

Python Introduction

Introduction

This file is a quick guide to installing and the basic usage of Python. It is based on guide developed by Truyen Tran (Deakin University). It is recommended that students unfamiliar with Python work through this document.

Python Overview

Python (<http://www.python.org/>) is a modern, general-purpose, object-oriented, high-level programming language.

General characteristics of Python:

- clean and simple language: Easy-to-read and intuitive code, easy-to-learn minimalistic syntax, maintainability scales well with size of projects.
- expressive language: Fewer lines of code, fewer bugs, easier to maintain.

Technical details:

- dynamically typed: No need to define the type of variables, function arguments or return types.
- automatic memory management: No need to explicitly allocate and deallocate memory for variables and data arrays. No memory leak bugs.
- interpreted: No need to compile the code. The Python interpreter reads and executes the python code directly.

Advantages:

- The main advantage is ease of programming, minimizing the time required to develop, debug and maintain the code.
- Well designed language that encourage many good programming practices:
 - Modular and object-oriented programming, good system for packaging and re-use of code. This often results in more transparent, maintainable and bug-free code.
 - Documentation tightly integrated with the code.
- A large standard library, and a large collection of add-on packages.

Disadvantages:

- Since Python is an interpreted and dynamically typed programming language, the execution of python code can be slow compared to compiled statically typed programming languages, such as C and Fortran.
- Somewhat decentralized, with different environment, packages and documentation spread out at different places. Can make it harder to get started.

Python Versions

There are two versions of Python that are being used by the community: Python 2 and Python 3. Python 3 will eventually supersede Python 2, and it is not backward-compatible with Python 2.

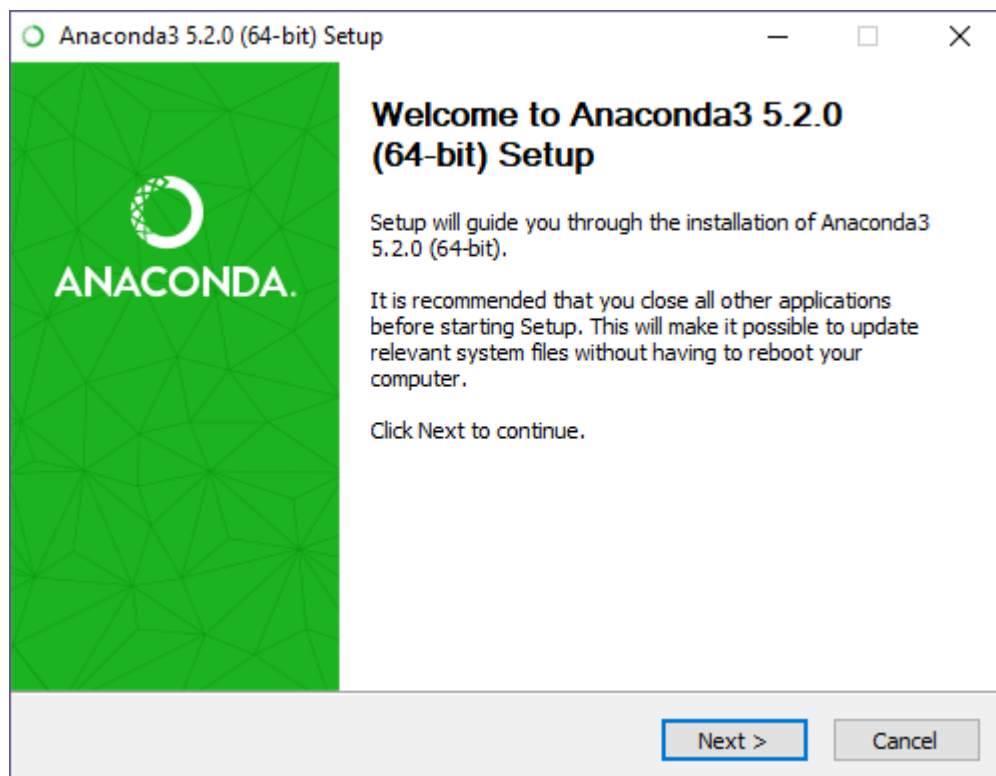
We will be using Python 3 (3.6 to be more specific) in this course.

Installing Python

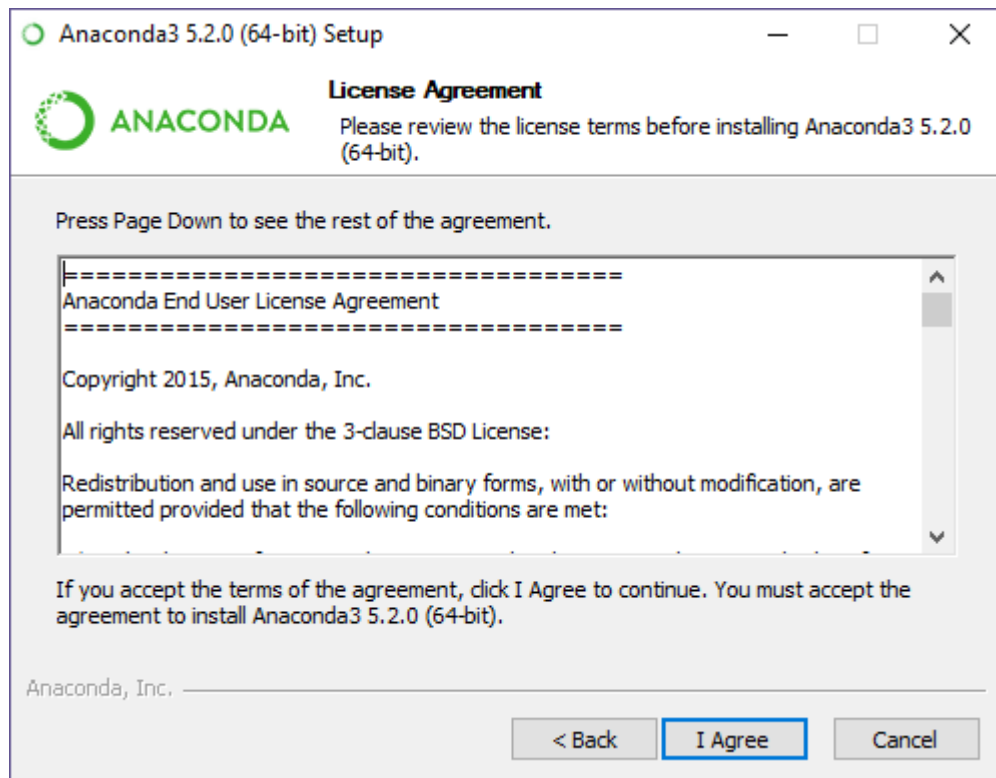
Note: If you are using a Deakin University PC, you can skip this section as Anaconda with Python 3.6 is already installed.

There are many ways to install Python and necessary modules for this course, but the most straightforward way is using a scientific distribution such as Anaconda Scientific Distribution (<https://store.continuum.io/cshop/anaconda/>). We highly recommend you to use anaconda because it comes with most of the modules you need for the course, e.g IPython, Jupyter, numpy, matplotlib, and Spyder and you don't need to install them individually. Follow these steps to install anaconda and **please pay attention to step 7 as you will encounter some problems if you miss it:**

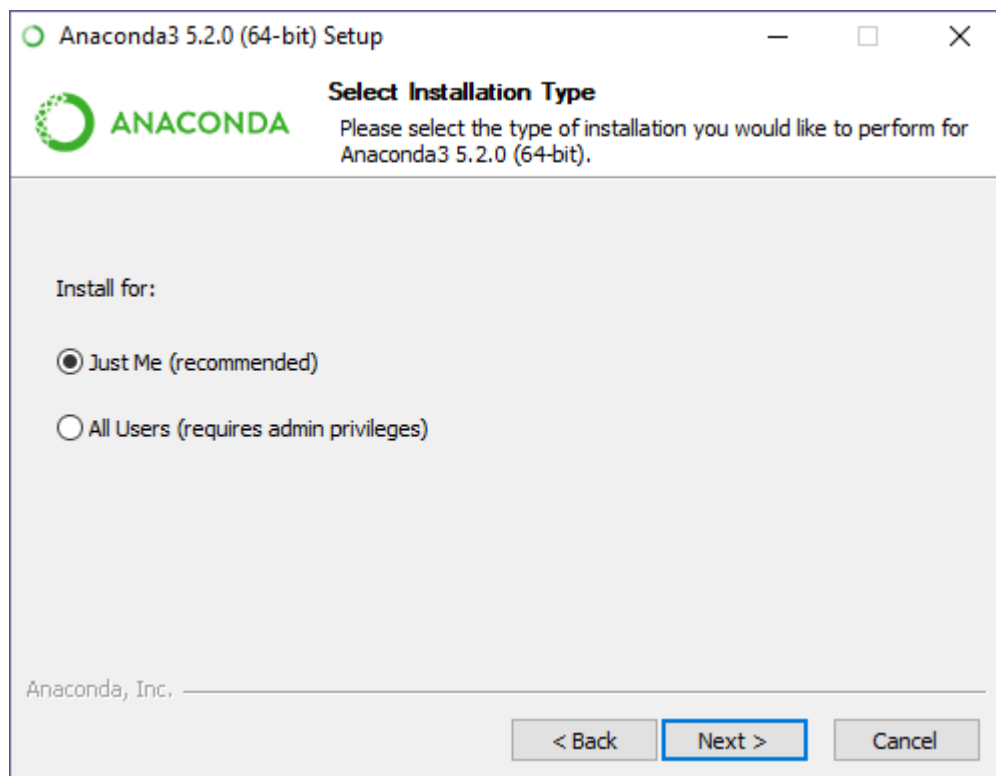
1. Download Anaconda with Python 3.6 compatible with your OS from <http://continuum.io/downloads>.
2. Double click on the downloaded file.
3. Click on Next



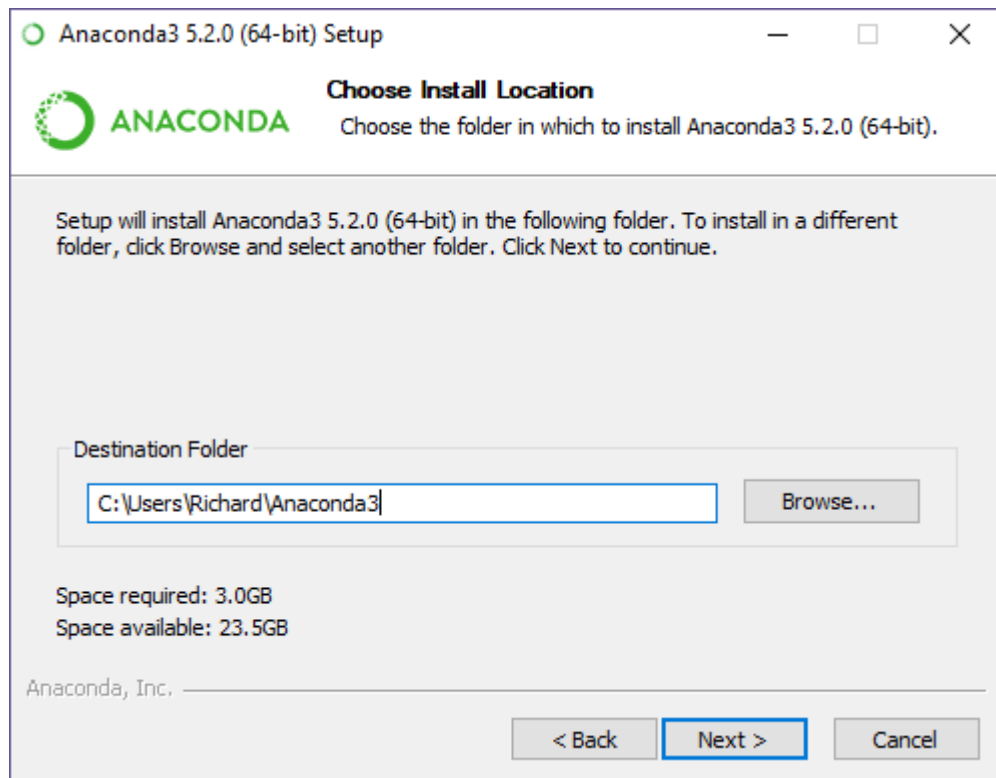
4. Click on "I Agree"



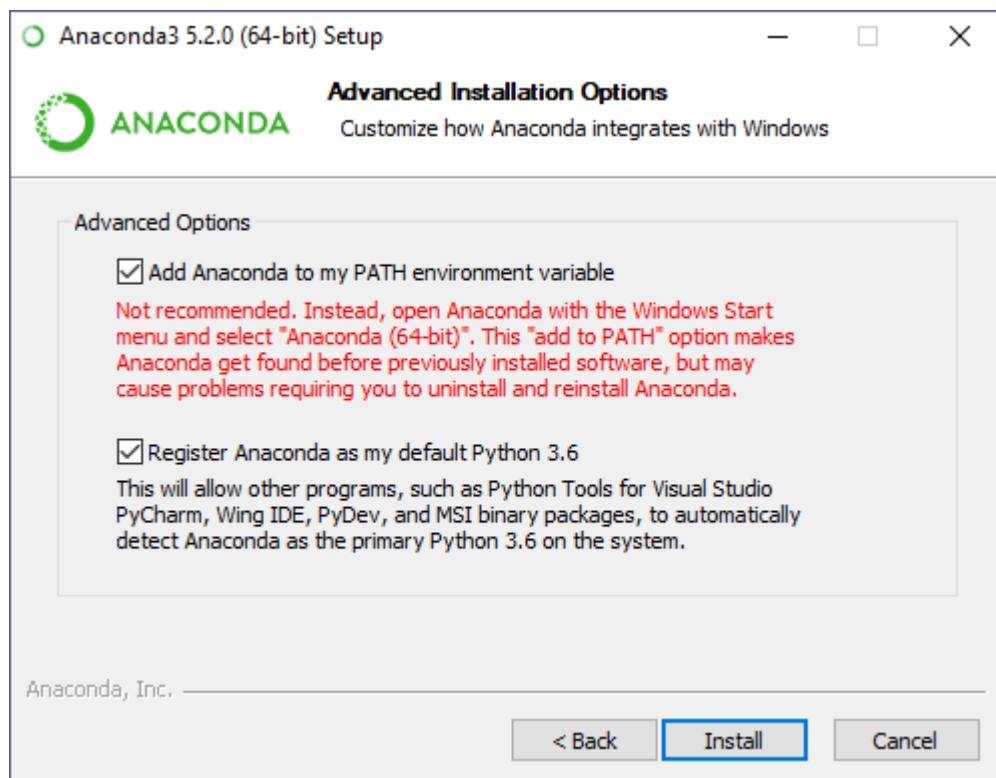
5. Click on Next



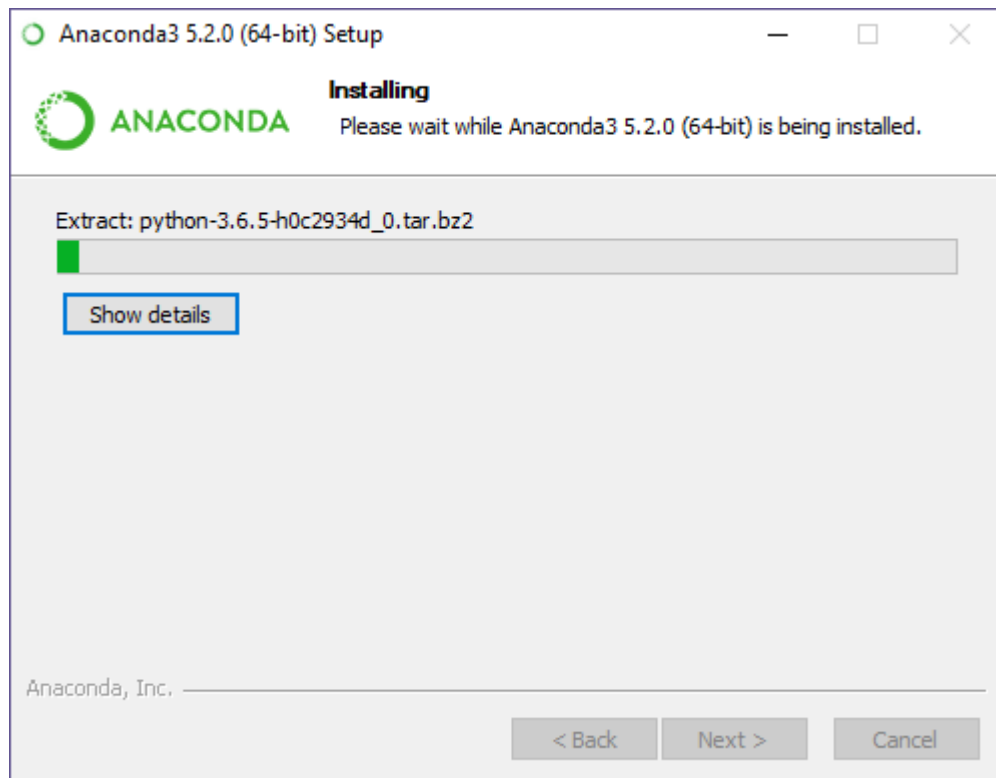
6. Click on Next



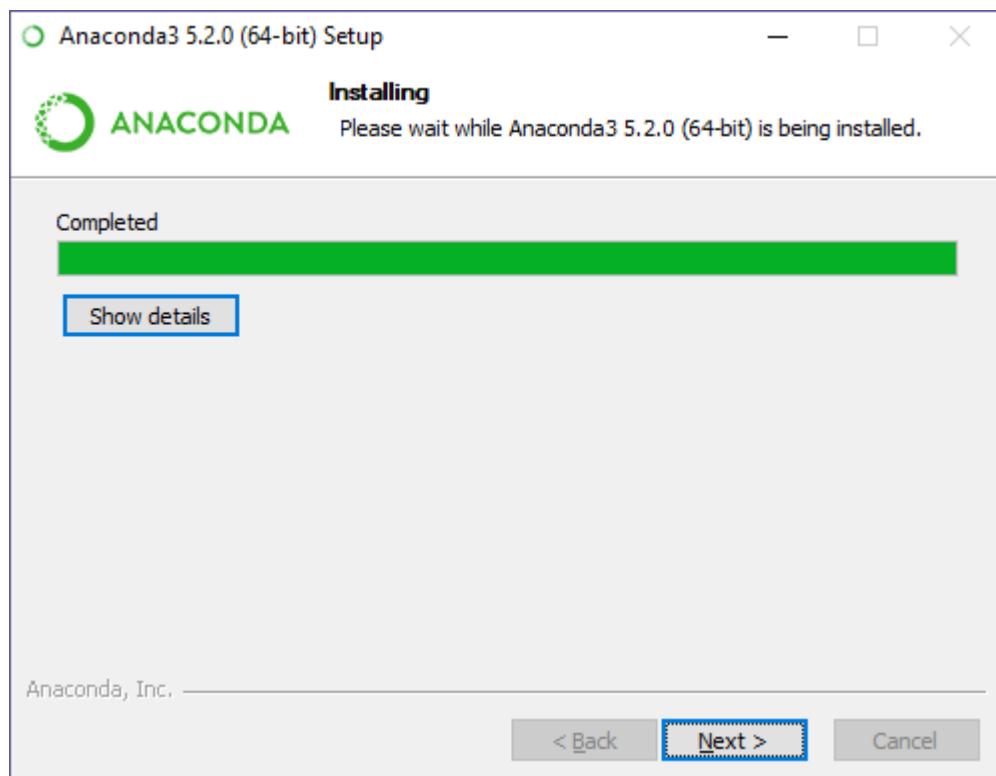
7. Check **both boxes** and Click Install. Ignore the warning.



8. Wait for the installation to complete. This is a large suite of packages and takes some time.



9. Click on Next and complete the installation when it is ready



10. Check out the Continuum documentation (<http://docs.continuum.io/anaconda/install.html>) if there is a problem with installation.
11. If it asks to install Microsoft Visual Studio code it is up to you as to whether you do.

Running from Command Prompt/terminals

We can start using python from command prompts (on Windows) or terminals (on Mac OS). A command prompt (Windows) or terminal (Mac) can be launched in different ways. This section introduces how to start a terminal from a specific folder. We suggest that for each practical, you should save all materials to a folder (e.g., Week1) and start a terminal from this folder.

Windows:

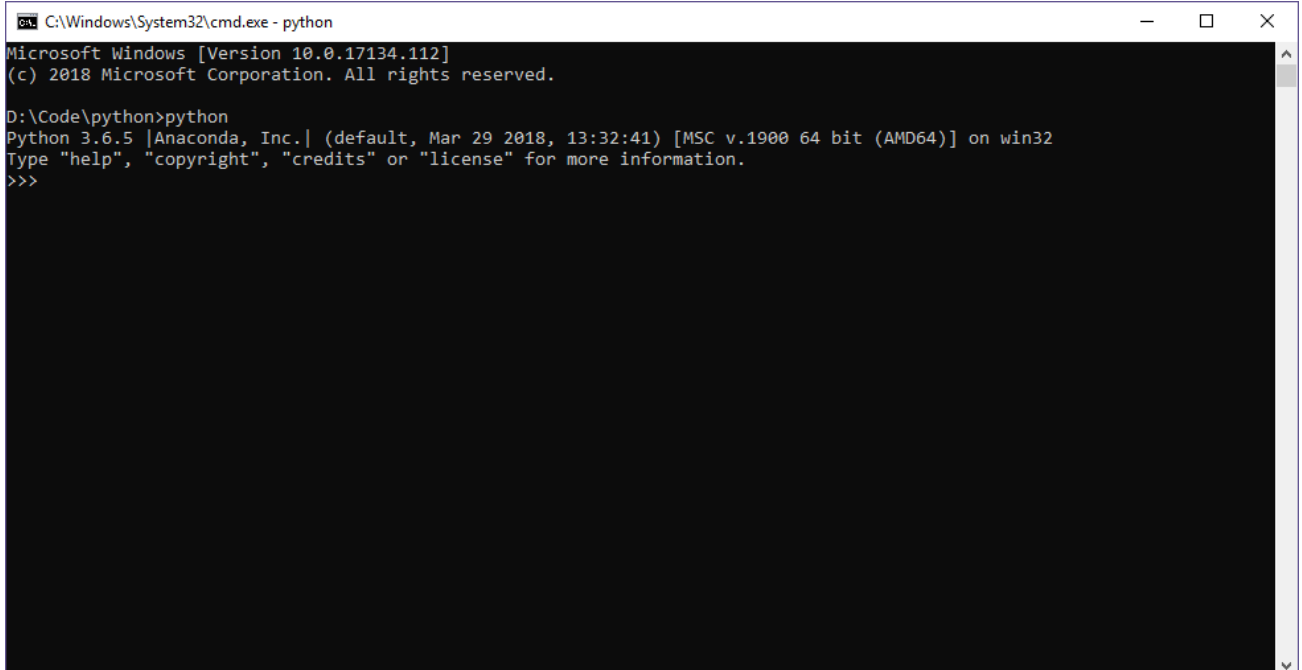
- Go to your folder (e.g., Week1)
- Click on the address bar, type cmd and Enter

Mac OS:

- Head into System Preferences and select Keyboard > Shortcuts > Services.
- Find “New Terminal at Folder” in the settings and click the box.
- Go to the directory of your folder in Finder, right-click on the folder and select “open Terminal”.

Python interpreter

The standard way to use the Python programming language is to use the Python interpreter to run python code. It reads and executes the file that is passed to it. Use the command “python” in a terminal to invoke the Python interpreter. We can also start the interpreter by simply typing python at the command line, and interactively type python code into the interpreter:



```
C:\Windows\System32\cmd.exe - python
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

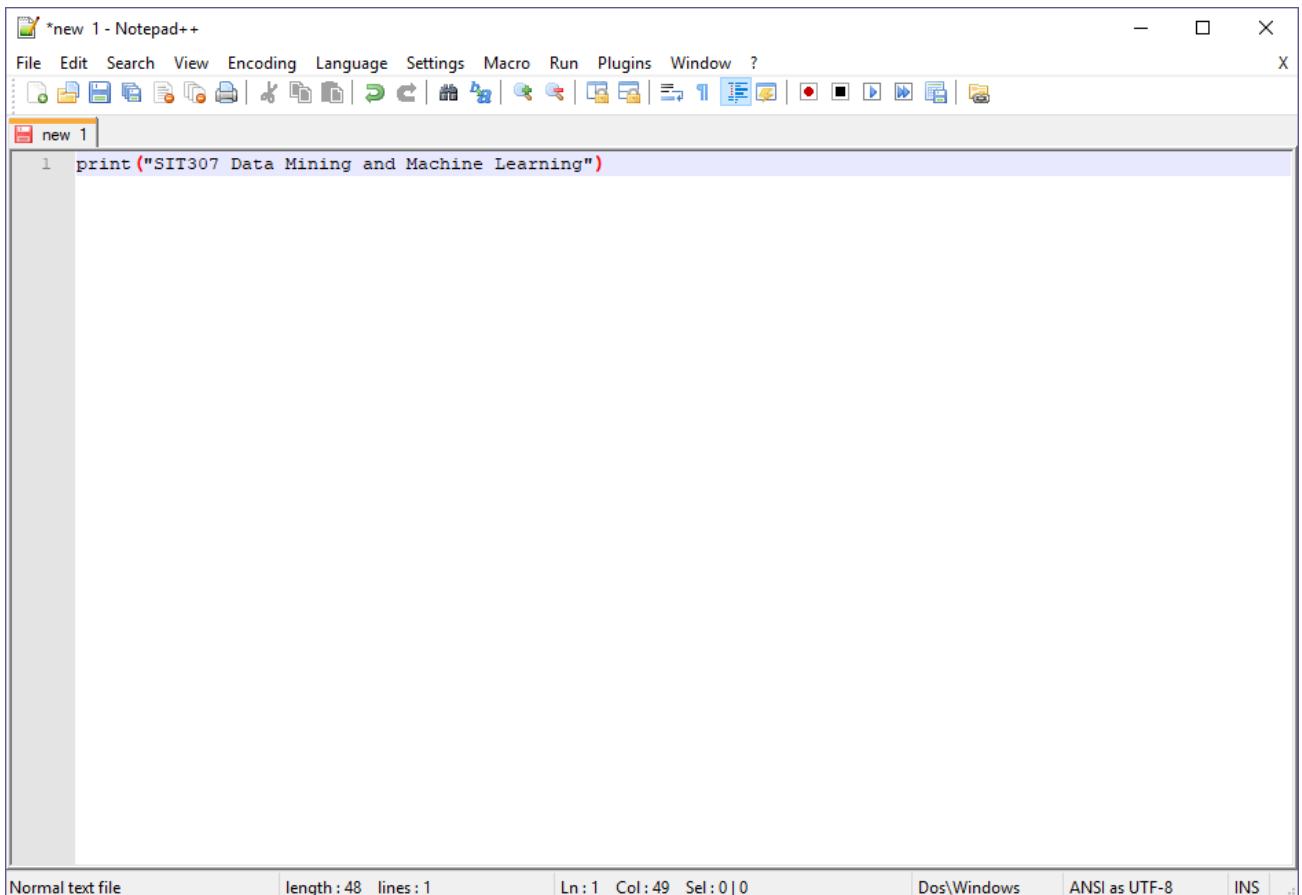
D:\Code\python>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This is a python interpreter and can run code that you enter. At the “>>>” prompt try entering `print(“SIT307 Data Mining and Machine Learning”)` and pressing enter. See the following screen shot:

```
C:\Windows\System32\cmd.exe - python
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\Code\python>python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> print("SIT307 Data Mining and Machine Learning")
SIT307 Data Mining and Machine Learning
>>> _
```

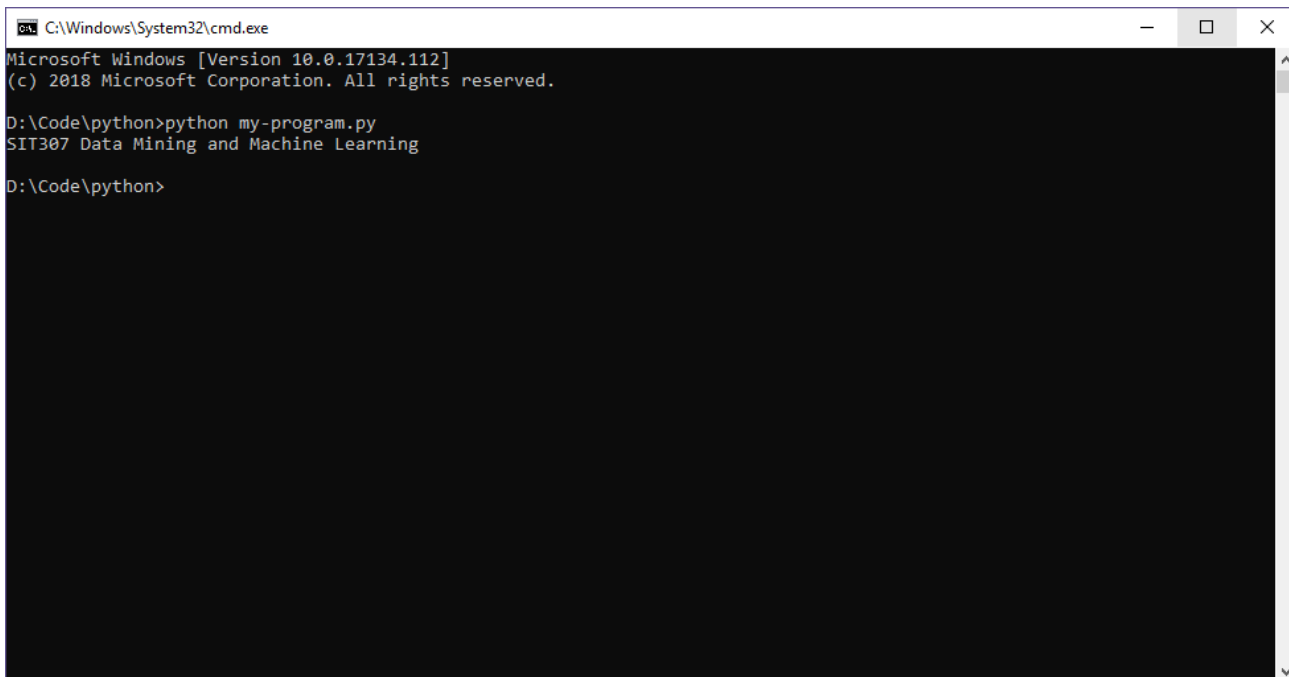
You can also write in a separate file and run it in the python interpreter. Create a file called my-program.py in the folder you launched your cmd window from.



```
*new 1 - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
new 1
1 print("SIT307 Data Mining and Machine Learning")

Normal text file    length : 48    lines : 1    Ln : 1    Col : 49    Sel : 0 | 0    Dos\Windows    ANSI as UTF-8    INS
```

And then try running it in the interpreter. Enter `exit()` to exit the python interpreter and return to the cmd prompt and then enter `'python my-program.py'` to run your program.



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

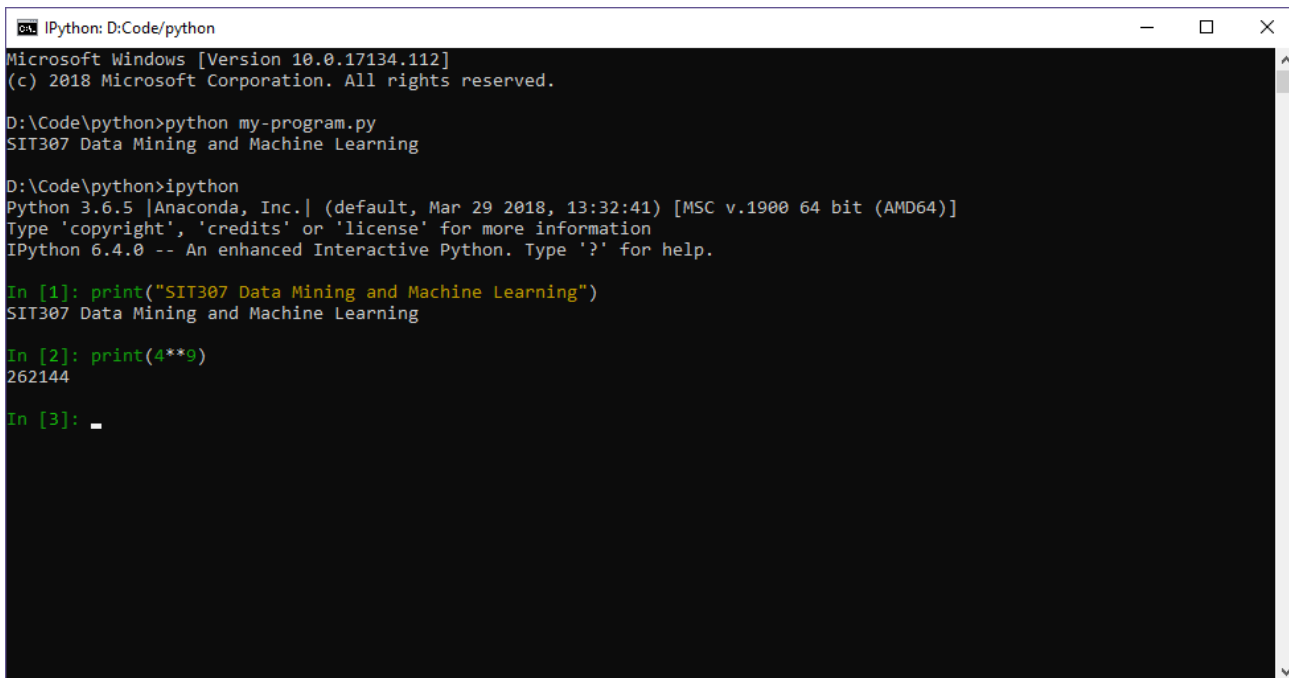
D:\Code\python>python my-program.py
SIT307 Data Mining and Machine Learning

D:\Code\python>
```

Using the interpreter is often how we want to work when developing scientific applications, or when doing small calculations. But the standard python interpreter is not very convenient for this kind of work, due to a number of limitations.

IPython: Interactive Python

IPython is an interactive shell that addresses the limitation of the standard python interpreter, and it is a workhorse for scientific use of python. It provides an interactive prompt to the python interpreter with a greatly improved user-friendliness. You can start IPython by running the command `ipython` in your terminal.



```
IPython: D:\Code\python
Microsoft Windows [Version 10.0.17134.112]
(c) 2018 Microsoft Corporation. All rights reserved.

D:\Code\python>python my-program.py
SIT307 Data Mining and Machine Learning

D:\Code\python>ipython
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print("SIT307 Data Mining and Machine Learning")
SIT307 Data Mining and Machine Learning

In [2]: print(4**9)
262144

In [3]: _
```

Some of the many useful features of IPython includes:

- Command history, which can be browsed with the up and down arrows on the keyboard.

- Tab auto-completion.
- In-line editing of code.
- Object introspection, and automatic extract of documentation strings from python objects like classes and functions.
- Good interaction with operating system shell.
- Support for multiple parallel back-end processes, that can run on computing clusters or cloud services like Amazon EE2.

Read further

- Python (<http://www.python.org>) The official Python web site
- Python tutorials (<http://docs.python.org/2/tutorial>) The official Python tutorials
- Introduction to Python Programming (<https://classroom.udacity.com/courses/ud1110>) A Udacity course
- Think Python (<http://www.greenteapress.com/thinkpython>) A free book on Python
- Google's Python Class (<https://developers.google.com/edu/python/>) Google's Python class
- Code Academy Python (<http://www.codecademy.com/tracks/python>). Learn Python on Code Academy
- Spyder (<http://code.google.com/p/spyderlib/>). A MATLAB-like IDE for scientific computing with python.