Lesson 1

How to install Anaconda in your computer

Click on the following link:

https://www.anaconda.com/products/distribution#Downloads

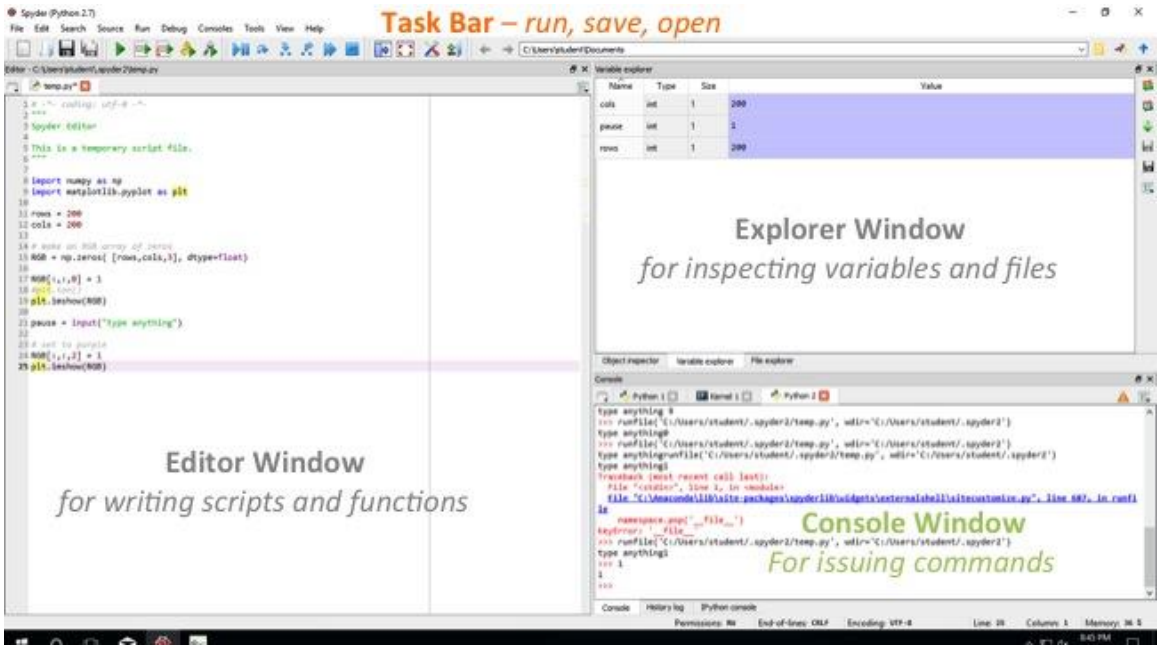
Then download the Anaconda Installer:

For Windows: **64-Bit Graphical Installer (594 MB)**

For MacOS: **64-Bit Graphical Installer (591 MB)**

Starting Out in Python using Spyder

Opening Spyder: from the START button, click on "Anaconda" then scroll down and click on "Spyder". Wait for it to start, and you'll see something like this:



The three windows do the following:

CONSOLE WINDOW – you can issue python commands one at a time, and see the results. The ipython console is a little more interactive than the regular console.

EDITOR WINDOW – here is where you will write scripts and functions, to be run as a block. The editor will make many of your tasks easier by imposing the right spacing, highlighting commands, pointing out simple errors, etc.

EXPLORER WINDOW – Lets you inspect variables and files in your directory.

The TASK BAR is fairly normal for most applications, though it has the several other buttons for running scripts and functions, and doing code debugging.

ALWAYS VIEW PYTHON FILES IN THE EDITOR. If you open a file in a normal word processing program, it will add hidden keywords and formatting that will make it unusable for coding. If you must view python files outside the editor, use Notepad++ in Windows, Textwrangler in Macintosh. (Other text readers can be used if you know how to set the preferences correctly.)

Python Basics

Variable Types, Lists, Character String Manipulation

Analyze what happens at the python command line:

```
>>  4 + 3                          >> variable1 = 4
>> 4/3                             >> variable2 = 3
>> 4.0/3                           >> variable1 + variable2
>> float(4)/3                      >> float(variable1)/variable2
>> 4*3                             >> new_variable = variable1 + variable2
>> 4**3                            >> print( new_variable )
```

What happens if you try this: >> 4 = variable1??

This is why "=" is called "assignment" rather than "equals"

Working with character strings:

```
>> first_name = 'Said'
>> last_name = 'Mejia'
>> first_name + last_name
>> MyName = first_name + "  " + last_name
>> print( MyName )
>> age = '52'
>>print( MyName+age )
>> age = 52
>> print( "Hello, my name is " + MyName + " and my age is "  + age ) #Error
>> print( "Hello, my name is " + MyName + " and my age is "  , age ) # , to define print integer numbers
>> print( "Hello, my name is " + MyName + " and my age is "  + str(age) )
>> print( "Hello, my name is %s and my age is %d"  %(MyName, age) )
```

Variable Types:

we will be working with integers, floats, and character strings. You may need to convert between them:

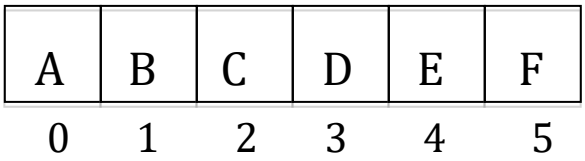
```
>> flt5 = float(5)          >> int5 = int(5.0)          >> str5 = str(5.0)
```

Lists and Indexing

```
>> mylist = ['Said',26,'New York']
```

```
>> print( mylist[0], mylist[1], mylist[2] )

>> print( “my name is”,mylist[0], “ and I live in “,mylist[2] )

>> mycities = [‘Houston’,’Baltimore’,’ Washington DC’]

>> print( mylist + mycities )

>> mycities = mycities + [‘New York’]

>> mycities[2] = ‘Washington DC’

>> mylist[1] = mylist[1]+1

>> mycities[1:4]

>> mycities[1:5]   # what’s going on??

>> len(mycities)   # this length function can be useful!
```

Slicing



The indices used in a list (and later in a numerical array) should be thought of as labelling the *boundaries* of each cell. If only one index is used, it is assumed to refer to the item in the cell to the right. If two indices are used separated by a colon, the list will be sliced into the region between the boundaries. So let’s give it a try:

```
>> testlist = [‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’]

>> print( testlist[2] )

>> print( testlist[2:3] )  # can you explain?

>> print( testlist[2:6] )

>> stringvar = ‘New York’

>> print( len(stringvar) )

>> print( stringvar[2] )

>> print( stringvar[4:8] )              # character strings behave like lists!
```