Introduction to Python

Modern conception of Data Science is the art and science of solving real world problems and for making data driven decisions. To begin with, it involves an amalgamation of three aspects. These aspects are:

1. Basic proficiency in understanding the data
2. Basic idea of the tools & technology (Python)
3. Domain knowledge.

With the basic understanding of those three aspects, you can begin the journey of understanding Data Science. With consistent effort, you can become fairly proficient in all three aspects over a period of time. The document is intended to help you become comfortable with the finer nuances of python (as a platform for working on data) and can be used as a handy reference for anything related to data science codes throughout the learning journey and beyond that.

Please keep in mind there is no one right way to write a code to achieve an intended result or an outcome. There can be multiple ways of doing things in Python. The examples presented in this document are just one of them. Hence there is ample scope for exploring on the same.

Python: [https://www.python.org]

Python is an open-source, object-oriented, high-level programming language with dynamic semantic and has high-level built in data structures. Python's simple and easy to learn syntax. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy and when the interpreter discovers an error, it raises an exception.
[Further Reading: https://www.python.org/doc/essays/comparisons/]

Now, you must be wondering what could be the difference between a Python Library, Module and Package. Let's try to understand them individually.

Module: It is a simple Python file that contains collections of functions and global variables and with having a .py extension file. It is an executable file and to organize all the modules we have the concept called Package in Python. The **Python** standard library contains well over 200 **modules**.

Package: The package is a simple directory having collections of modules. This directory contains Python modules and also having_init_.py file by which the interpreter interprets it as a Package. The package is simply a namespace. The package also contains sub-packages inside it.

Library: A collection of related functionality of codes that allows you to perform many tasks without writing your code. It is a reusable chunk of code that we can use by importing it in our program, we can just use it by importing that library and calling the method of that library with period(.). There are over 137,000 python libraries present today.

It's a bit confusing isn't it? Let's make another attempt.
We know that a module is a file with some Python code, and a package is a directory for sub packages and modules. But the line between a package and a Python library is quite blurred.
A Python library is a reusable chunk of code that you may want to include in your programs/ projects. Compared to languages like C++ or C, a Python library does not pertain to any specific context in Python. Here, a 'library' loosely describes a collection of core modules.
Essentially, then, a library is a collection of modules. A package is a library that can be installed using a package manager.

Python Standard Library

The Python Standard Library is a collection of exact syntax, token, and semantics of Python. It comes bundled with core Python distribution. We mentioned this when we began with an introduction.

It is written in C, and handles functionality like I/O and other core modules. All this functionality together makes Python the language it is.

More than 200 core modules sit at the heart of the standard library. This library ships with Python.

But in addition to this library, you can also access a growing collection of several thousand components from the Python Package Index (PyPI).

[Further Reading: https://docs.python.org/3/library/]

Most popular Libraries for Data Science are:

SciPy: SciPy (pronounced "Sigh Pie") is a Python-based ecosystem of open-source software for mathematics, science, and engineering.

Matplotlib: A Python 2D plotting library which produces publication quality figures (graphs & charts) in a variety of formats and interactive environments across platforms.

Note: **matplotlib**. **pyplot** is a collection of functions that make **matplotlib** work like MATLAB.

Each **pyplot** function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc

Seaborn: It is a data visualization library built on top of matplotlib and closely integrated with pandas data structures in Python. Visualization is the central part of Seaborn which helps in exploration and understanding of data.

Note: **sns**.**set()** is used to set the seaborn theme. You can either customize seaborn theme or use one of six variations of the default theme. Which are called deep, muted, pastel, bright, dark, and colorblind.

Pandas: A fast, powerful, flexible and easy-to-use open source data analysis and manipulation tool.

Numpy: A library adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

Example1:

Importing Libraries/Modules

Use 'import' and aliasing statement 'as'

import pandas **as** pd

Here, we are importing pandas module with an alias 'pd'.

Example2:

Importing pandas library and call read_csv method using alias of pandas i.e.

pd. import pandas as pd

df = pd.read_csv("file_name.csv")

Note: The **OS module in Python** provides functions for creating and removing a directory (folder), fetching its contents, changing and identifying the current directory, etc. You first need to import the os module to interact with the underlying operating system.

DataFrame is a 2-dimensional labelled data structure with columns of potentially different types. You can think of it like a spreadsheet or SQL table. It is generally the most commonly used **pandas** object.