**Python is an interpreted language**

**PYTHON SYNTAX**

“PRINT” = put on the screen 🡪 Then open brackets and write within speech marks, close brackets

EXPONENTIATION = the base to the power of other number e.g 2 \* 2 \*2 => 2\*\*3

ROGUE VALUE = is used to test a program, not a real value, ends the loop

MODULO = returns the value which is the remainder left over e.g ( 5 % 2) leaves 1, will return 1

VARIABLE – Is anything after = sign

= means assign the value to

== means actual “equals”

% =used as a modulo divion, can be used as part of string to mean format the number in a particular way

# = a comment

“”” = multi line comment

TO INDENT = (when defining function) hit space 4 times

RETURN = finishes off the instructions

DEF = used to define a function, so instead of typing a full command e.g

Move 2

Turn 90

Return

() = use brackets when defining a function - empty brackets automatically goes to a string

RESIGNING A VARIABLE = can say the variable is a certain number at one point and then change it later on

PASS = is an instruction which means do nothing

PROMPT (to create a dialogue box)

VARIABLE =INPUT(prompt) 🡪 brings up the dialogue box when it is run

COLOURS OF TEXT – instructions (blue)

Variables (black)

Light Green (text)

Blue (numbers)

Dark green (comment)

Change colour of text 🡪 Tools 🡪 options 🡪 editor options 🡪 syntax colours

SPACE IN VARIABLE = \_ (underscore)

PRINT(TYPE,(enter variable)) 🡪 this shows you what the data type will be whether int, float, string, Boolean

\t = gives two spaces

\t\t = gives a tab

\n =new line

\r – carriage return (takes you back to the beginning)

"%.2f" % 🡪 defines the number of decimal places e.g shows number to two decimal places

**STRINGS AND CONSOLE OUTPUT**

STRING = Regular English phrases, can include numbers, letters, symbols etc.

STRING LITERAL = literally writing it down between quotation marks either “ “ or ‘ ‘

INT = Integar (whole number which is either positive or negative)

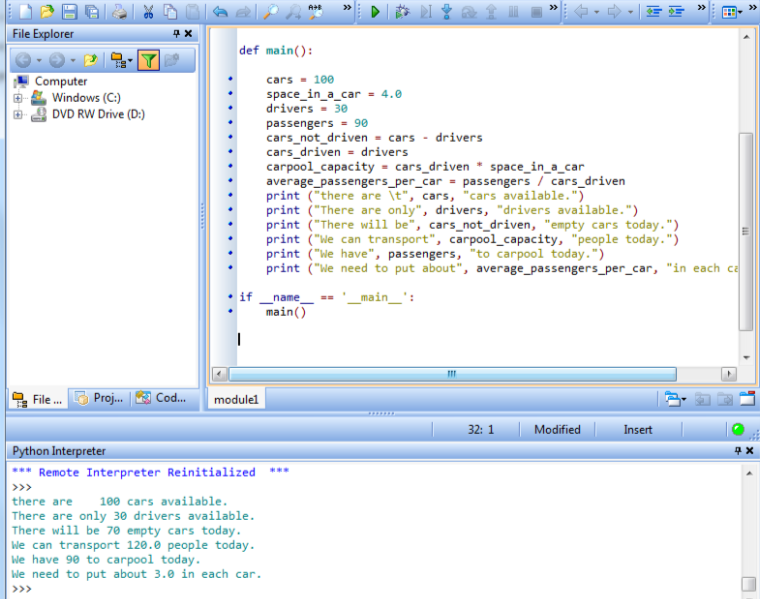
FLOAT = decimal

BOOLEAN = true/false where there are two possible outcomes

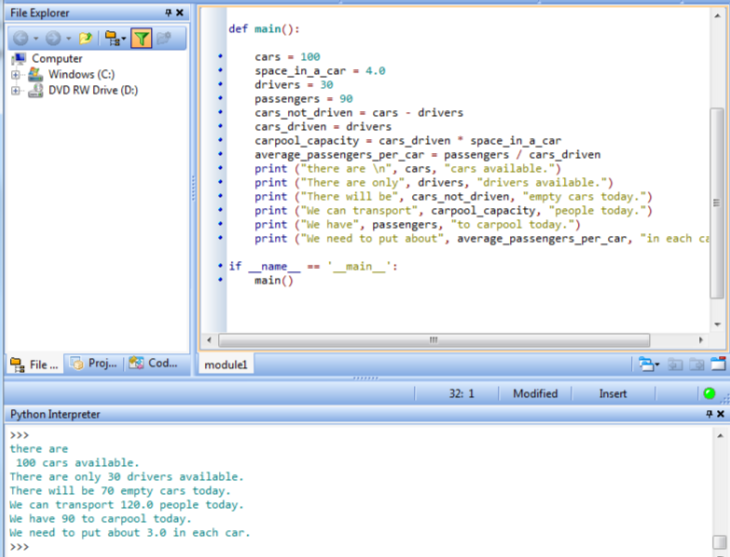
LOOPING STRUCTURE = Wher true/false is used, if the statement is true it will continue to loop if false it will drop down to the next instruction

**Python Exercises**

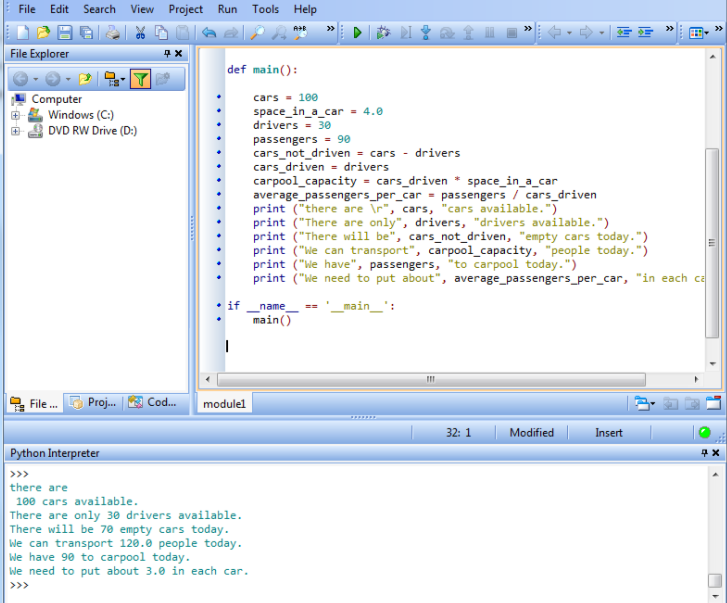
**Exercise 1**



\n = new line

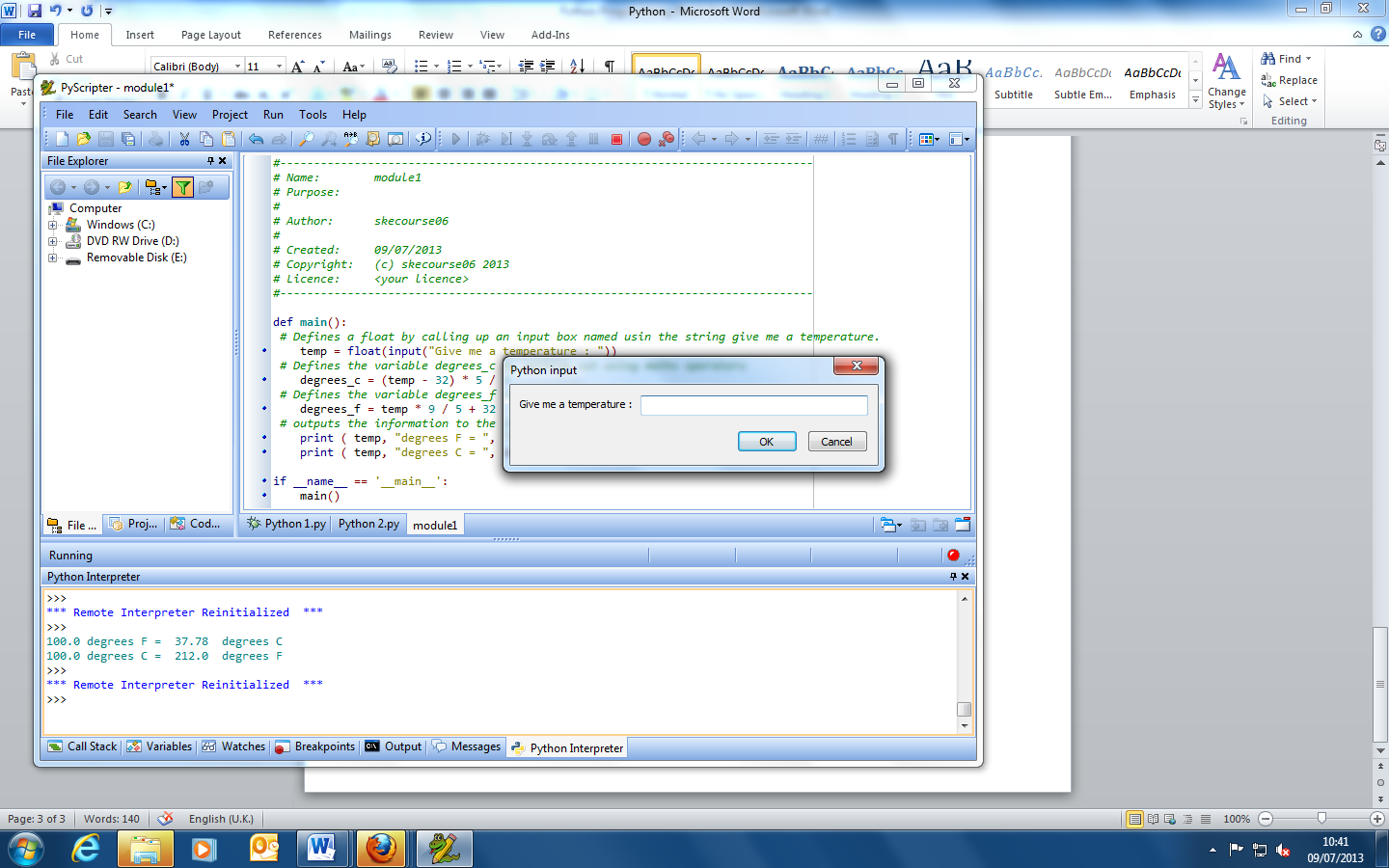


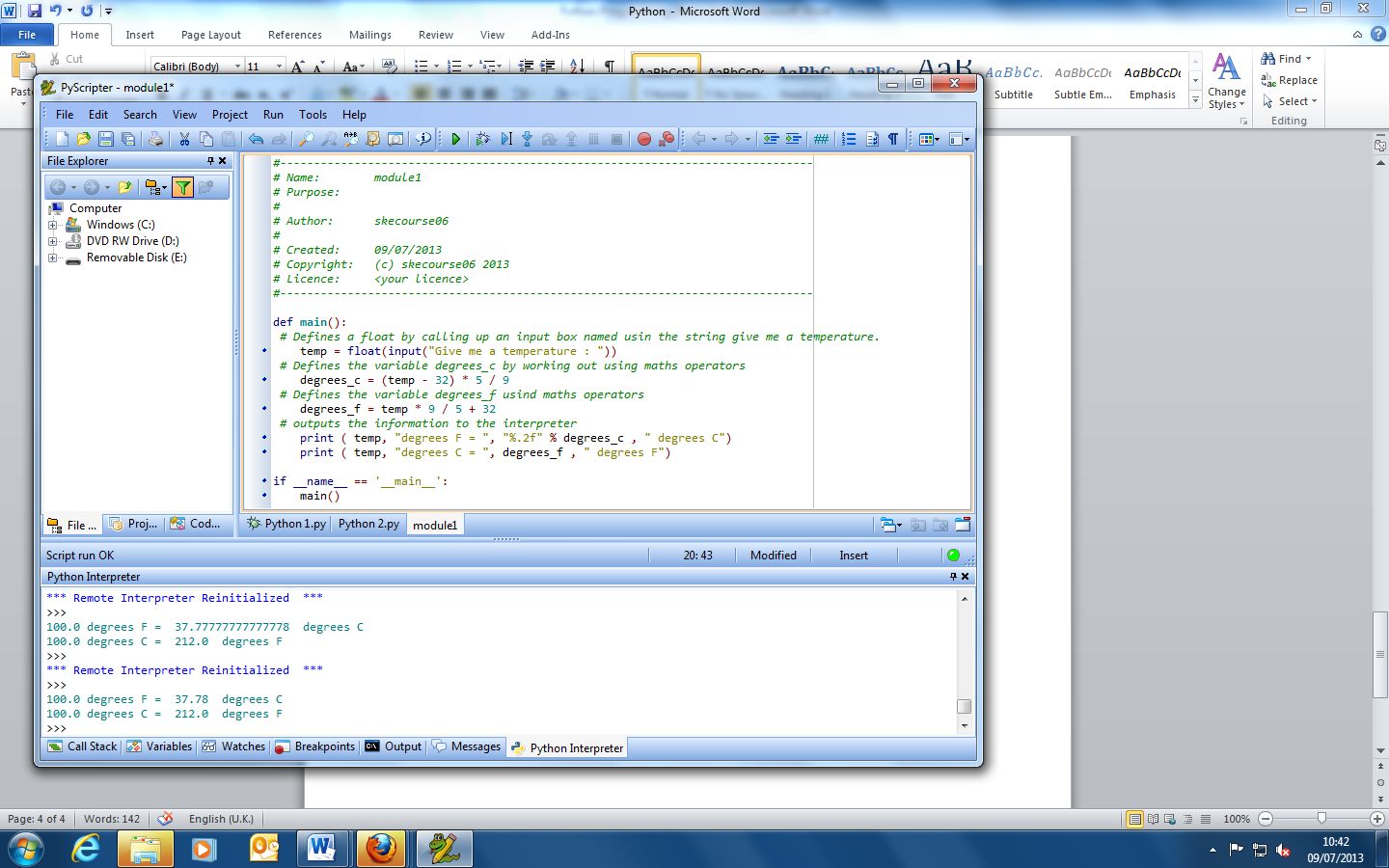
\r = return carriage



\n and \r do exactly the same thing. They are functions left over from typewriters. They are carriage return and line feed. If you ever get a problem with a new line check these functions

**Exercise 2**



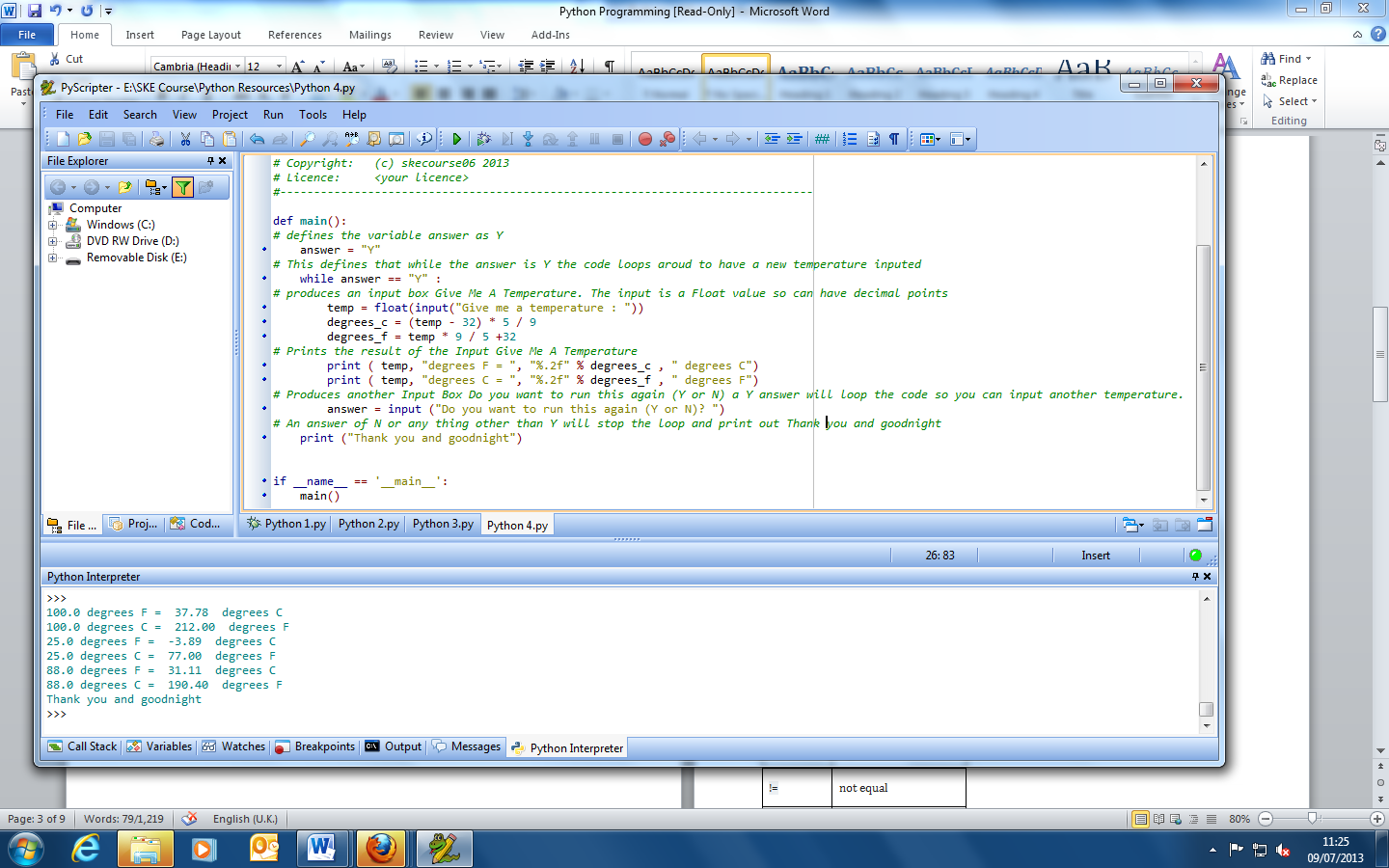


“%.2f” = defines the output to 2 decimal places. What we are saying is format the number in this way.

while repeats a section of code as long as the Boolean value of answer is true (Y). As long as the state of while answer == Y more temperatures can be inputed.

while 1: is an infinite loop and can’t be broken

while 0: completely bypasses the route



Exercise 4

1. Type this into PyScripter and run it. Test the program with positive numbers

Note that to use Maths functions that are not listed in the table above, you have to IMPORT a MATHS module. See line 1

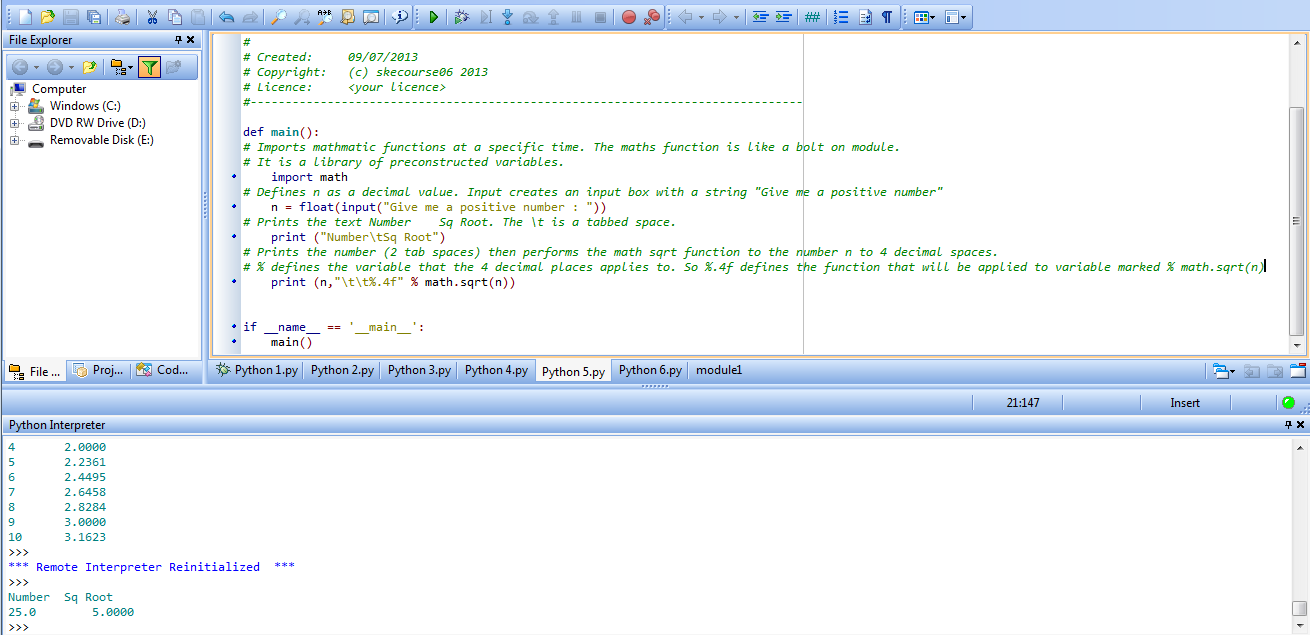
2. Write a comment above each line explaining what it does.

import math

n = float(input("Give me a positive number : "))

print ("Number\tSq Root")

print (n,"\t\t%.4f" % math.sqrt(n))



3. Type this into PyScripter and run it.

a. NB import math has moved – does it still work? Try moving it to the end – does it still work?

b. Also note the subtle change to the print line. What difference does it make?

4. Type this into PyScripter and run it.

5. Write a comment above each line explaining what it does.

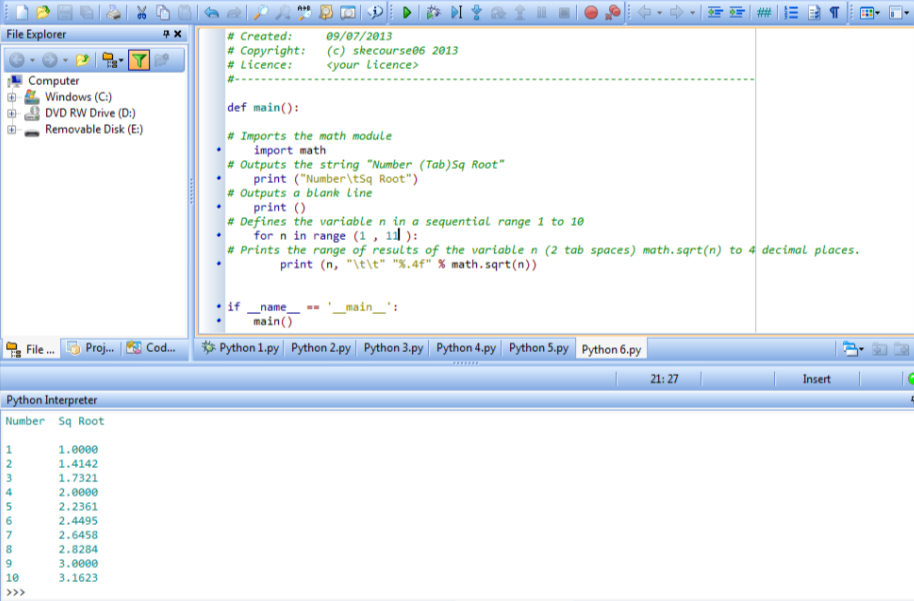
import math

print ("Number\tSq Root")

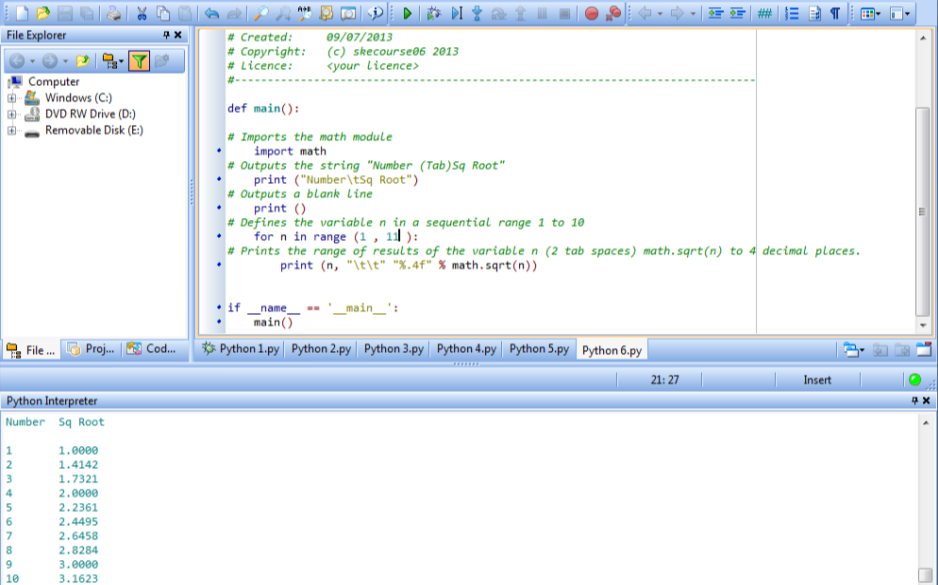
print ()

for n in range (1 , 10 ):

print (n, "\t\t" "%.4f" % math.sqrt(n))



6. Experiment with different numbers in line 4.



Exercise 5

1. Type this into PyScripter and run it. Test the program fully with suitable test data. You can only truly test it if you know what the expected outcomes are!

2. Write a comment above each line explaining what it does.

print(" Discount Calculator. There is 12.5% discount on items over £60")

cost=float(input("Ticket price"))

if cost > 60 :

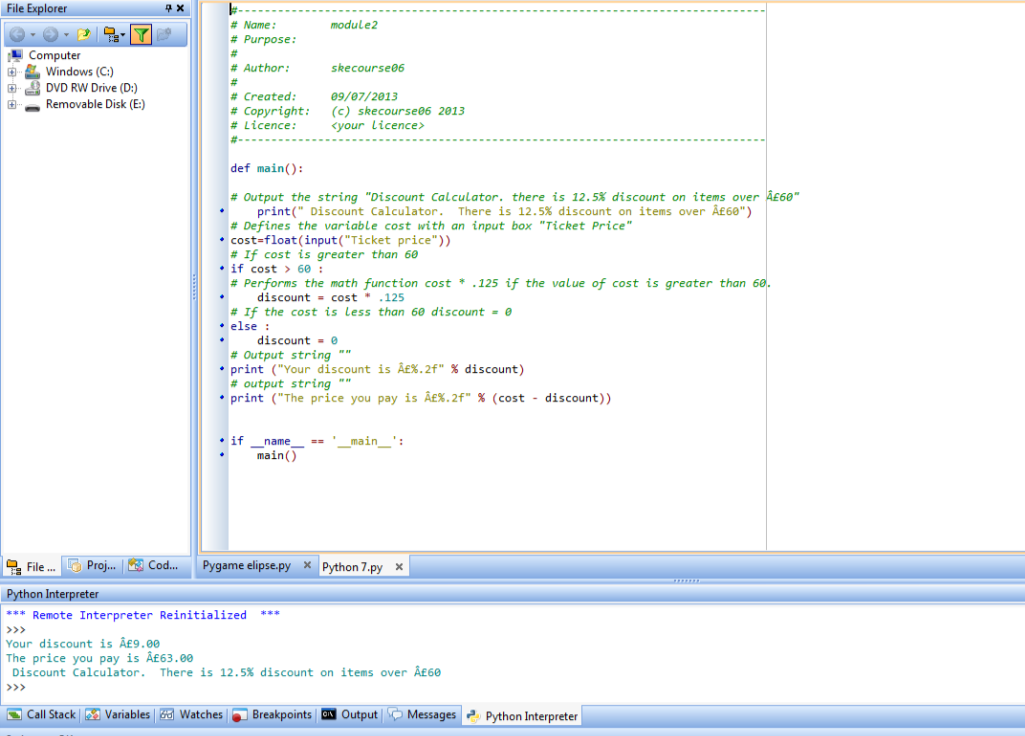
discount = cost \* .125

else :

discount = 0

print ("Your discount is £%.2f" % discount)

print ("The price you pay is £%.2f" % (cost - discount))

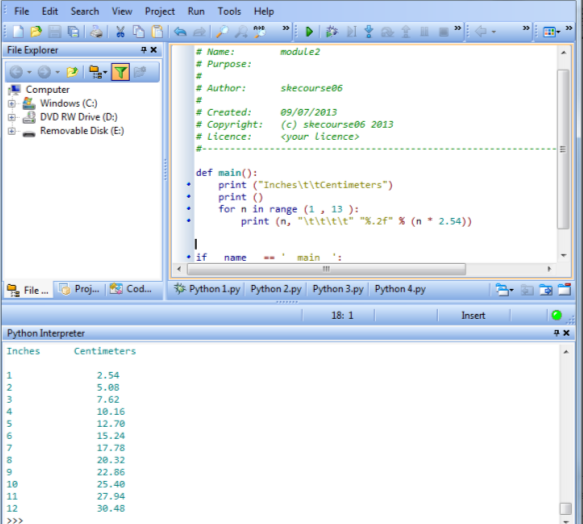


Exercise 6

1. Write a program to produce a conversion table from inches to centimetres (1” is 2.53 cm ). The table should list inches from 1 to 12 in steps of 1.

2. Adjust the above program to one where the user inputs the start number of inches and the end value.

3. Adjust the program in 2 to go from 1 to 12 in steps of 0.1”



Exercise 7

Write a program to grade percentage test scores which the user enters during execution. To stop the program, the user will enter a negative score. The grades boundaries are:

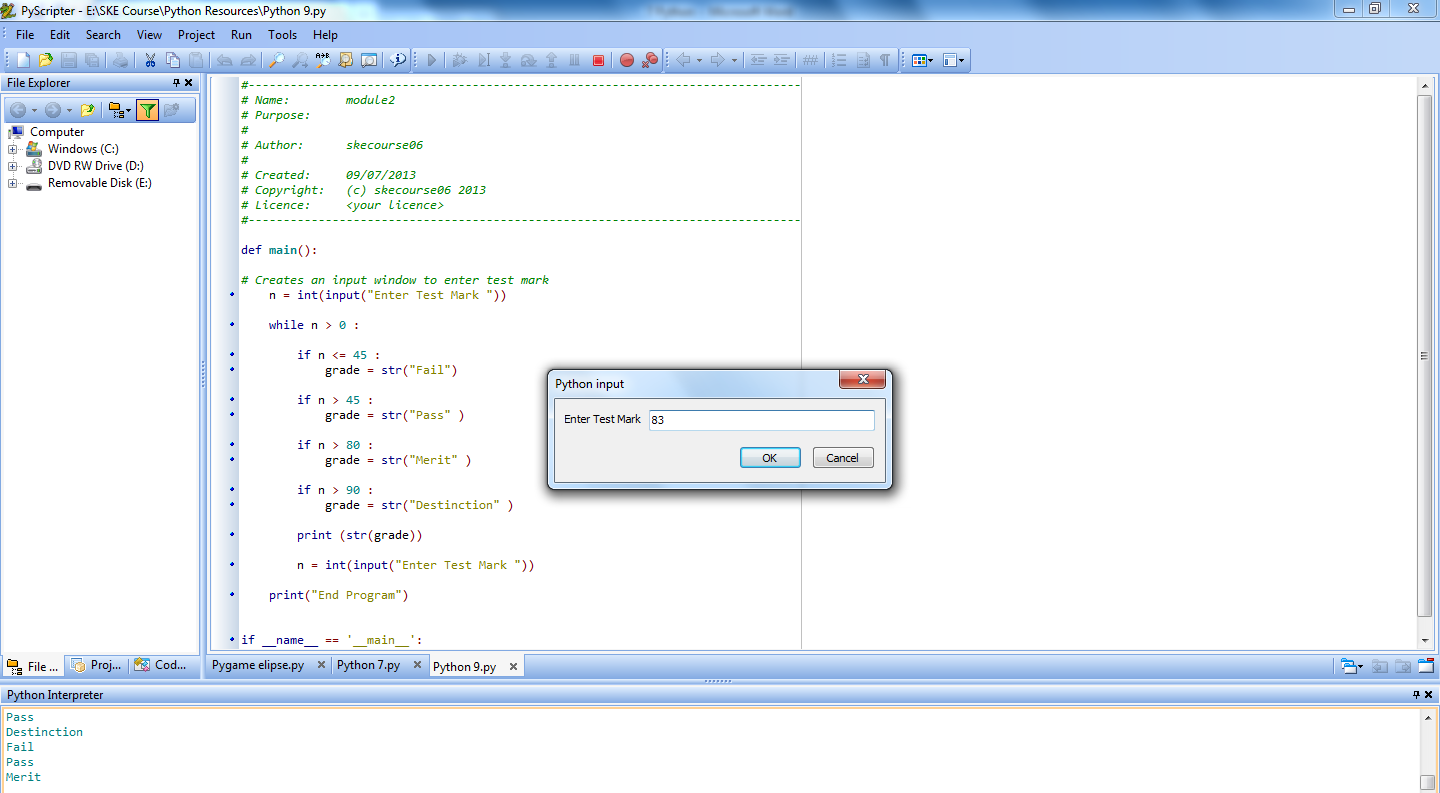
Grade Mark

Distinction >90

Merit >80

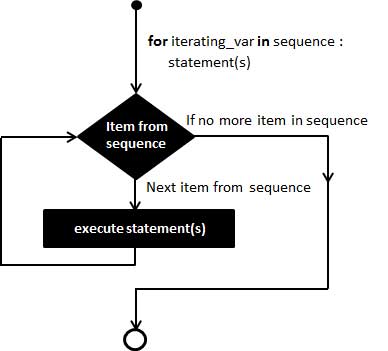
Pass >45

Fail <=45



Looping structures

For:



While:

