Getting access to Python
Mathematical Programming with Python

https://people.sc.fsu.edu/~jburkardt/classes/python_2025/access/access.pdf

Python

Python is one of the most popular modern programming languages.

- *Predominant language for machine learning;*
- *Free;*
- *Windows, MacOS, and Linux;*
- *Use interactively, or as a program, or mixed with other programs*
- *Has similarities to MATLAB;*

1 Access to Python

This class will involve many programming examples, exercises, and homework. The programming language will be Python, version 3.0. Along with Python, we will need associated libraries such as `numpy`, `scipy`, `matplotlib`. There are several ways for you to access Python.

- it might already be installed on your computer;
- Google Colab gives you interactive online access;
- You can install Anaconda on your computer.

2 Already installed?

Your computer may already have a version of Python installed. However, we need to be sure you have Python version 3, and the extra libraries we will be using. Also, you need to know how to access a terminal or command window, in order to start Python.

- Windows: a powershell prompt should appear in your Start menu. Click on it;
- MacOS: In the Finder, open the `/Applications/Utilities` folder, then double-click `Terminal`;
- Linux: right click on the background screen, to get a menu that includes `Open Terminal`.

Within the terminal window, try to run a Python session by typing the command `python`. If you don't have Python, then you will get an error message like *Command not found*. Otherwise, you will start a Python session, which will also identify the version of Python you have. In that case, use the `import` command to request access to libraries. If you get *ModuleNotFoundError* then that particular library is not available. Terminate your Python session with the `quit()` command.

```
python               <-- This starts a Python session
                     <-- You will see a version number, such as Python 3.8.8.
                     <-- Make sure the first digit is 3!
>> import graphviz   <-- Just checking if these libraries are available.
>> import matplotlib
>> import numpy
>> import pandas
>> import PIL
>> import scipy
>> import sympy
>> quit()
```

If your Python version is 3 or higher, and all the libraries showed up, then you are very lucky, and you can run Python on your laptop with no further worries.

3 Google Colab

Google Colab is an online web service that allows you to create Python programs and run them online through your browser. The files you create are saved on Colab for your use when you login again later. You can also transfer Python files between your laptop and Colab.

You can access Colab at:

`https://colab.research.google.com`

You start by opening a notebook. The notebook consists of code cells and text cells. A code cell is a box in which you can type Python commands. A text cell is a box in which you can type comments. Colab has all the libraries we will be using.

Your homework assignments can be created and run on Colab. When you are satisfied with your results, you can submit a copy of your files to the class Canvas directory.

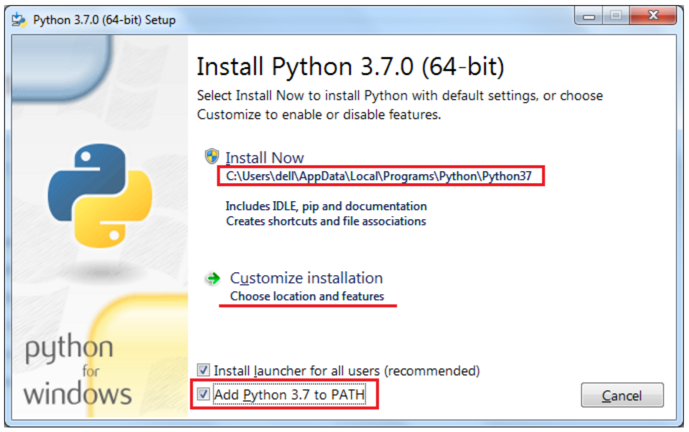
4 Install Anaconda

If you don't have a satisfactory version of Python installed on your laptop, and you don't want to use Colab, you can have Python installed by downloading Anaconda, and its package manager `conda`.

Go to the site

`https://www.anaconda.com/products.individual`

and download a copy of Anaconda for your operating system.



Python 3.7.0 (64-bit) Setup

Install Python 3.7.0 (64-bit)

Select Install Now to install Python with default settings, or choose Customize to enable or disable features.

Install Now
C:\Users\dell\AppData\Local\Programs\Python\Python37

Includes IDLE, pip and documentation
Creates shortcuts and file associations

Customize installation
Choose location and features

☑ Install launcher for all users (recommended)
☑ Add Python 3.7 to PATH

Anaconda Download for Windows

5 Editors, Code Editors, IDE's

To create your Python programs, you will need some kind of program that can create and modify text files. You probably already have a text editor that you use, and that will be enough for this class. However, note that programs like Microsoft Word do not generate plain text files, and so are not suitable for our work.

You may, however, be interested in finding a better way to create your programs. Programmers describe three levels of such tools, starting with simple text editors, then more ambitious code editors, and then rising to the sophisticated integrated development environments. You should look around at the many options, and compare what your friends and classmates use.

The simplest such programs are called *text editors*. These programs create and modify any kind of text file. They don't know that you are writing a Python file, so they won't notice any simple mistakes you make. Editors include Atom, BBedit, emacs, vi, vim.

A *source code editor* will understand that the text file you are working on is actually using a particular programming language. It will usually guess this from the filename extension. Python files have the form *name.py* for instance. The source code editor knows some of the rules of the language. It may use color to highlight keywords. It may automatically indent where appropriate. It may warn you about some simple misspellings, missing commas or parentheses, and other mistakes. Source code editors include Notepad++, Sublime Text.

An *Integrated Development Environment* includes the features of a source code editor, but also opens up a second window in which you can execute the code as you are working on it. It may also offer debugging tools. This makes it easy to create a first draft, run it in the second window, see the error, jump back to the edit window for correction, and continue to switch back and forth between editing and execution in a smooth way. IDE's include IDLE, JupyterLab, PyCharm, Spyder, Visual Studio, VSCode.

6 Look ahead at Python

We will try to teach you Python starting with the basics. Nonetheless, it can be helpful for you to try to get an advance look at the language. If you have no programming experience at all, this may take some time to get used to. If you have experience in programming in C, C++, Fortran, Java, or other procedural languages, then a lot of the features of Python will be familiar. If, better yet, you have worked with an interactive language like MATLAB or R, then Python will seem very similar, and it will just be a question of "translating" how you do things.

One excellent way to start is to refer to the online tutorial available at:

`https://docs.python.org/3/tutorial/`

and work through some of the initial examples.

You don't need to become an expert, but you should start by focusing on:

- how you define and use data;
- how `if` statements make decisions;
- how `for` and `while` carry out repetitive operations on data;
- how you define and use functions to modify data;
- how the `import` command makes a library accessible;
- how lists, vectors and arrays organize multiple data items;