Preparations for Python Courses

From July some of our Python courses will be conducted online. The courses are all example led and these examples will be available after the course using Jupyter Notebook (see below). Jupyter Notebook is browser based and it would be beneficial if you familiarise yourself with using the product before the course starts. We will also make limited use of the command line, so make sure you know how to obtain a command prompt.

However, although not required for attending the course, you may prefer to develop your Python programs using other means. There are many ways to run Python code to suit most tastes, from the simple command line interpreter to a full blown Interactive Development Environment such as Dawn. This guide introduces you to some of the more common methods. The trainer will be using both Jupyter Notebook and Eclipse on the course, but as stated previously you will be able to follow along with just Jupyter Notebook.

To begin with we need to configure Python and be able to execute a simple script. At Diamond, configuring Python is achieved using the module load command:

```
module load python/3.7
```

To see that this works you need to run a simple script. Type the following script into a file:

```
x = 100
y = 200
z = x + y
print(z)
```

and try one of the methods below to run this script.

Indentation is significant in Python scripts, so make sure each line of the above script starts in column one.

Note that Python 2 has reached end of life and therefore we will be using Python 3 throughout the course and in this guide.

1. Running Python from the command line

After you have issued the module load command, to start up Python 3 type:

```
python
```

and you will get to a python prompt. This is called the REPL. Simply enter python commands at the prompt (or type each line of the above script). If you want to see the value of a variable you can just type the name of the variable; there is no need to use a print statement. This is the simplest way to run Python, but becomes unwieldy for larger programs.

To exit, type:

```
exit()
```

However, you will loose all your code when you exit the REPL.

2. Using Jupyter Notebook

This is a much better approach. You can easily save your work and you can easily check for errors in your code. You can even convert a notebook into a presentation.

Launch the Jupyter notebook with:

```
jupyter notebook --browser=firefox
```

The Jupyter notebook lets you create multiple browser consoles to enter Python code. To find out more try the website:

```
https://realpython.com/jupyter-notebook-introduction/
```

3. Using the Python Debugger on the Command Line

Create your Python script (e.g. myscript.py) with a text editor and then run your script using:

```
python -m pdb myscript.py
```

Once the debugger starts you can enter debug commands. Use ? to get a list of commands. The most useful commands are:

```
l    list the contents of the script
n    execute the (n)ext line
s    (s)tep into functions
r    (r)eturn from a function
b 5  breakpoint on line 5
c    continue to next breakpoint
```

The debugger is quite good, but its probably better to use pudb as the debugger (see below).

4. Using pudb to Debug on the Command Line

Install pudb:

```
pip intall pudb --user
```

Then type:

```
python -m pudb myscript.py
```

A simple graphical interface will appear to allow you to debug your program. Commands are essentially the same as pdb, but you get a nice graphical representation of your source code. Type ? to get a list of commands.

5. More Sophisticated Environments

If you want a full blown Interactive Development Environment (IDE) you can use one of:

```
PyCharm
Eclipse
Dawn
```

However, be warned that these IDEs are quite complicated with lots of menu options and many windows. There are lots of guides on how to use these IDEs on the internet. Note that Dawn is essentially a variant of Eclipse which has been customised by Diamond. Many people find these IDEs too complicated for their taste; they were developed for professional software engineers. You need module load commands for each of these IDEs. PyCharm comes in two flavours: community and professional. Note that you need a licence to run PyCarm Professional. The community edition is the default at Diamond.

To run one of these IDEs type one of the following pairs of commands.

```
module load dawn
dawn&
```

```
module load eclipse
eclipse&
```

```
module load pycharm
pycharm&
```