Python Starter Kit

What is Python?

Python is a high-level, interpreted general-purpose programming language. It is object-oriented. Python 2.0 was released in 2000 and Python 3.0 was released in 2008. Due to concern about how much code was written for Python 2, support for Python 2.7 was extended to 2020. Because of this, I recommend starting from or making the switch to **Python 3.x** (currently in version Python 3.7).

Python vs MATLAB?

MATLAB is a popular numerical computing environment and programming language. MATLAB typically refers to the whole package, *include the IDE* (integrated development environment, more on this later). MATLAB has valuable features for scientific computing and is typically easier to use for beginners but has some fundamental problems. First is it's **expensive**. UCSD has a license, but that may not be true for everywhere you go. Furthermore, the algorithms are **proprietary**, so you cannot look at the source code and have to just trust MATLAB. It is also different from other languages, so it makes transitioning a bit more difficult than if you become proficient with Python.

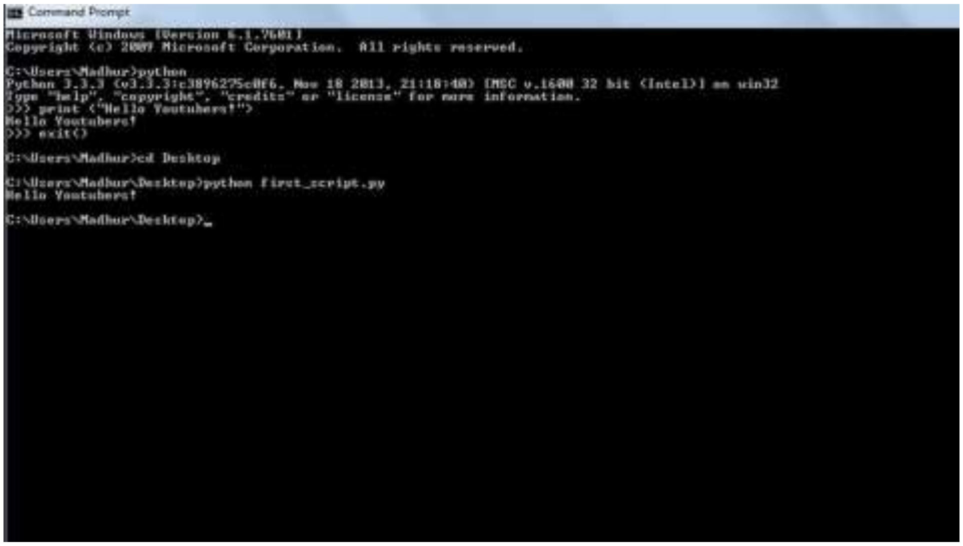
So why is Python great? It's **free**! It's also a beautiful, powerful language that is used in industry and the scientific community alike. While there is a start-up cost to learning it, the ceiling is much higher. It also has better capabilities as an actual language allowing for namespaces, introspection, string manipulation, etc.

What do I need for Python?

First, you'll need an IDE. Also, to get similar scientific computing functionality as MATLAB in Python, you need additional packages. Anaconda can help with both!

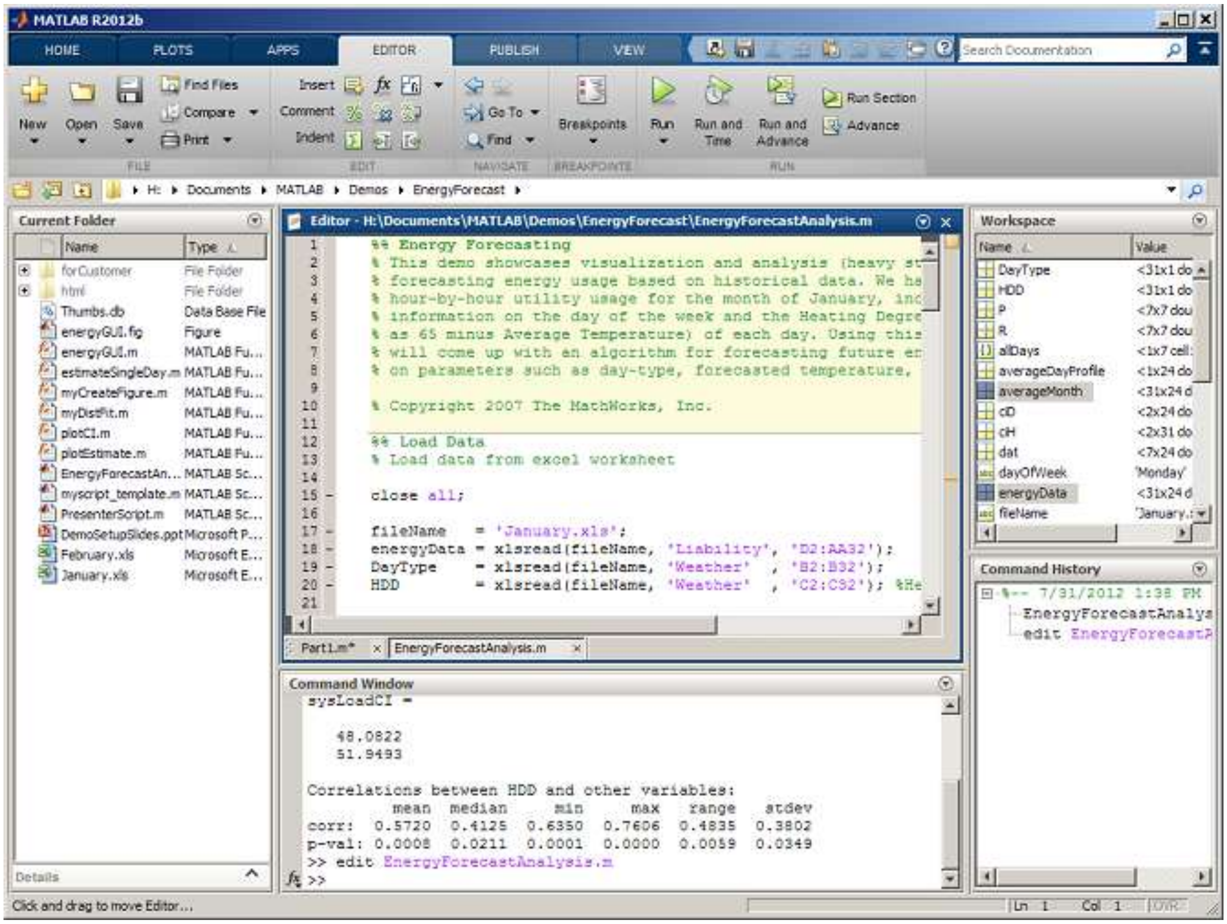
What is an IDE?

IDE stands for integrated development environment and is a software application that provides comprehensive facilities to computer programmers for software development. Basically, it is a user interface making programming easier and maximizing efficiency. It's the difference between this (no IDE):



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

C:\Users\Madhur>python
Python 3.3.3 (v3.3.3:c3896275c0f6, Nov 18 2013, 21:18:40) [MSC v.1600 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> print ("Hello Youtubers!")
Hello Youtubers!
>>> exit()

C:\Users\Madhur>cd Desktop

C:\Users\Madhur\Desktop>python first_script.py
Hello Youtubers!

C:\Users\Madhur\Desktop>_
```

And this (with IDE) MATLAB:



Name	Type
forCustomer	File Folder
html	File Folder
Thumbs.db	Data Base File
energyGUI.fig	Figure
energyGUI.m	MATLAB Fu...
estimateSingleDay.m	MATLAB Fu...
myCreateFigure.m	MATLAB Fu...
myDistFit.m	MATLAB Fu...
plotCI.m	MATLAB Fu...
plotEstimate.m	MATLAB Fu...
EnergyForecastAn...	MATLAB Sc...
myscript_template.m	MATLAB Sc...
PresenterScript.m	MATLAB Sc...
DemoSetupSlides.ppt	Microsoft P...
February.xls	Microsoft E...
January.xls	Microsoft E...

Editor - H:\Documents\MATLAB\Demos\EnergyForecast\EnergyForecastAnalysis.m

```
%% Energy Forecasting
% This demo showcases visualization and analysis (heavy st
% forecasting energy usage based on historical data. We ha
% hour-by-hour utility usage for the month of January, inc
% information on the day of the week and the Heating Degre
% as 65 minus Average Temperature) of each day. Using this
% will come up with an algorithm for forecasting future en
% on parameters such as day-type, forecasted temperature,

% Copyright 2007 The MathWorks, Inc.

%% Load Data
% Load data from excel worksheet

close all;

fileName   = 'January.xls';
energyData = xlsread(fileName, 'Liability', 'D2:AA32');
DayType    = xlsread(fileName, 'Weather'  , 'B2:B32');
HDD        = xlsread(fileName, 'Weather'  , 'C2:C32'); %He

```

Name	Value
DayType	<31x1 do
HDD	<31x1 do
P	<7x7 dou
R	<7x7 dou
allDays	<1x7 cell:
averageDayProfile	<1x24 do
averageMonth	<31x24 d
cD	<2x24 do
cH	<2x31 do
dat	<7x24 do
dayOfWeek	'Monday'
energyData	<31x24 d
fileName	'January.:

```
%-- 7/31/2012 1:38 PM
  EnergyForecastAnalys
  edit EnergyForecastA
```

```
sysLoadCI =

   48.0822
   51.9493

Correlations between HDD and other variables:
          mean   median      min      max    range    stdev
corr:   0.5720   0.4125   0.6350   0.7606   0.4835   0.3802
p-val:  0.0008   0.0211   0.0001   0.0000   0.0059   0.0349
>> edit EnergyForecastAnalysis.m
>>
```

Which IDE do I get?

This is a matter of personal preference, but here I will review some basic options. For scientific computing, it is a good idea to get Anaconda then you can choose your specific IDE (Spyder, Jupyter Notebook, PyCharm, Visual Studios, etc.) afterwards.

Anaconda

Anaconda is a fully-featured scientific distribution and is great for beginners getting into scientific programming. It works on Linux, Windows, and Mac OS X. It comes with a lot of packages specifically for data science and computing.

Anaconda Navigator Cheat Sheet

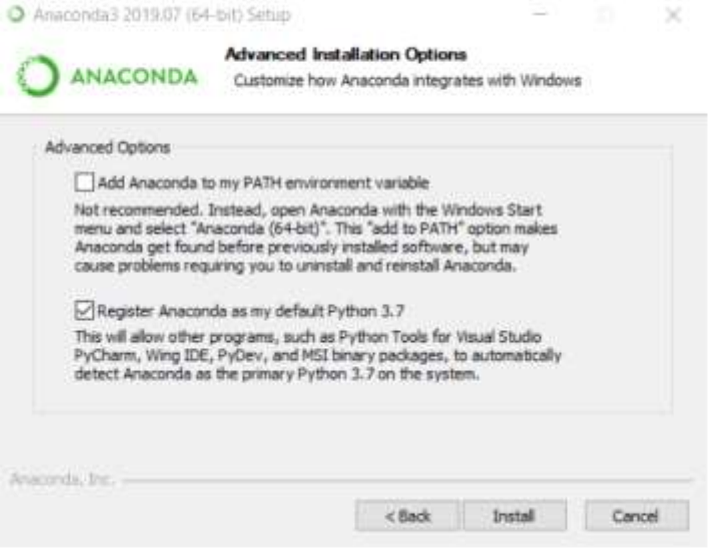
Documentation

Installation:

Download link

Like I touched on before, I recommend downloading Anaconda for **Python 3.7** because Python 2.7 will soon no longer be maintained. Anaconda is free and works with most common operating systems (Windows, Mac OS, Linux). You can either download full Anaconda (5 GB with all packages included) or Miniconda, which is a mini version of Anaconda that does not include all packages if you are worried about disk space. Anaconda has more detailed installation instructions for each type of operating system under the "Documentation".

One option before you download is to "Register Anaconda as my default Python 3.7," so that PyCharm will automatically detect Anaconda (and all of its packages).



Anaconda3 2019.07 (64-bit) Setup

Advanced Installation Options

Customize how Anaconda integrates with Windows

Advanced Options

□ Add Anaconda to my PATH environment variable

Not recommended. Instead, open Anaconda with the Windows Start menu and select "Anaconda (64-bit)". This "add to PATH" option makes Anaconda get found before previously installed software, but may cause problems requring you to uninstall and reinstall Anaconda.

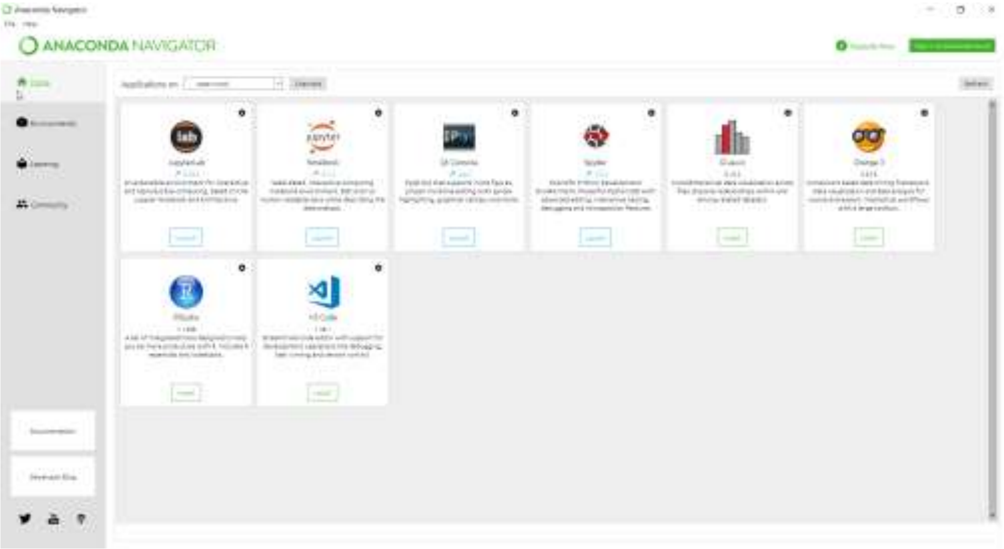
☑ Register Anaconda as my default Python 3.7

This will allow other programs, such as Python Tools for Visual Studio PyCharm, Wing IDE, PyDev, and MSI binary packages, to automatically detect Anaconda as the primary Python 3.7 on the system.

After you install Anaconda, **it will give you the option to also install PyCharm, if you want**. I recommend doing this if you might want to use PyCharm as an IDE.

Getting Started with Anaconda:

When you download Anaconda, it comes with **conda** and **Anaconda Navigator**, as well as Python and hundreds of scientific packages. After you install Anaconda (or Miniconda), you can either use Anaconda Navigator if you prefer a desktop graphical user interface (GUI) or Anaconda prompt/conda if you don't (or your terminal on Linux or macOS). If you aren't sure, try Anaconda Navigator first; you can find it in your program menu. It should look something like this:



Now that you have Anaconda downloaded, you have options with how you want to actually program:

- Spyder
- Jupyter Notebook
- PyCharm
- Other

Below I'll discuss some options briefly, but here are some more details. It also might be worth looking at a few to decide what you prefer most.

Spyder IDE

On the Navigator's Home tab, scroll to Spyder and click "Install" to install Spyder. Afterwards, the Install button will change to a "Launch" button.

Spyder is a very popular IDE and comes with Anaconda already. I haven't used it a ton, but it offers advanced editing, debugging, and data exploration features including things like syntax highlighting and auto code completion features. It isn't quite as powerful as PyCharm but should work cleanly for anything we are doing in this course.

More information

Jupyter Notebook

Jupyter Notebook is a powerful tool for data analysis because it is separated into cells, so you can run each cell separately. It also works really well as a learning tool because you can write text as well as code. For example, you don't have to have a word document and a code file, you can just put everything into a Jupyter Notebook. Note: Jupyter Notebooks use the extension .ipynb instead of .py.

Similar to Spyder, you can install and launch Jupyter Notebook from the Anaconda Navigator.

More information

PyCharm

PyCharm is a Python-specific IDE created by Jet Brains. It is really great for code completion, error detection, quick fixing, debugging, etc. The interface is also customizable. It has a free version which is great for what we need, but also a professional version for other features. The main con is that installation and setup is slightly more difficult, but I really think this is a great tool.

You can download PyCharm specifically for Anaconda or download the general PyCharm and use conda environments to use the packages included with Anaconda.

More information

Other

There are still many other options that you can look into if you want! Visual Studios is great to become familiar with because it is used as the main code editor for other languages although it is not as tailored for Python specifically. There is also Pydev, Idle, Wing, etc. You can even use Python from the command line for shorter programs.

Now what?

Now that you have downloaded Anaconda and discovered your favorite IDE/code editor, you can go through the Python tutorial and Python scientific tutorial on the Neurodynamics website! If you have any other questions, please talk to us. We are here to help you, not intimidate you!

Other reading:

https://en.wikipedia.org/wiki/Python_(programming_language)

https://sites.google.com/site/pythonforscientists/python-vs-matlab

https://en.wikipedia.org/wiki/Integrated_development_environment