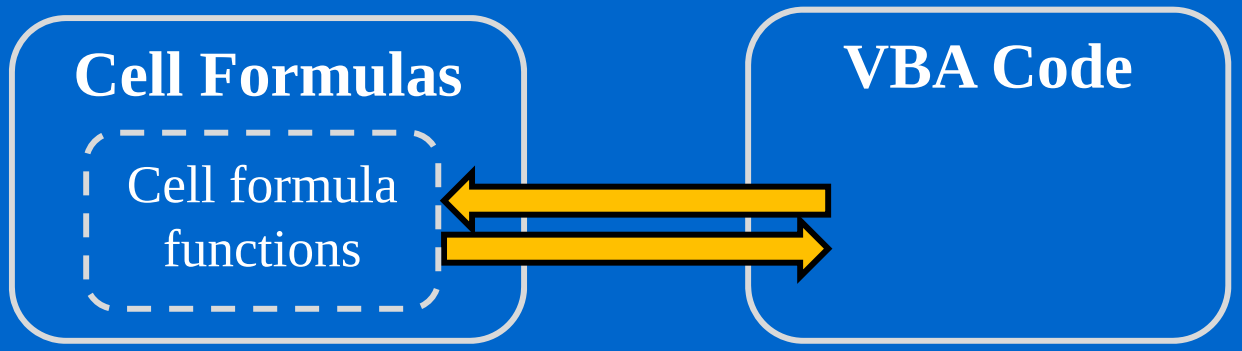
COMP1022Q
Introduction to Computing with Excel VBA

Using Cell Formula Functions in VBA

Gibson Lam and David Rossiter

This Presentation

- Although we haven't seen it before, VBA code can use any cell formula (such as `SUM()`, `COUNTIF()`, etc)



- Sometimes this is very helpful, because it means you don't have to work out how to do something in VBA if you already know how to do it in a cell formula
- Also, worksheet function typically (but not always) runs faster than any VBA code that you could write which does the same thing

How to Use Worksheet Functions in VBA

- Excel cell formula functions are also called *worksheet functions* because they are usually in a worksheet cell
- In VBA, you can use any worksheet function like this:

`WorksheetFunction.`*`name_of_function`*`(`*`...parameters...`*`)`

- Here is some example VBA code which shows the `SUM` worksheet function being used to add together the values of cells B4 to B10:

```
Dim Total As Integer
Total = _
    WorksheetFunction.Sum( Range("B4:B10") )
```

A cell reference has to be given to the function as a VBA range, like this, otherwise it won't work

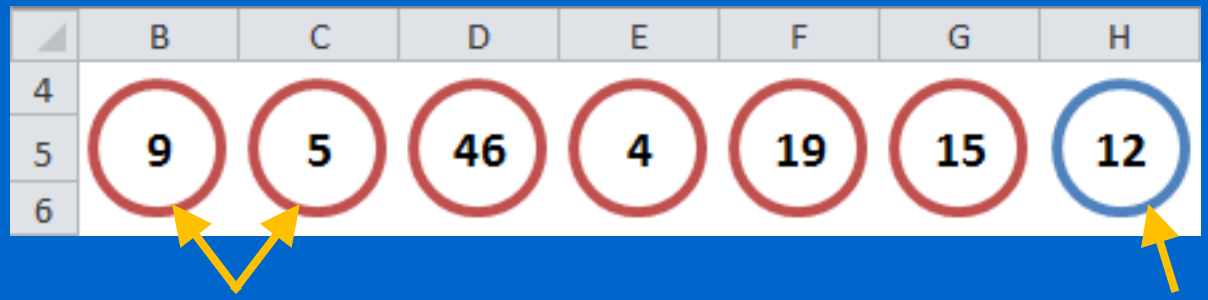
This means 'the code is continued on the next line'. We are only using it here because we can't fit everything in one line in the PPT file!

An Example – Lottery Number Generator

- Now we will show two sets of VBA code which do the same thing
- Both of them generate lottery numbers
- First we will show an example written using VBA code and VBA functions only
- Then we will show a version of the code which is shorter, because it is written using a combination of VBA code and worksheet functions

An Example – Mark Six Number Generator

- As you already know, *Mark Six* is the HK lottery
- It uses 7 unique numbers out of a possible 49 numbers
- In the following two slides, a Mark Six number generator is shown, written using VBA code
- It randomly draws 7 unique numbers from 1 to 49 and shows them in cells B5:H5, like this:



The circles have already been added manually, i.e. they are not added using VBA code

In Hong Kong, the last number is called the special number

The Code Without Worksheet Functions 1/2

```
Dim Ball As Integer, Number As Integer
Dim Count As Integer, Column As Integer

Randomize

Ball = 1
While Ball <= 7
  Number = Int(Rnd() * 49) + 1

  Count = 0
  For Column = 2 To 8
    If Number = Cells(5, Column) Then
      Count = Count + 1
    End If
  Next Column
```

The variable Ball starts at 1. The while loop runs while Ball is smaller than or equal to 7, because we need 7 balls

A random number is generated from 1 to 49 inclusive

This for loop counts the number of times the newly generated number appears in the current list of drawn numbers

We do this because we need to make sure the number we just created isn't already being used

The Code Without Worksheet Functions 2/2

When `Count` is 0, that means the newly generated number has not been drawn before - so that means it can be added to the list of drawn numbers

```
    If Count = 0 Then
      If Ball < 7 Then
        MsgBox "The next number is " & Number & "!"
      Else
        MsgBox "The special number is " & Number & "!"
      End If

      Cells(5, Ball + 1) = Number

      Ball = Ball + 1
    End If
  Wend
```

In HK the last ball is called the 'special number'

Store the number in the worksheet at the appropriate cell, and move on to the next ball

Using Worksheet Functions in the Example

- The code shown in the previous slides works fine
- It can correctly generate 7 unique numbers
- Now let's simplify the code by using these worksheet functions:

  `RANDBETWEEN(a, b)`

  - Returns a random number between and including `a` and `b`

  `COUNTIF(range, criteria)`

  - Counts the values satisfying `criteria` inside `range`

The Code Using Worksheet Functions

```
Dim Ball As Integer, Number As Integer
Dim Count As Integer, Column As Integer

Ball = 1
While Ball <= 7
   Number = WorksheetFunction.RandBetween(1, 49)

   If WorksheetFunction.
        CountIf(Range("B5:H5"), Number) = 0 Then
      ...the number is added to the
         cell, code not shown here...
      Ball = Ball + 1
   End If
Wend
```

A random number is generated from 1 to 49 inclusive, this is easier than the VBA `Randomize` and `Rnd` code we used before

Instead of using a VBA for loop, a `COUNTIF` function is used to count the occurrence of the newly generated number in the drawn number list, it's also much easier than the previous code we used!