
R E 2 7 0 8

1

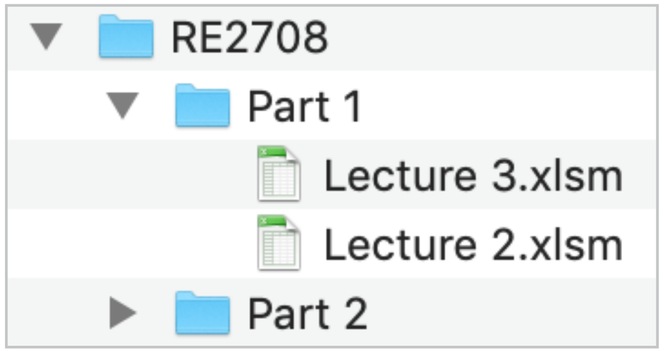
CONNECTING WITH EXCEL

EXCEL

Open a new Excel file.

Save it as ‘Excel Macro-Enabled Workbook’:

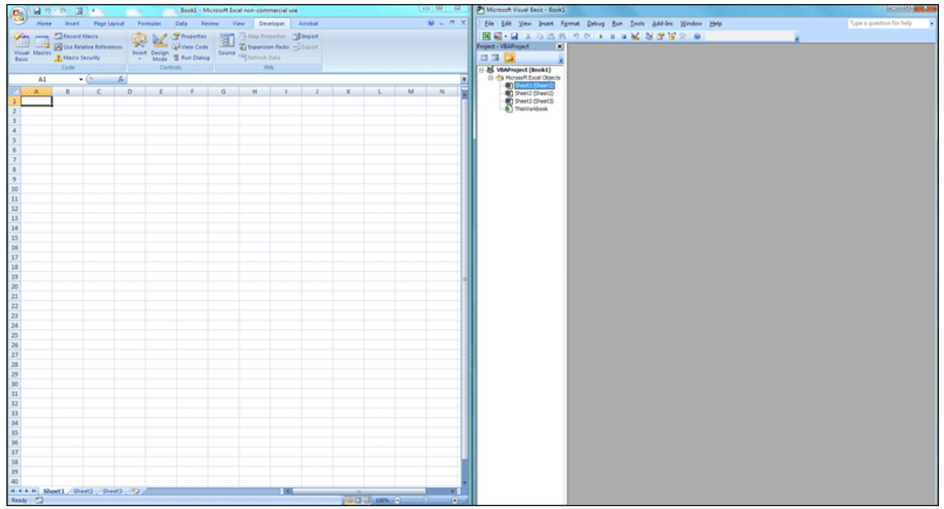
Lecture3.xlsm



To make work easier, let’s start by placing the worksheet and the VBA Editor window next to each other.

On Windows, this is easiest done by pressing the Win-key and the right-left arrows, respectively.

On Mac, this has to be done manually.



VBA allows us to interact directly with Excel.

Let’s start by writing some text in cell ***“C3”***.

```
Sub MyCode1()

    ActiveCell.Value = "Brian"

End Sub
```

Next, let's assign Brian a grade of ***80*** marks in his final exam:

Did he pass his exam?

VBA can help us find out.

Here is the code:

	A	B	C
1			
2			Exam result
3		Brian	80
4			
5			

```
Sub MyCode2()

    If ActiveCell.Value > 50 Then
        MsgBox ("Pass")
    Else
        MsgBox ("Fail")

End Sub
```

Can we make this a little more professional?

(Nobody likes message boxes…)

```
Sub MyCode2()

    If ActiveCell.Value > 50 Then
        ActiveCell.Value = "Pass"
    Else
       ActiveCell.Value = "Fail"
    End If

End Sub
```

But: Is this solution OK? Not really.

Brian's grade gets overwritten by the text results. That can't be right.

… and even more professional …

By using the:

Offset(Rows, Columns)

function.

```
Sub MyCode2()

    If ActiveCell.Value > 50 Then
        ActiveCell.Offset(0, 1).Value = "Pass"
    Else
       ActiveCell.Offset(0, 1).Value = "Fail"
    End If

End Sub
```

It’s much better now.

Brian’s exam result is written down just next to his grade.

2

THE DOT NOTATION

Understanding object-oriented programming

WHAT IS AN **OBJECT**?

(in real life)

ALL OBJECTS HAVE:

- FEATURES (**PROPERTIES**)

Examples:

The rocket's height is 50m.
The rocket's colour is white.
The student's exam grade is 80.
The classroom's capacity is 100 chairs.
The air temperature in Singapore today is 28C.
The water temperature is 23C.

- METHODS (**PROCEDURES/ROUTINES**)

Examples:

Rocket - start engine!
Rocket - stop engine!
Student - wake up!
Student - enter classroom!
Singapore - start rain!
Singapore - stop rain!

VISUAL BASIC IS AN OBJECT-ORIENTED PROGRAMMING LANGUAGE

(which makes our life **very** easy)

We access *Properties* and *Methods* of an object using the “dot” notation:

Changing the value of the current active cell to “Brian”

```
ActiveCell.Value = "Brian"
```

Accessing a cell that is neighbouring the current one:

```
ActiveCell.Offset(0, 1)
```

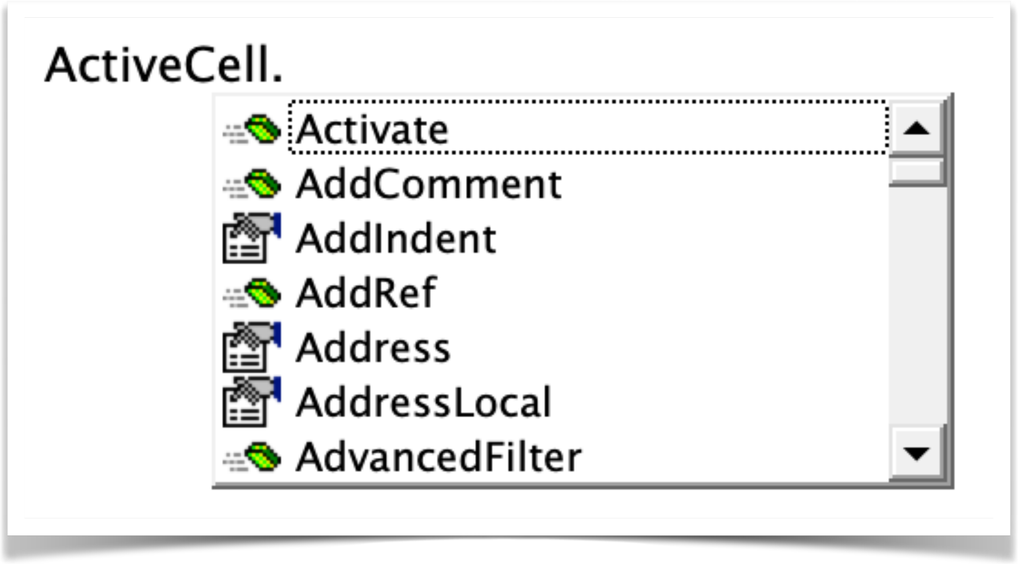
Adding a comment, copying or deleting the current active cell:

```
ActiveCell.AddComment
ActiveCell.Copy
ActiveCell.Delete
```

HOW CAN WE POSSIBLY KNOW AND NAVIGATE **ALL** THE PROPERTIES AND METHODS OF AN OBJECT?

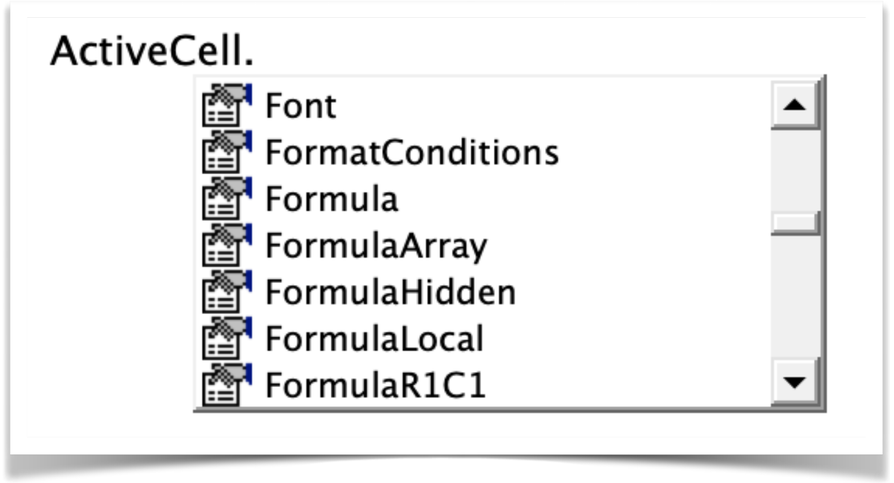
Good news:

Visual Basic Editors offers Autocompletion of object properties and methods.

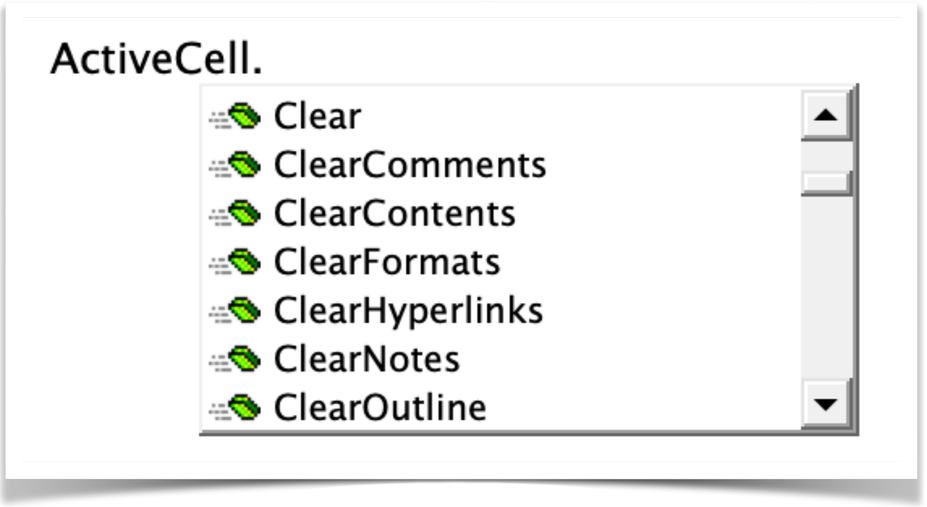


Useful icons and colour coding.

PROPERTIES



METHODS



WHAT IF WE WANT EVEN MORE HELP?

Good news:

Visual Basic Editor offers Suggestive typing.

```
ActiveCell.Offset(
        Offset([RowOffset], [ColumnOffset]) As Range
```

```
ActiveCell.AddComment(
        AddComment([Text]) As Comment
```

3

BRINGING IT ALL TOGETHER

COMPUTER THINKING

AND PROGRAMMING

CONDITIONAL STATEMENTS (IF-THEN-ELSE)

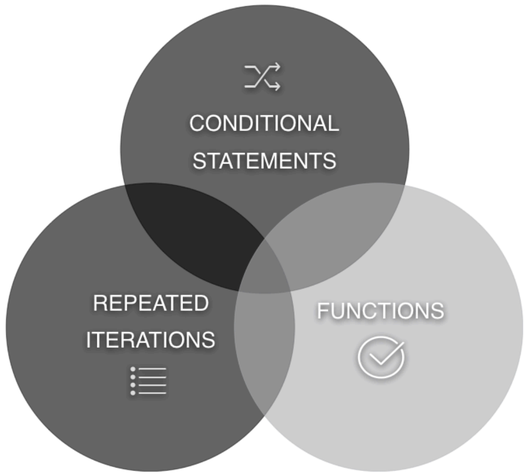
Most of our thoughts are decisions! Similarly, most of the operations that we ask the computer to do for us are conditional statements.

REPEATED ITERATIONS (FOR)

The primary reason to use a computer at all is because it can perform repetitive tasks.

FUNCTIONS

The reason why we write functions is to be able to apply the same operations to many different objects.



OBJECTS

FEATURES (**PROPERTIES**)

and

METHODS (PROCEDURES/ROUTINES)

4

THE PLAYGROUND

Designing applications

1 The **Range** object

The most frequently used Visual Basic object is **Range**.

It allows us to automatically select a range of cells for which we may want to change a property (e.g. the font size or the colour), or which we may want to change in some way (e.g. to copy-paste or delete).

```
Sub MyCode3()

    Range("A1:A3").Value = 2019
    Range("A1:A3").Font.Name = "Times New Roman"
    Range("A1:A3").Font.Size = 30
    Range("A1:A3").Interior.Color = vbWhite

End Sub
```

2 The **Worksheets** object

The Worksheets object allows us to work with the entire worksheet, change its name, print it out, or adjust its features in any way that is necessary.

```
Sub MyCode4()

    Worksheets(1).Name = "Data"
    Worksheets(1).Cells.Interior.Color = vbWhite

End Sub
```

3 The **Applications** object

The Applications object is the most comprehensive. It is a container for the entire Excel application running on our machine. We can interact with it in a number of ways, e.g. by opening a new workbook, saving the current one, or closing Excel altogether.

```
Sub MyCode5()

    Application.Quit

End Sub
```

SEE YOU NEXT WEEK!

HOMEWORK

WORKING WITH OBJECTS IN VBA

Before we turn to a more extensive discussion of objects during the next lectures, use the **Autocomplete** facility in Visual Basic Editor to find interesting and potentially useful properties and methods for the main objects that we have covered so far: ‘ActiveCell’, and ‘Range’.