

Twitter Thread by Pratham Prasoon



Pratham Prasoon

@PrasoonPratham



Are you planning to learn to Python for machine learning this year?

Here's everything you need to get started.



In this thread we'll look at all the concepts in Python you need to know for machine learning along with free resources to help you out.

All of this is based on my experience of successfully teaching 300+ students how to code using Python.

(2 / 19)

You can use many languages for machine learning, why Python?

Because of 2 reasons:

- Comparatively easier to learn than other languages
- Has the biggest and most mature community

This makes Python a no-brainer to learn for machine learning as a beginner.

(3 / 19)

These are the absolute basics which you must know about:

- Basic terminal commands
- Basic arithmetic (+, -, /, *)
- Accepting user input
- For & While loops
- Exception handling
- If-Else statements
- Functions, modules & Imports

(4 / 19)

Then comes the more tougher concepts which you must know about:

- Object oriented programming in Python:Classes, Objects, Methods
- PIP (Pypi)
- List slicing
- String formatting
- Dictionaries & Tuples
- Managing environments
- Dunder methods like `__init__`

(5 / 19)

This are even more advanced concepts but you do not need then to start machine learning:

- Lambda functions
- Built in libraries like CSV, requests, Sqlite
- Map and Filter
- `*args` and `**kwargs`
- Async
- Decorators

(6 / 19)

From what I've observed, most beginners just find it really difficult just to get the Python environment setup and then using the terminal becomes an even bigger nightmare for them.

Let's tackle this issue.

(7 / 19)

You need to install:

- Anaconda for managing environments (different versions of Python)
- Python3
- Machine learning packages like Sckit learn and TensorFlow using pip when needed

(8 / 19)

Anaconda installation guide for ■

MacOS: ■docs.■anaconda.■com/anaconda/install/mac-os/

Windows: ■docs.■anaconda.■com/anaconda/install/windows/

Linux: ■docs.■anaconda.■com/anaconda/install/linux/

(9 / 19)

Home

Anaconda Individual Edition

Installation

Installing on Windows

Installing on macOS

Installing on Linux

Installing on Linux POWER

Installing in silent mode

Installing for multiple users

Verifying your installation

Anaconda installer file hashes

Updating from older versions

Uninstalling Anaconda



Installing on macOS

Note

Using Anaconda in a commercial setting? You may need to use [Anaconda Commercial Edition](#). If you have purchased Commercial Edition, please proceed to the [Authenticating Commercial Edition](#) section for installation here.

Haven't purchased Commercial Edition yet? Visit <https://anaconda.cloud/register> to get started.

You can install Anaconda using either the graphical installer ("wizard") or the command line ("mar"). If you are unsure, choose the graphical install.

macOS graphical install

1. Download the graphical [macOS installer](#) for your version of Python.
2. RECOMMENDED: [Verify data integrity with SHA-256](#). For more information on hashes, see [Why hash verification?](#)

MacOS and Linux have Python pre-installed, for windows you'll have to install it yourself and it is really easy to mess up the install.

Here's a guide with step by step instructions which will help you.

bit.ly/3rbDoyl

(10 / 19)

ts

Windows
iller
tion

ing the

ig

ig

ng an

ft Store

Issues

g

able

ding

undles

Python

3. Using Python on Windows

This document aims to give an overview of Windows-specific behaviour you should know about when using Python on Microsoft Windows.

Unlike most Unix systems and services, Windows does not include a system supported installation of Python. To make Python available, the CPython team has compiled Windows installers (MSI packages) with every [release](#) for many years. These installers are primarily intended to add a per-user installation of Python, with the core interpreter and library being used by a single user. The installer is also able to install for all users of a single machine, and a separate ZIP file is available for application-local distributions.

As specified in [PEP 11](#), a Python release only supports a Windows platform while Microsoft considers the platform under extended support. This means that Python 3.9 supports Windows 8.1 and newer. If you require Windows 7 support, please install Python 3.8.

There are a number of different installers available for Windows, each with certain benefits and downsides.

[The full installer](#) contains all components and is the best option for developers using Python for any kind of project.

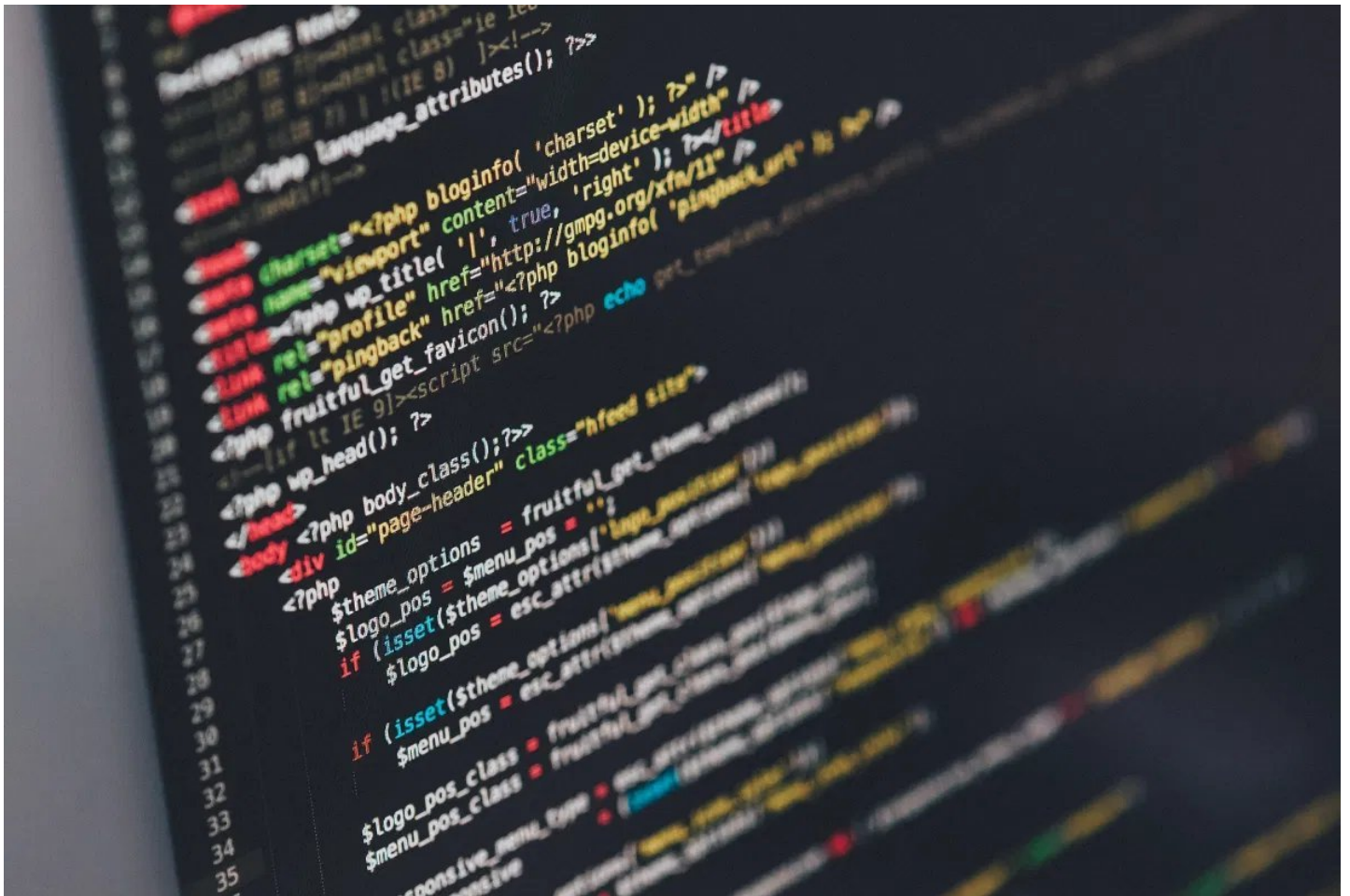
[The Microsoft Store package](#) is a simple installation of Python that is suitable for running scripts and packages, and using IDLE or other development environments. It requires Windows 10, but can be safely installed without corrupting other programs. It also provides many convenient commands for launching Python and its tools.

After you do all of that, you need a place to write your code which is called a "code editor".

Here are some popular ones

- VS code: Feature rich
- Sublime: Light and simple
- Jupyter: Useful for prototyping
- Pycharm: Full blown IDE i.e. has loads of features.

(11 / 19)



If all of that seems complicated to you, I suggest you use Google colab, Kaggle notebooks or repl.it. These are online editors which have everything setup for you.

Not to mention colab and kaggle notebooks give you a free GPU for your machine learning workloads.

(12 / 19)

Links for these editors

Colab : colab.research.google.com

Kaggle Notebooks : kaggle.com/notebooks/welcome

Repl : repl.it

The screenshot shows a Google Colab interface. The notebook title is "Drive FUSE example.ipynb". The left sidebar contains a file explorer with a tree structure:

- drive
 - My Drive
 - Colab Notebooks
 - 07_structured_data.ipynb
 - BigQuery recipes
 - Copy of nima colab.ipynb
 - Exported DataFrame sheet.gsheets
 - Pickle + Drive FUSE example.ipynb
 - Sample Excel file.gsheets
 - Sample Excel file.xlsx
 - TFGan tutorial in Colab.txt
 - created.txt
 - foo.txt
 - script.ipynb
 - to_upload (1).ipynb
 - to_upload (2).ipynb
 - to_upload (3).ipynb
 - to_upload.ipynb
 - variables.pickle
 - sample_data

The main area shows a code cell with the following code:

```
[1] 1 # Load the Drive helper and mount
    2 from google.colab import drive
    3
    4 # This will prompt for authorization.
    5 drive.mount('/content/drive')
```

Below the code cell, a message indicates: "Mounted at /content/drive". Below that, a file explorer view shows the contents of the mounted drive:

07_structured_data.ipynb	Sample Excel file.xlsx
BigQuery recipes	script.ipynb
Colab Notebooks	TFGan tutorial in Colab.txt
Copy of nima colab.ipynb	to_upload (1).ipynb
created.txt	to_upload (2).ipynb
Exported DataFrame sheet.gsheets	to_upload (3).ipynb
foo.txt	to_upload.ipynb
Pickle + Drive FUSE example.ipynb	variables.pickle
Sample Excel file.gsheets	

The Basic & Intermediate Python course on freecodecamp go over pretty much all Python concepts you need for machine learning which I have mentioned above.

Basics: [youtube.com/watch?v=rfscVS0vtbw](https://www.youtube.com/watch?v=rfscVS0vtbw)

Intermediate: [youtube.com/watch?v=HGOBQPFzWko](https://www.youtube.com/watch?v=HGOBQPFzWko)



Python in 4 hours

Full Course

Another thing which most beginners skip is knowing how to use the terminal properly and the know-how of navigating around folders.

Here's a brilliant website which gives you an overview of the windows command prompt, enough for you to get started.

bit.ly/34tmnGd

(15 / 19)

Windows Command Prompt in 15 Minutes

A few Command Prompt commands that you will need to master to make it through COS 126. The following is a brief tutorial covering the most important

What is Command Prompt?

Command Prompt allows you to work in an environment that looks more like a traditional operating system as opposed to the icon based Windows environment. You use your keyboard. You won't use your mouse at all. Command Prompt works at a lower level than Windows. This means that you will have more control. A disadvantage is that it is less user-friendly.

We use Command Prompt in COS 126 to compile and execute your Java programs. Learning the Command Prompt also provides a gradual transition to Unix and Linux, which are prevalent in science, engineering, and industry.

To start Command Prompt select *Start -> Run* and type `cmd` in the box.

Command Prompt shows up as a black terminal window. The *command prompt* should look something like:

As you type commands. The boldface type below (that follows the command prompt) is what you should type as you work through this tutorial. Windows does not use case. That means that command `cd` is the same as `CD`. It also means that, in Windows, file `HelloWorld.java` is the same as `helloworld.java`. This is NOT the case in Unix. Be very careful!!!

Some Useful Commands

To compile a Java program, use the `javac` command. Your program should compile without any errors or warnings (or if there are warnings be absolutely sure you understand the warnings). If there is a warning, it is a flaw in your program).

```
>javac HelloWorld.java
```

The story is a bit different on Linux and Mac, their terminals are extremely powerful and packed to the brim with features, here's a tutorial which will help you get started with the basics ■

■■youtube.■com/watch?v=oxuRxtRO2Ag

(16 / 19)



Keep in mind that you should learn how to use the linux terminal because at some point in your machine learning journey you will have to deal with linux.

It is not important to learn it at the start but I do recommend it.

(17 / 19)

This tutorial will help you in knowing how to work with folders, this is important!

Windows: ■youtube.■com/watch?v=HDmwiJxzlrw

Mac: ■youtube.■com/watch?v=3TAEC-1YUZw

Linux: ■youtube.■com/watch?v=HbgzrKJvDRw

(18 / 19)

