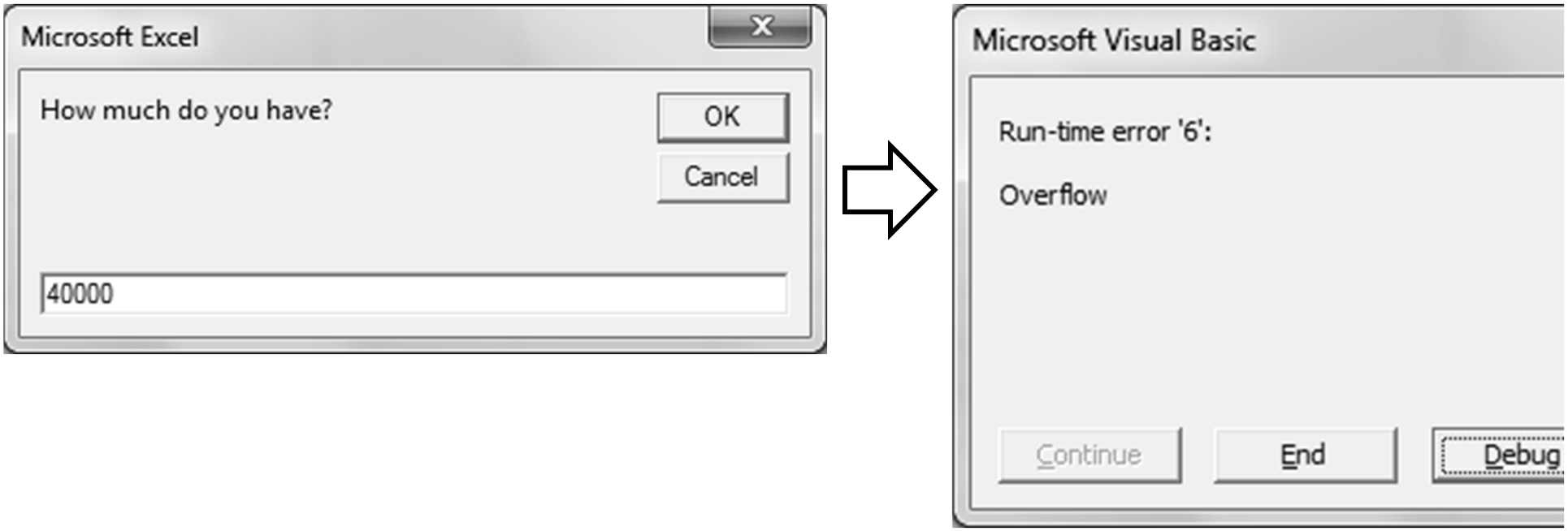
- If you want to store an integer number in VBA, you can use an *Integer* variable, i.e.

```
Dim Money As Integer
Money = InputBox("How much do you have?")
MsgBox "You have $" & Money
```



Range of an Integer Variable

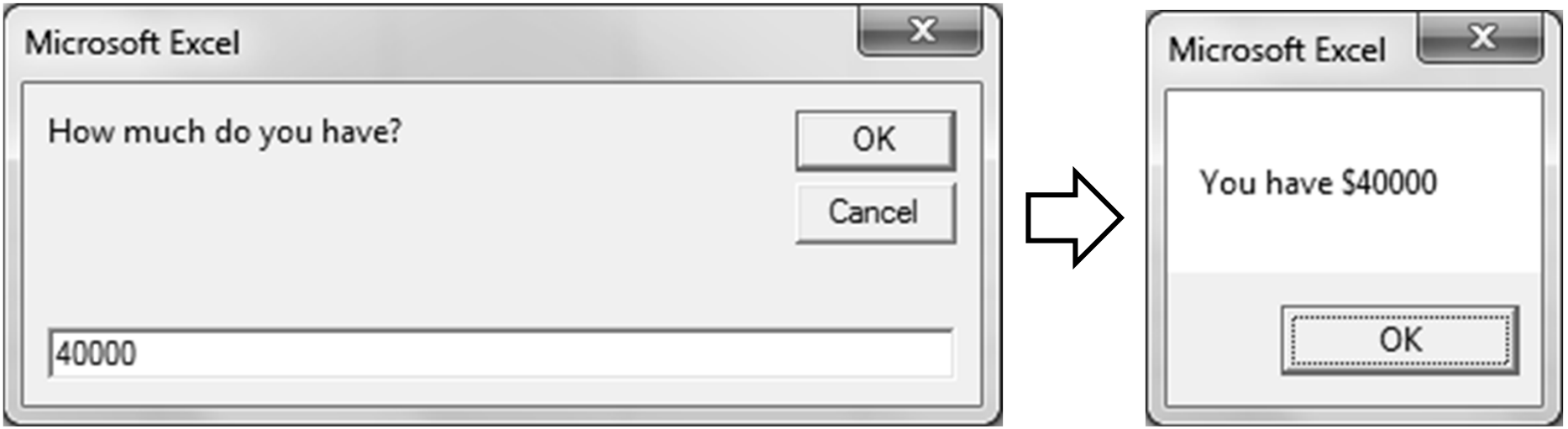
- An integer variable can only handle a number in the range -32,768 to 32,767
- If you try to put a number that is outside that range into an Integer variable, the code will show an error



Long Variables

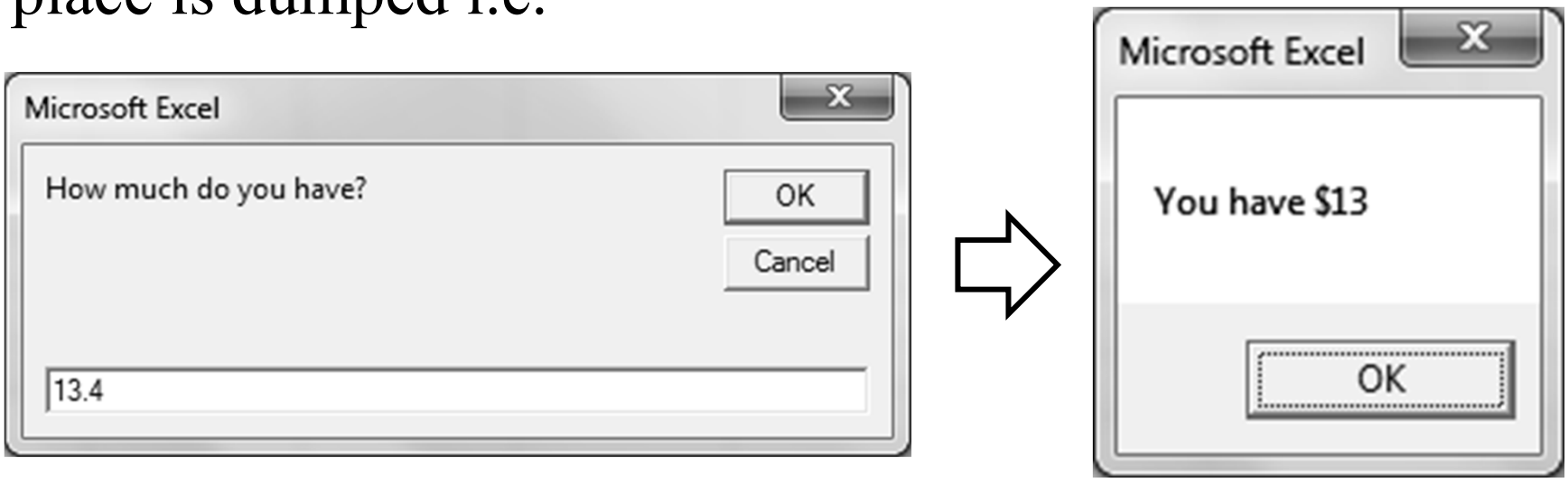
- If you want to store a number larger than 32,767 or smaller than -32,768, use a *Long* variable
- A *Long* variable can store a number in the range -2,147,483,648 to 2,147,483,647

```
Dim Money As Long
Money = InputBox("How much do you have?")
MsgBox "You have $" & Money
```



Using a Number with a Decimal Place

- If you try to enter a number which has a decimal place (such as 13.4) in the previous two examples, the number is automatically rounded up or down and the decimal place is dumped i.e.

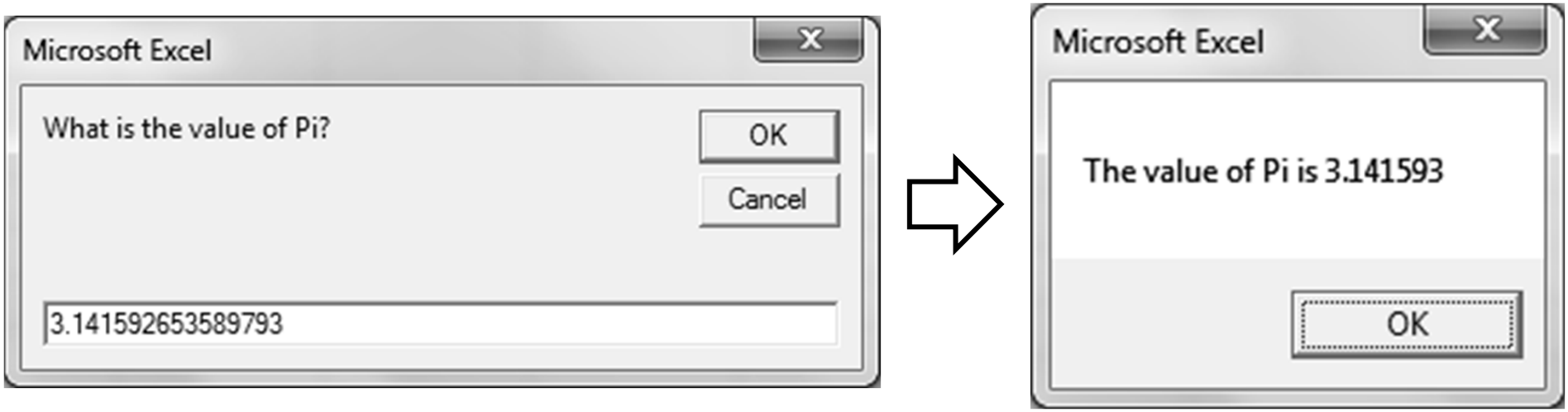


- If you want to keep the decimal place, you need to use a variable type which can handle it e.g. *Single* or *Double*

Single and Double Variables 1/2

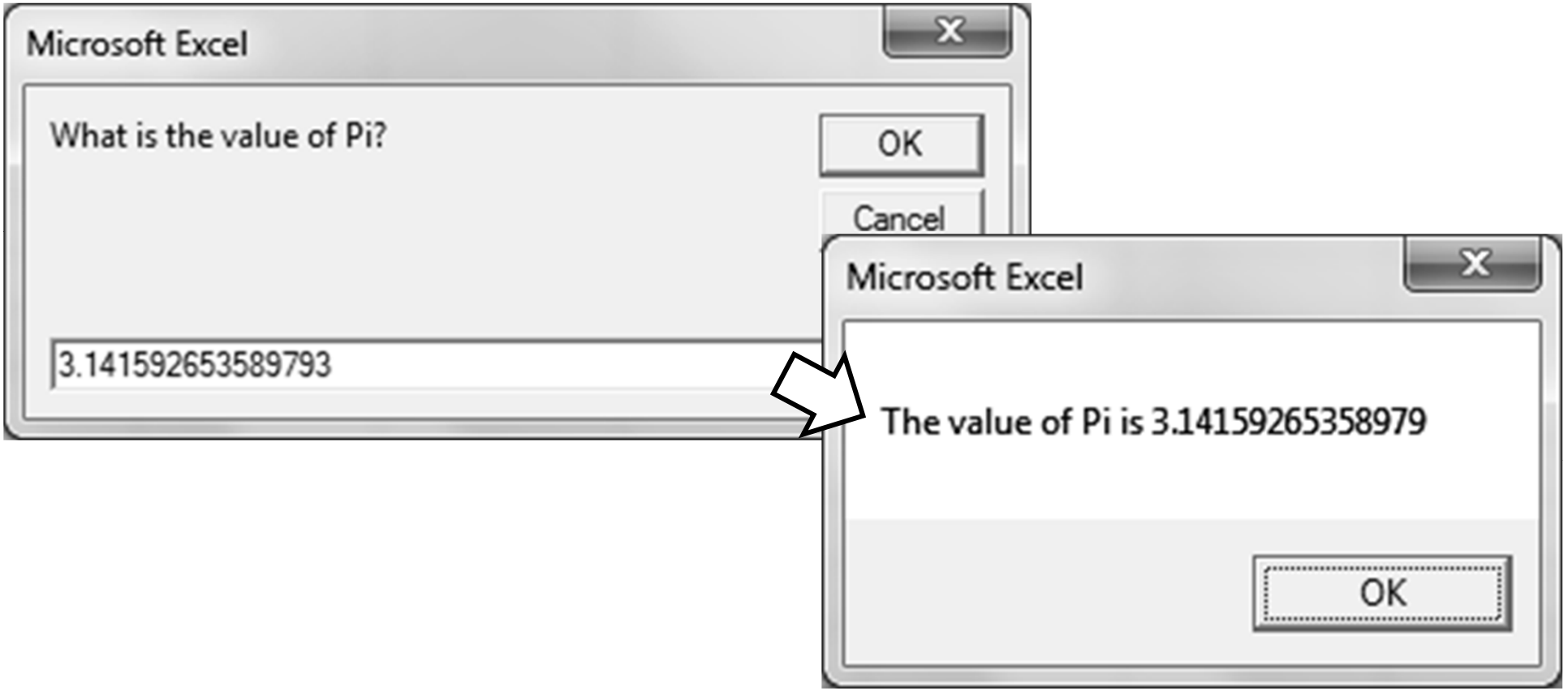
- *Single* and *Double* variables can both handle a decimal place, but *Double* has more accuracy

```
Dim Pi As Single
Pi = InputBox("What is the value of Pi?")
MsgBox "The value of Pi is " & Pi
```



Single and Double Variables 2/2

```
Dim Pi As Double
Pi = InputBox("What is the value of Pi?")
MsgBox "The value of Pi is " & Pi
```



Boolean Variables

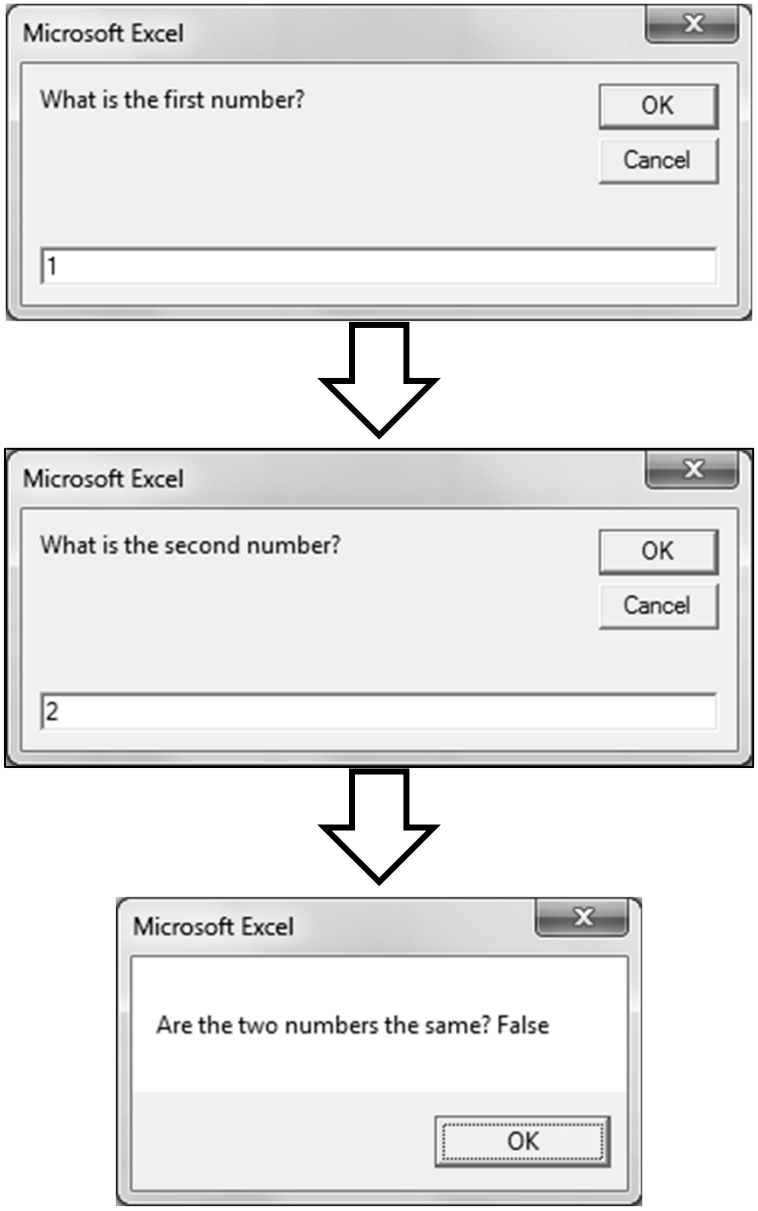
- A *Boolean* variable is used to store one of these two values only: *True* or *False*
- For example, the comparison of two numbers can be put in a variable like this:

```
Dim Number1 As Double
Dim Number2 As Double
Dim Comparison As Boolean
```

- We will ask Excel to run this code (i.e. use the code), see next slide

```
Number1 = InputBox("What is the first number?")
Number2 = InputBox("What is the second number?")
Comparison = (Number1 = Number2)
MsgBox "Are the two numbers the same? " & Comparison
```

- Running the code when the two numbers are different



- Running the code when the two numbers are the same