Programming for Science
Workshop 1

This workshop aims to give you some experience of using the Python interpreter and the Notepad++ editor for writing straightforward Python programs. If you have any difficulties or don't understand what you are supposed to be doing or why you are doing it, please consult one of the people taking the workshop.

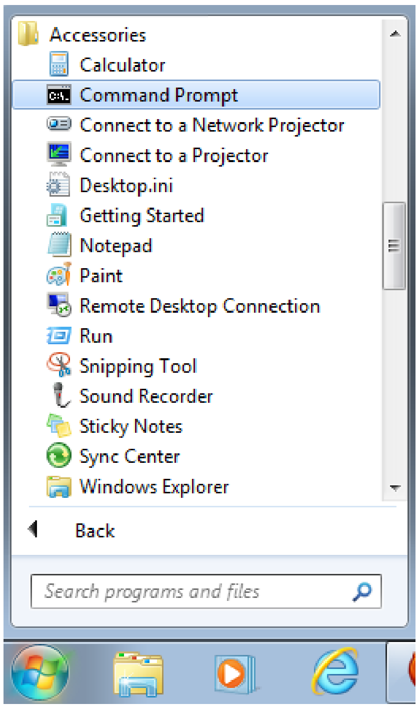
If the photocopy doesn't reproduce the pictures well, you can look at the PDF version from the ELE page: `http://empslocal.ex.ac.uk/studyres/ECM1408`.

1 Hello Python

To start we'll get the Python interpeter in a command window (a.k.a terminal) and print a string with it. Follow these steps

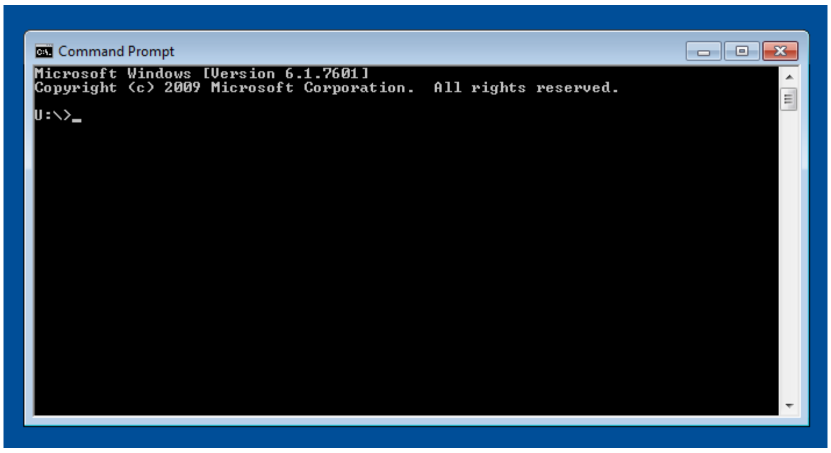
1. Login to Windows

2. Get a command window by selecting "Command Prompt" from the "Accessories" section of the Windows "Start" menu in the bottom left hand corner:



You might prefer to use "Console" (main section of the Start menu) which is a better, more configurable alternative to the standard Command Prompt. We'll use the standard Command Prompt here.

3. You should get a screen with the > prompt.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

U:\>_
```

Note that this is not the Python prompt, but the Windows prompt. The text before the prompt (`U:\` in the picture) tells you the "drive" (or device) and the directory (folder) that you are in. In the picture, the drive is the U drive and the directory is the top directory there, indicated by the '\'.

A short digression on the console: The console or command window is a way of interacting with the computer. It allows you to invoke programs by typing their names, and is the traditional way of interacting with computers. We are going to use it because it makes clear the distinction between the computer (running Windows), the interpreter (Python), and a Python program itself.

A couple of useful commands for the console are:

dir This shows you the files in the directory or folder that you're currently in. You're probably used to looking at these with Windows Explorer, but the same information is (less prettily) available with dir.

cd Short for "change directory" allows you to change your current directory to another. If in your current directory you have a sub-directory named `programming`, you can change to it with the command `cd \programming`. You could then see what files are in it with the command `dir`

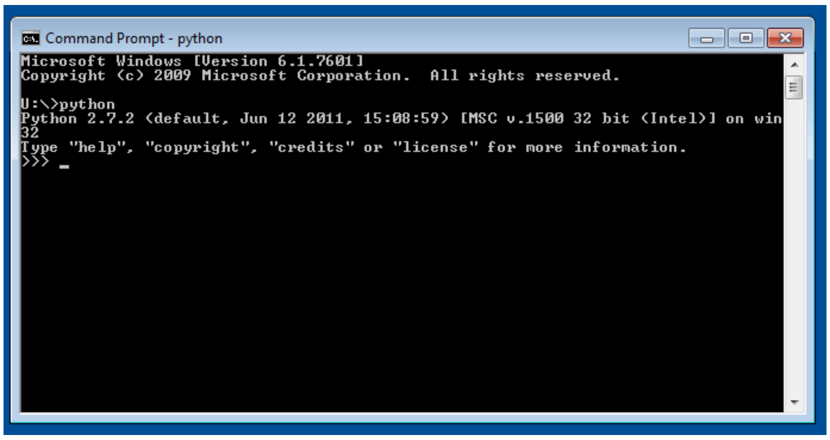
Please make sure you distinguish between commands to Windows (via the console) and commands/statements for the Python interpreter. Commands for one won't work for the other. As a reminder, the prompts are:

>	Windows console
>>>	Python interpreter

4. At the command prompt, type:

```
python
```

to invoke the Python interpreter. You should see the few lines giving details about the interpreter and then the python prompt; something like:



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

U:\>python
Python 2.7.2 (default, Jun 12 2011, 15:08:59) [MSC v.1500 32 bit (Intel)] on win
32
Type "help", "copyright", "credits" or "license" for more information.
>>> _
```

5. At the Python prompt (`>>>`), type a single statement:

```
print 'Hello, world!'
```

(Of course, you don't type the `>>>`.)

6. Try entering and evaluating some statements at the interpreter, using it as a glorified calculator. If you get an exception, make sure you know why.

7. Enter the following statements to create some variables:

```
radius = 26.0
message = 'This parrot has ceased to be'
count = 15
```

Check that the variables have the value you expect and check their types with the `type` function (e.g. `type(radius)`).

If you are in a workshop in the early part of the week, we may not have talked about string variables (such as message). Briefly, a string is a sequence of characters created by putting the characters between matching quotes and you can concatenate strings with the `+` operator. Thus

```
>>> name = "Richard"
>>> greeting = 'Hello there'
>>> name
'Richard'
>>> greeting
'Hello there'
>>> print greeting, name
Hello there Richard
>>> both = greeting + name
>>> print both
Hello thereRichard
>>> # Concatenate greeting, a string consisting of a
>>> # single space and name
>>> both = greeting + " " + name
>>> print both
Hello there Richard
```

You can also repeat strings with the `*` operator:

```
>>> greeting = "Hello "
>>> print greeting*3
Hello Hello Hello
```

Please see the lecture slides for more to do with strings.

Execute the following statements, but before you do decide what you expect the result to be; if there is a difference make sure you understand why:

```
radius*2 + 5
count*2 + 10
count/10
count/20
count/20.0
countSquared = count**2
print countSquared
x = 1/count**0.5
print message + '!'
print message + '!'*count
count = count + 5
   count = count - 5 # Note the indentation
radius/(count-20)
```

If you're doing this before Monday's lecture, note that when one integer is divided by another Python rounds the result down to the nearest integer. This apparently strange behaviour is to make sure that the result is an integer, but can be surprising if you're not expecting it.

8. Invent some more expressions or try out some from the lecture slides. Try anything: the worst that can happen is an exception.

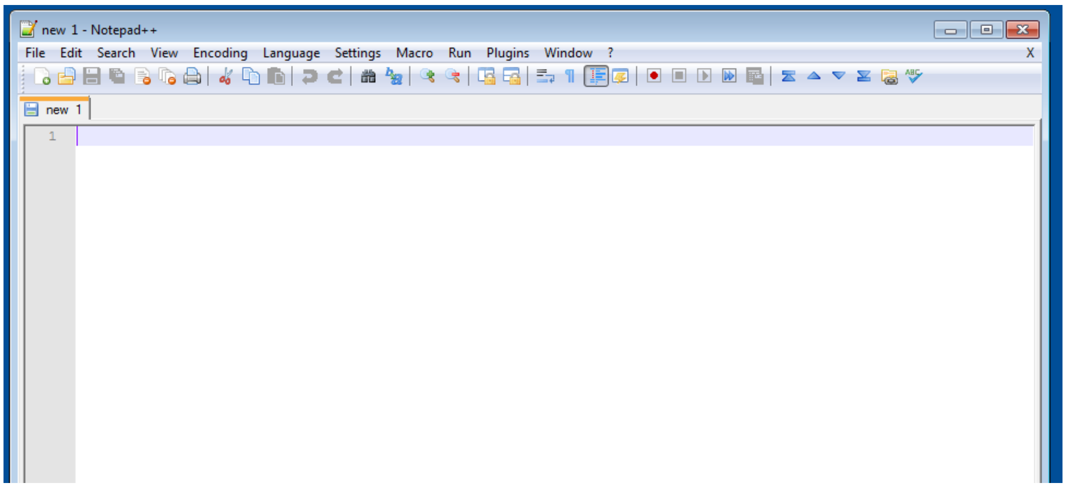
9. Close the interpreter by typing `quit()` or `exit()` or `^D` (control-D).

2 Writing a program with an editor

Here we'll write a couple of small programs using the Notepad++ editor. Never use Word for editing Python programs, it's not at all suited to them. Notepad++ is installed on the machines in Harrison and you can get it for free for your own computer; see the ELE page for more information on editors.

1. Start Notepad++ by selecting it from the "Start" menu.

2. You should get a screen that looks like:

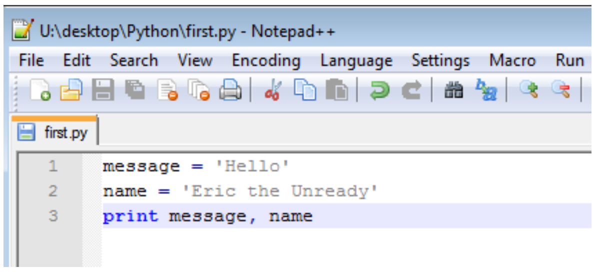

new 1 - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1

3. In order ensure that Notepad++ treats your program as a Python program, use the “Save As” menu to save the (empty) file as a Python file (say `first.py`). The `.py` extension is important, but you can choose whatever name you like; it’ll be (much) less convenient to choose a name with spaces in it. Make sure you know which folder you saved the program in. It’s probably best to make a new folder for your work. Having done this once, Notepad++ will treat all your files in this session as Python; if you open a file with the `.py` extension, it will automatically be recognised as Python.

4. We’ll start by writing another very short program. Use the editor to write these three lines (but use your own name instead of ’Eric the Unready’):

```
message = 'Hello'
name = 'Eric the Unready'
print message, name
```

In Notepad++ your program should be something like:


U:\desktop\Python\first.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run
first.py

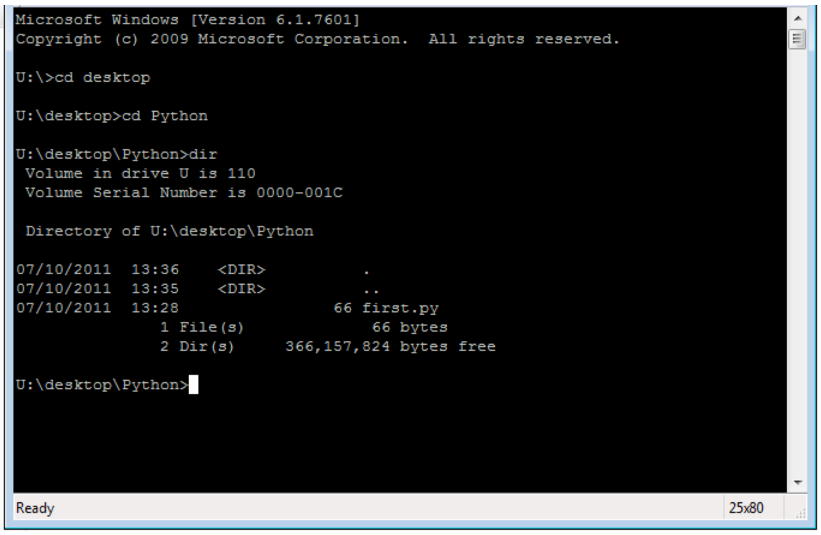
```
message = 'Hello'
name = 'Eric the Unready'
print message, name
```

Notice that the editor colours the variables, strings and keywords (`print`) differently.

5. Save your program to a file.

6. In the command window, navigate to the directory where you saved the program and check that it’s there. (Recall the notes on `cd` and `dir`.)

Here’s an example where the file `first.py` is saved in the directory `\desktop\Python` on the `U:` drive; you can choose a different directory, of course.



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

U:\>cd desktop

U:\desktop>cd Python

U:\desktop\Python>dir
 Volume in drive U is 110
 Volume Serial Number is 0000-001C

 Directory of U:\desktop\Python

07/10/2011  13:36    <DIR>          .
07/10/2011  13:35    <DIR>          ..
07/10/2011  13:28                66 first.py
               1 File(s)             66 bytes
               2 Dir(s)     366,157,824 bytes free

U:\desktop\Python>
```

Ready 25x80

7. Execute the program by typing (at the command prompt (>) in the console):

```
python first.py
```

Is the output what you expected?

8. Introduce a deliberate syntax error (eg, an indentation error) into your program, save the file and run it again. What happens? Make sure you understand the exception that is thrown. Correct the error and check the program works again.

9. Write a program `fahrenheit.py` that converts temperatures from Celsius to Fahrenheit. In case you've forgotten, degrees Fahrenheit are obtained by multiplying degrees Celsius by 9/5 and adding 32. Your program should start with a variable that is initialised to the temperature in Celsius; then set another variable equal to the temperature in Fahrenheit and print the result (with an explanatory message).

10. Finally, we'll write a program to calculate the roots of the quadratic equation $ax^2+bx+c=0$. Recall that the roots are given by the formula:

$$x = \frac{-b \pm \sqrt{b^2-4ac}}{2a}$$

Edit the following code using Notepad++. In Python everything on a line following a # character is a comment; it is ignored by the interpeter. In complex programs it's very important to comment the code so that others (and you) know what it's doing. Try to get into the habit of commenting each program with a line or two saying what it does and commenting the tricky or interesting bits of the code.

```
# Calculate the roots of ax**2 + bx + c = 0

# Set the coefficients (make sure they're floats)
a = 1.0
b = -6.0
c = -2.0

# Find the roots
```

```
discriminant = b*b - 4*a*c
# Take the square root
discriminant = discriminant**0.5  # Problems if negative!

xplus = (-b + discriminant)/(2*a)
xminus = (-b - discriminant)/(2*a)

print 'The roots of ax**2 + bx + c = 0 with'
# Comma at end of print statement prevents new line
print 'a =', a, ' b =', b, ' c =', c,
print 'are', xminus, xplus
```

3 Install Python on your own computer

If you've brought your own laptop, it may be a good idea to do this in the workshop; if not, please install Python at home.

Follow the instructions on the ELE page, `http://empslocal.ex.ac.uk/studyres/ECM1408`, to install Python on your own machine if you haven't got it. I highly recommend the Enthought Python Distribution: it's relatively easy to install and comes with some extras that we'll need later. Even if you have a Mac that comes with Python, this is probably worth installing.

If you haven't also got a good editor, it's highly recommended that you install one such as Notepad++ on Windows or TextMate on Macs. Editra is reasonable and works on both Windows and Macs. See the ELE page for download links.

4 Exercises

Here are some questions and exercises that **you** should do. You may well have time to complete them in this workshop, but if not make sure you finish them by the next workshop. Model answers will be on the ELE page at the end of the week.

1. What is the difference between `a = 17` and `a = '17'`?

2. Write down what you think the result of executing each line of the following program in turn should be.

```
a = 17
b = '17'
print 3*a
print 3*b
print a*b
c = a + b
```

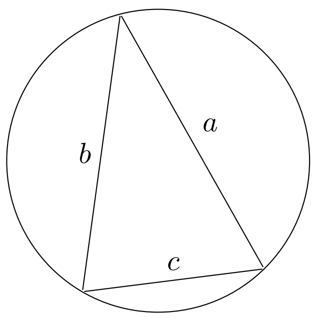
Now try it with the interpreter.

3. The diameter d of a circle that passes through the vertices of a triangle whose edges have lengths a, b and c is given by:

$$d = \frac{abc}{2\sqrt{s(s-a)(s-b)(s-c)}}$$

where s is called the semiperimeter of the triangle and is defined as

$$s = (a + b + c)/2$$



Write and test a program `circumcircle.py` that calculates and prints the radius of such a circle, given the edge lengths as variables `a`, `b` and `c`. What is the radius of the circumcircle of a triangle with edges 4, 6 and 3? Make sure you lay the program out well (no prizes for the most concise program) and comment it.

Note:

- You can calculate the square root by importing it from the math library (see the lecture slides) or you could use exponentiation with a power 0.5.
- The actual result is not very interesting. This is an exercise in writing and running a straightforward Python program.