

COMP1022Q
Introduction to Computing with Excel VBA

Using the Set Keyword

Gibson Lam and David Rossiter

The Use of the Set Keyword

- Previously you have seen this line of code:

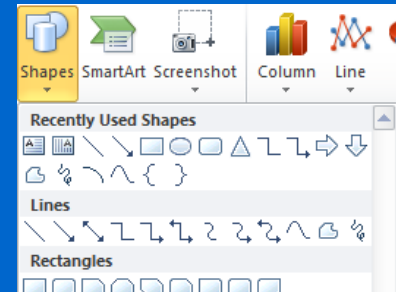
```
Set Square = ActiveSheet.Shapes.AddShape( _  
    msoShapeRectangle, 125, 100, 150, 150)
```

- The above code creates a shape (which is a square) and then stores the newly created shape in a variable called Square
- After running this line of code you can then do things to the shape through the Square variable
- Instead of using 'Square = ...' the above code needs to use 'Set Square = ...'

The Set Keyword is for Objects

- The keyword `Set` has to be used when you want to store an *object* in a variable
- You can think of an object as a group of values, which are called *properties*
- For example, an Excel shape is a group of properties which include height, width, colour and so on

The
shape
toolbar



An Excel Shape

width

height

colour

and more...

Two Common Objects in VBA

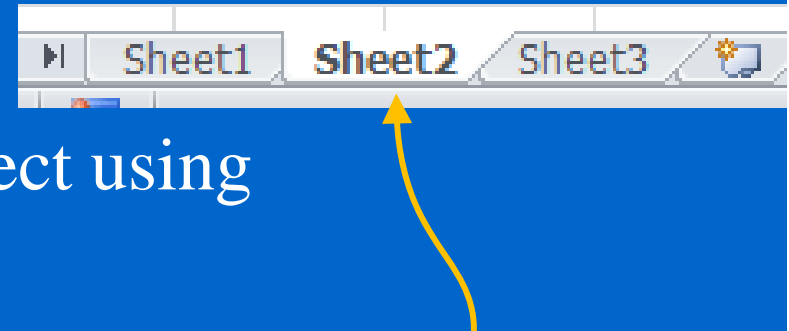
- Here are two common objects you can use in Excel VBA:
 - The range object
 - A range object is the cell(s) in a worksheet
 - You can get a range object using `Range ()`, like this:

```
Set MyCell = Range( "A5" )
```

- The worksheet object

- You can get a worksheet object using `Worksheets ()`, like this:

```
Set MyWorksheet = Worksheets( "Sheet2" )
```



When to Use/Not Use Set


- When you want to put a single value (such as an integer or a piece of text) into a variable, you do not need to use Set, for example:

```
Dim Width As Integer  
Width = 100
```

- When you want to put an object (such as a shape, a range or a worksheet) into a variable, you need to use Set, for example:

```
Dim MyCell As Range  
Set MyCell = Range( "B5" )  
MyCell.Interior.ColorIndex = 1
```

*Change the cell
background to
black using the
MyCell variable*



Default Property of the Range Object

- A common way to put something in a cell is this:

```
Range("D2").Value = "I am D2!"
```

- A common way to get the contents of a cell is this:

```
Content = Range("D2").Value
```

- This code also works:

```
Range("D2") = "I am D2!"    'Put thing into D2  
Content = Range("D2")      'Get content from D2
```

- They work because there is a default property of the range object, which is the `Value` property
- If you don't specify any particular property for an object, it uses the default property (if there is one)

The Set Keyword and the Default Property

- Because of the default property you have to be careful when using the range object
- For example, here are two lines of VBA code:
 - 1) `MyVar = Range("A4")`
 - 2) `Set MyVar = Range("A4")`
- What is the difference?
 - Line 1 puts the `Value` property into `MyVar`
 - Line 2 puts the range object (the entire object) into `MyVar` and then, if you want to, you can get the cell content using `MyVar.Value`

An Example of Using the Set Keyword and a For Loop

- This example changes eight cells to black diagonally

```
Set CurrentCell = Range("B4")
```

```
For Num = 1 To 8
```

```
    CurrentCell.Font.ColorIndex = 2
```

```
    CurrentCell.Interior.ColorIndex = 1
```

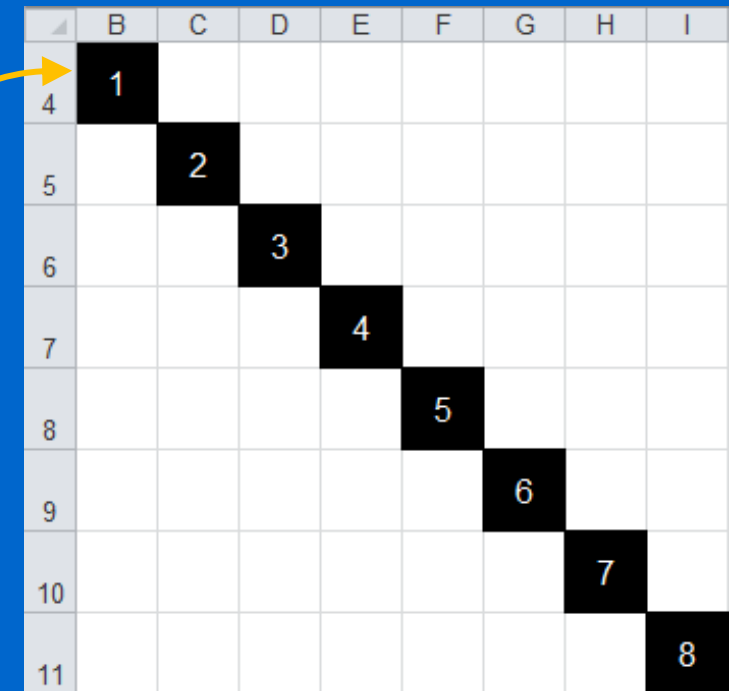
```
    CurrentCell = Num
```

*This line actually means:
CurrentCell.Value = Num*

```
    Set CurrentCell = CurrentCell.Offset(1, 1)
```

```
Next
```

The variable CurrentCell becomes the next cell diagonally each time the loop content is executed



	B	C	D	E	F	G	H	I
4	1							
5		2						
6			3					
7				4				
8					5			
9						6		
10							7	
11								8