

The background of the image is a soft-focus, monochromatic landscape. In the center, a dark, craggy rock formation rises from a misty base. Several tall evergreen trees are scattered across the rocks. At the very top of the central peak, there's a small, weathered stone structure, possibly a ruin or a look-out point. The overall mood is mysterious and ethereal, with heavy fog obscuring the lower half of the scene.

RE 2708

THE BUILDING BLOCKS

REMEMBER



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.



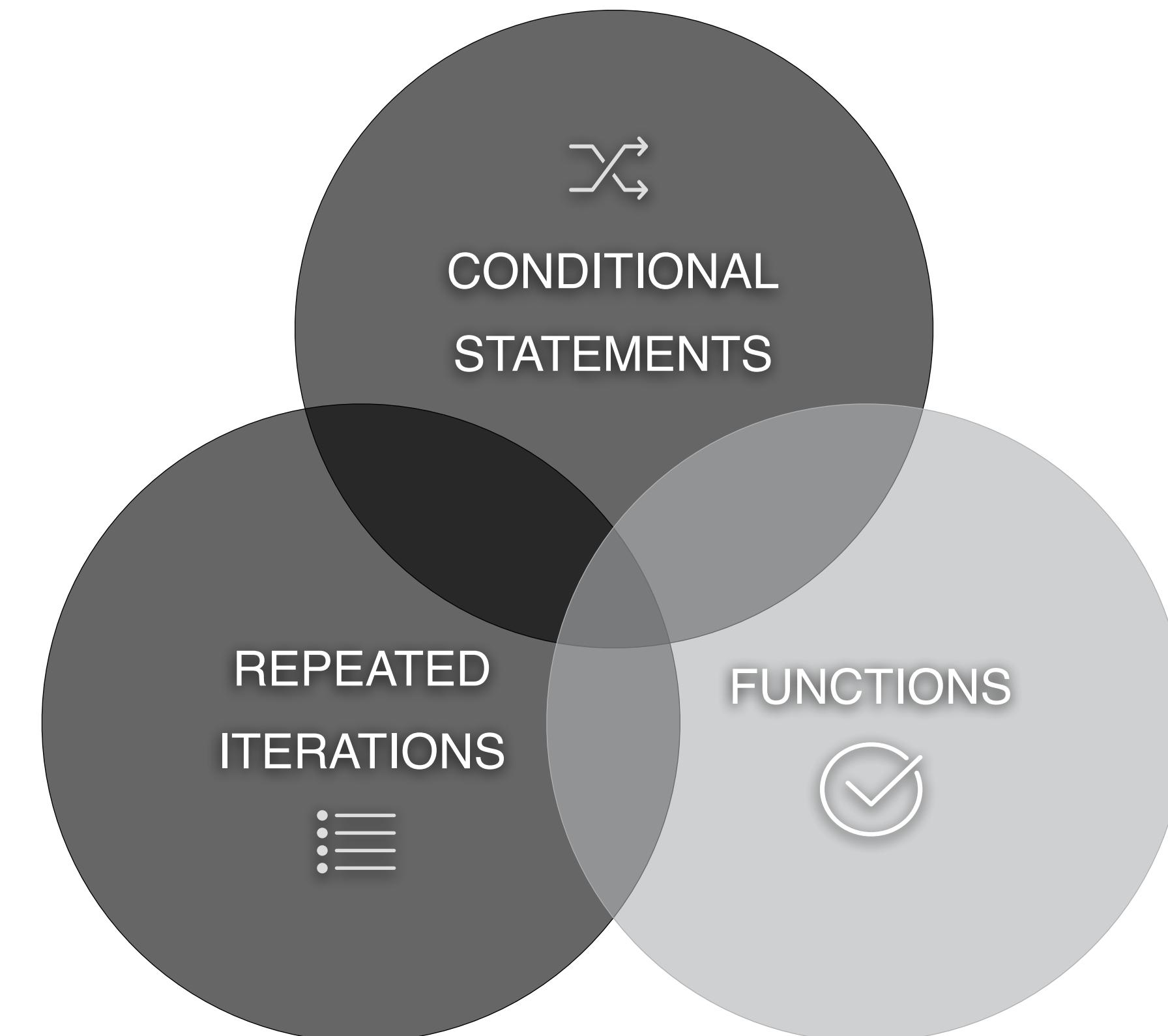
REPEATED ITERATIONS (FOR)

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.



1



WRITING CODE

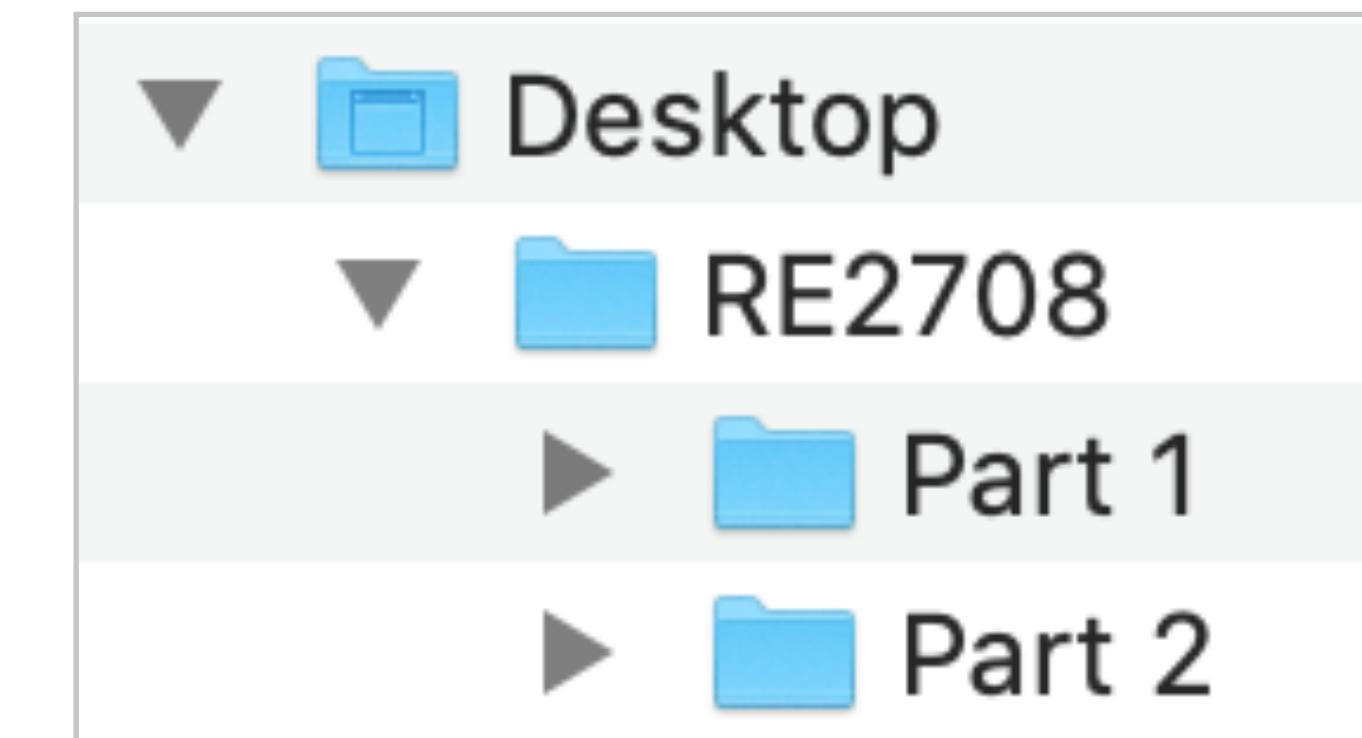


W R I T I N G

Start by building a new folder where we will place all the Excel and Python files in this module:



To organise the material better, you can add a sub-folder for each of our two parts.



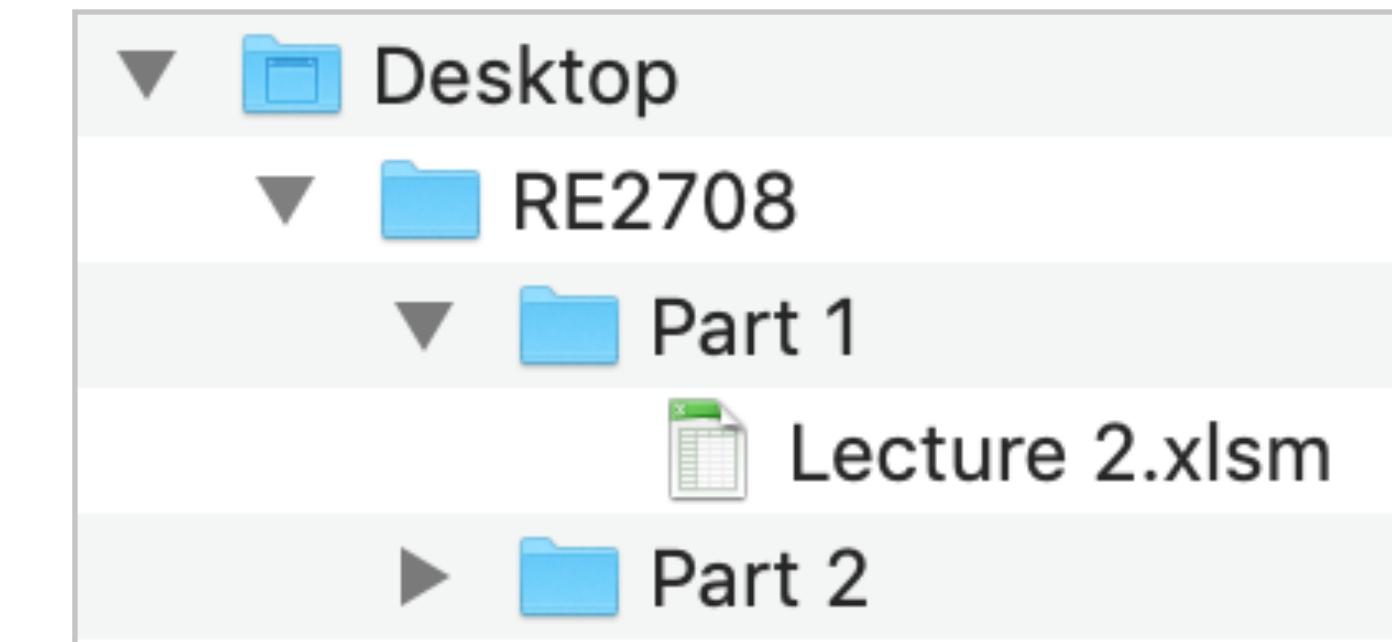


W R I T I N G

Open a new Excel file.

Save it as 'Excel Macro-Enabled Workbook':

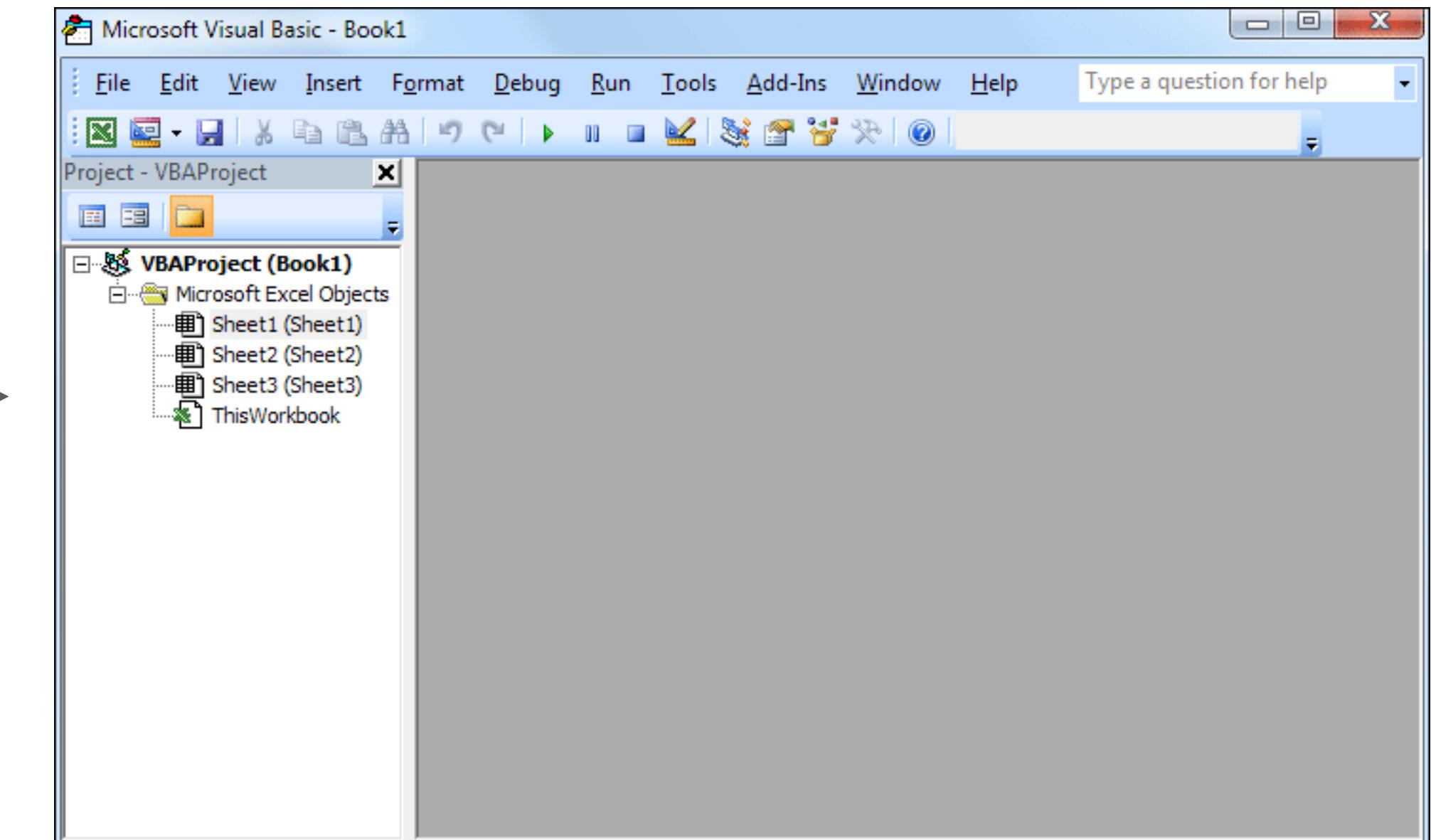
Lecture2.xlsm



From the *Developer* tab, on the *Code* panel, you can click the **Visual Basic** button.

On the **Controls** panel of the Developer tab, click **View Code**.

Alternatively, a keyboard shortcut is to hold down the left ALT key on your keyboard. Keep it held down and press the F11 key.



VBA routines can be attached to any **worksheet**, or to the **workbook** as a whole.



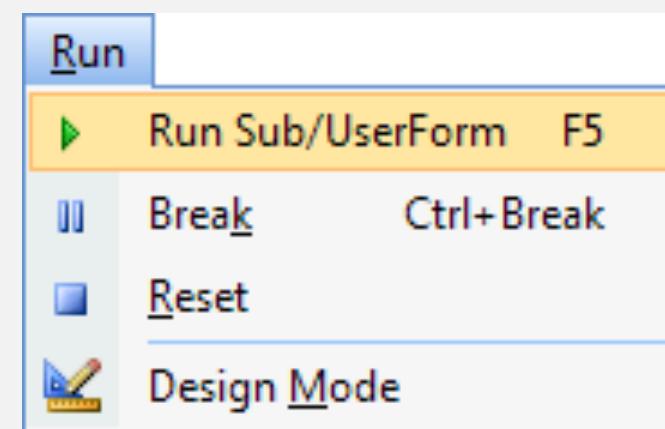


W R I T I N G

To run the code, find the toolbar at the top of the editor and click the green (*Windows*) or blue (*Mac*) triangle:



Another way to run the code is to click the **Run** menu. From the Run menu, select **Run Sub/User Form**.



A shortcut to run your code is to press the F5 key on your keyboard (*Windows*) or fn-F5 (*Mac*).

When you run this code you should see a message box.

Write the following code:

```
Sub MyCode1()
    MsgBox ("Hello World!")
End Sub
```

After running the code you should see this:





W R I T I N G

1

Conditional statements

First, please note that Visual Basic asks us to define a so-called sub-routine, using the word '**Sub**'.

**Note: A 'Macro' is another term for a 'Sub'.*

Second, we define a numerical **variable** 'Age', to which we assign the value 23.

**Note: Variables can have any names using combinations of letters and numbers. Variable names cannot start with a number.*

Third, we use the **If-Then-Else** statement to define a text variable 'Status', to which we assign the value "Young" if the Age is below 30 years, and "Old" otherwise.

Finally, we open a message box, to print the content of the variable 'Status'.



Write the following code:

Sub MyCode2()

Age = 23

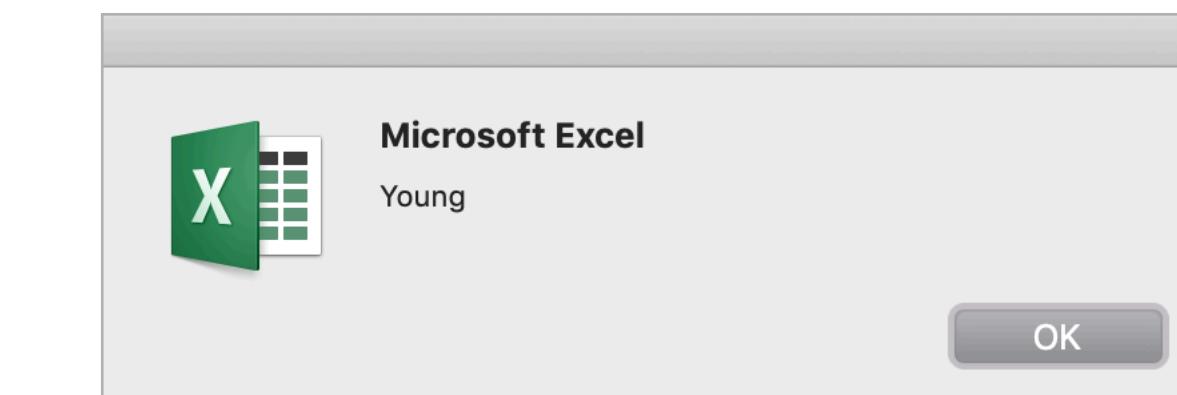
If Age < 35 Then
 Status = "Young"

Else
 Status = "Old"
End If

MsgBox (Status)

End Sub

After running the code you should see this:





W R I T I N G

2

Repeated iterations

Defining a list is not that simple...



Try to write the following code:

```
Sub MyCode3()
```

```
Names = {"John", "Brian", "Alexandra"}
```

```
End Sub
```

You should get the following **error message**:



Microsoft Visual Basic

Compile error:

Invalid character

OK



2

Repeated iterations

We need to indicate the exact size of the list.



Each element of the list is then assigned one
after the other.



Note that the index of the element is indicated
in parentheses.

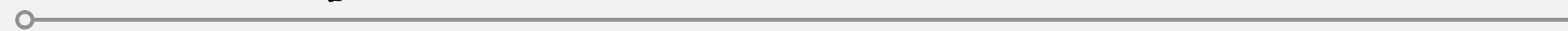


This is how **lists** are properly defined:

```
Dim Names(3) As String  
Dim Age(3) As Integer
```

```
Names(1) = "John"  
Names(2) = "Brian"  
Names(3) = "Alexandra"
```

```
Age(1) = 23  
Age(2) = 31  
Age(3) = 39
```





W R I T I N G

2

Repeated iterations

Finally, we write down the **For** statement:



Using the **list** that we have defined, we can now run the repeated iteration:

```
For i = 1 To 3
    If Age(i) < 30 Then
        Status = "Young"
    Else
        Status = "Old"
    End If

    MsgBox (Status)
    Next
```



W R I T I N G

2

Repeated iterations

And here is the complete code:

```
Sub MyCode3()
```

```
Dim Names(3) As String  
Dim Age(3) As Integer
```

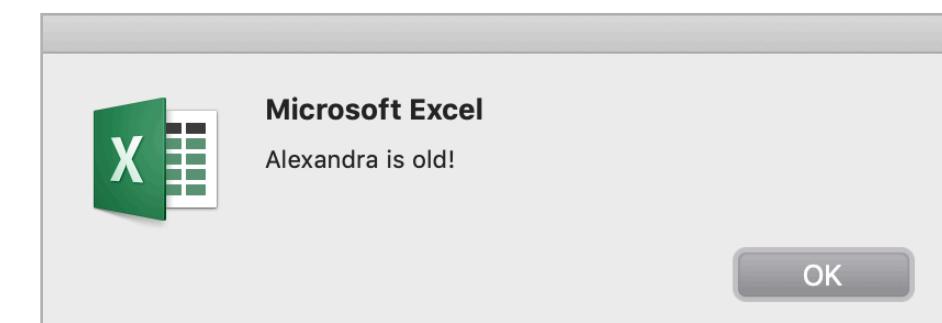
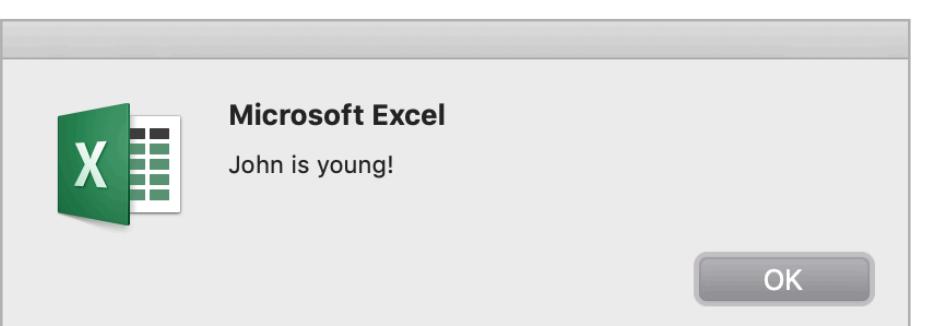
```
Names(1) = "John"  
Names(2) = "Brian"  
Names(3) = "Alexandra"
```

```
Age(1) = 23  
Age(2) = 31  
Age(3) = 39
```

```
For i = 1 To 3  
    If Age(i) < 30 Then  
        Status = "young"  
    Else  
        Status = "old"  
    End If
```

```
    MsgBox (Names(i) + " is " + Status + "!")  
Next
```

```
End Sub
```





W R I T I N G

3 Functions

There are **two reasons** why we use functions:

- To keep the code clean and organised.
- To keep the code flexible.

Every function has one or more **arguments**.

For example, in our case the argument is 'Age'.

Function Status(Age)

```
If Age < 30 Then  
    Status = "Young"  
Else  
    Status = "Old"  
End If
```

End Function

Sub MyCode4()

```
MsgBox (Status(25))
```

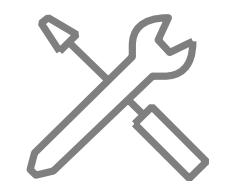
End Sub



Microsoft Excel
Young

OK

2



DESIGNING CODE



1

Variables

☆ Numbers

Age = 23

☆ Text (String)

Status = "Young"

☆ Single value

(Declaring single-value variables by **Dim** is **optional**)

```
Dim Names(3) As String  
Dim Age(3) As Integer
```

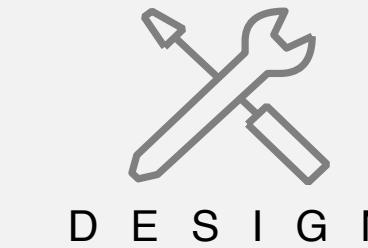
☆ List

(Declaring list variables by **Dim** is **not optional**)

```
Names(1) = "John"  
Names(2) = "Brian"  
Names(3) = "Alexandra"
```

```
Age(1) = 23  
Age(2) = 31  
Age(3) = 39
```





2

Operations

☆ Logical

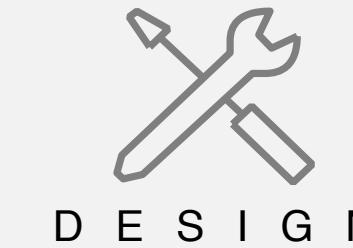
(If you have more complicated conditions, use **And**, **Or**, etc.)

Age < 30

☆ Text operations

Names(i) + " is " + Status + "!"





3

Statements

If-Then-Else

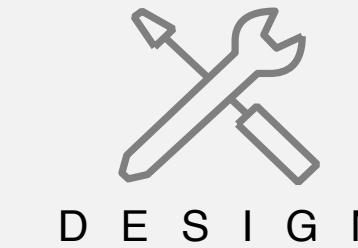
- ☆ Conditional statement

For-Next

- ☆ Repeated iterations

```
For i = 1 To 3
    If Age(i) < 30 Then
        Status = "Young"
    Else
        Status = "Old"
    End If

    MsgBox (Status)
    Next
```



4

Functions

☆ Self-defined
(‘Status’)

☆ System-defined
(‘MsgBox’)

Function Status(Age)

MsgBox



3



THE FINAL PRODUCT



P R O D U C T

1

Some functions

- ☆ Self-defined
(‘Status’)

2

A main *Sub* routine

- (‘Macro’)

Our workflow:

Function Status(Age)

```
If Age < 30 Then
    Status = "Young"
Else
    Status = "Old"
End If
```

End Function

Sub MyCode4()

```
MsgBox (Status(25))
```

End Sub



P R O D U C T

1

Some functions

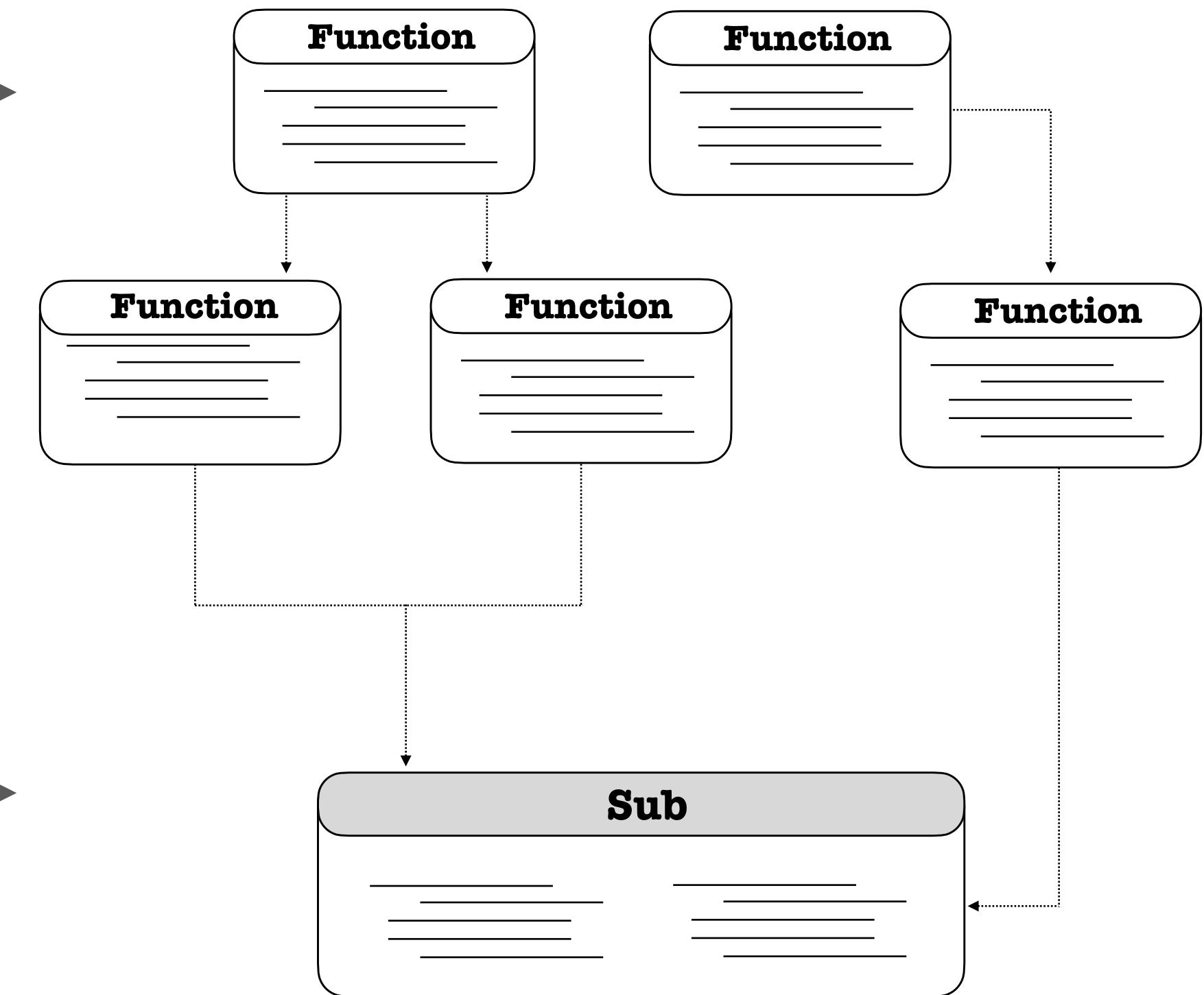
- ★ Self-defined
(‘Status’)

2

A main *Sub* routine

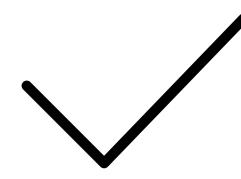
(‘Macro’)

A typical **workflow**:



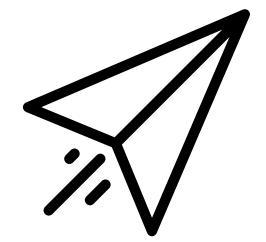
R E M E M B E R

OUR GOALS



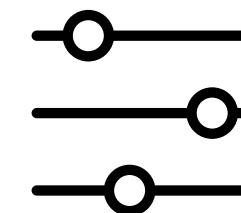
PRECISION

Using variables.
Numbers and text.



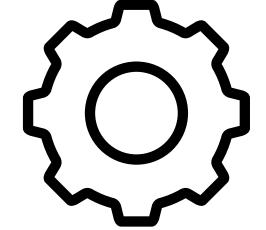
AUTOMATION

Using repeated
iterations.



FLEXIBILITY

Using Functions.



DISCIPLINE

Using Sub routines.