Python Basics – Data Structures

Contents

Contents

- Installing python
- Python Environment
- Assignments & Naming Convention
- Data Structures
- Important Packages

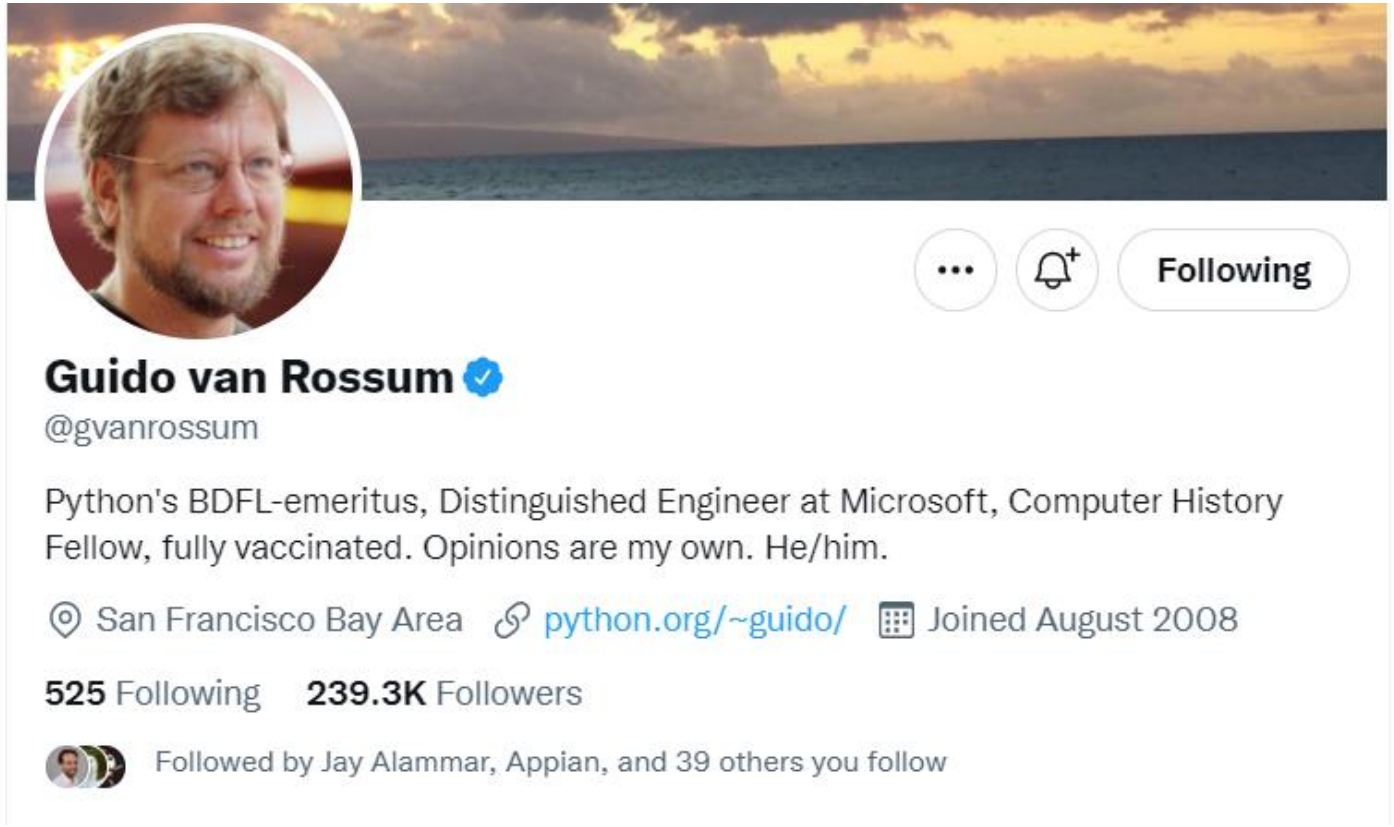
Introduction to Python & History

What is python

- It’s a multi purpose programming language
- Open Source (Free)
- Powerful scripting language with simple Syntax
- Used by many data scientists and developers
- Human-readable syntax and well Documented
- www.python.org

History

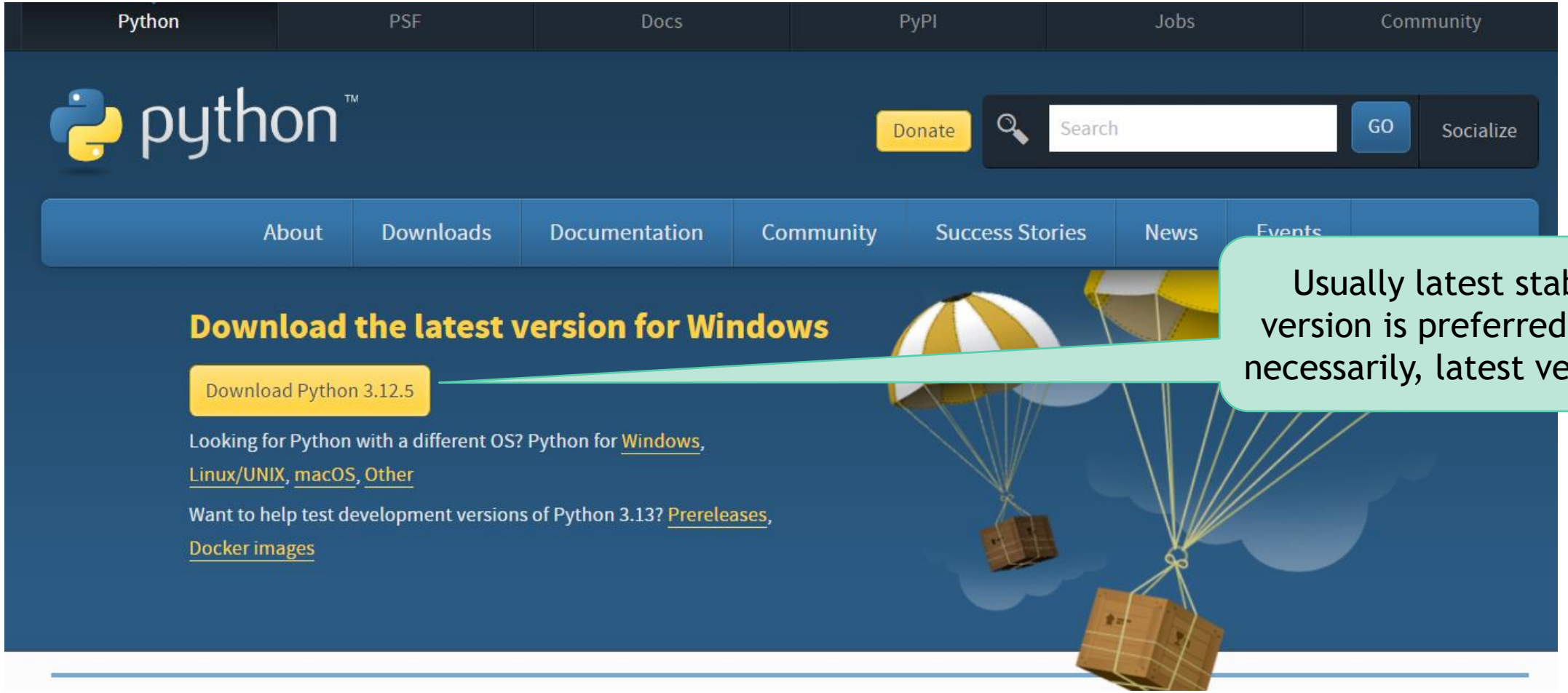
- Python language was developed by Guido van Rossum (Benevolent Dictator for Life).



Python History

- First Python version released in 1991
- Python 2 was released in 2000
- Python 3 was released in 2008
- Python 3.4 (2014)
- Python 3.6 (2016)
- Python 3.9 (2020)
- Python 3.11 (2022)
- Python 3.12 (2023)

Which Version to install?



Usually latest stable version is preferred, not necessarily, latest version.

Installing Python & Python IDEs

Writing and executing python programs

- Python has many options to write and execute a program
- You can use Text Editors or Command line interfaces or Notebook or an IDE
- Anaconda distribution has all the required software's inbuilt. We just need to download and install it.

Installing Python, Anaconda

- Download and install Anaconda3
- It automatically installs
  - Ipython
  - Jupyter notebook
  - Spyder IDE

Most Widely used IDEs

- Data Science Projects
  - Python Notebooks -Jupyter notebook/ Colab notebook
  - Spyder
- Software Development Projects
  - Visual Studio Code (VS Code)
  - PyCharm

Jypyter Notebook

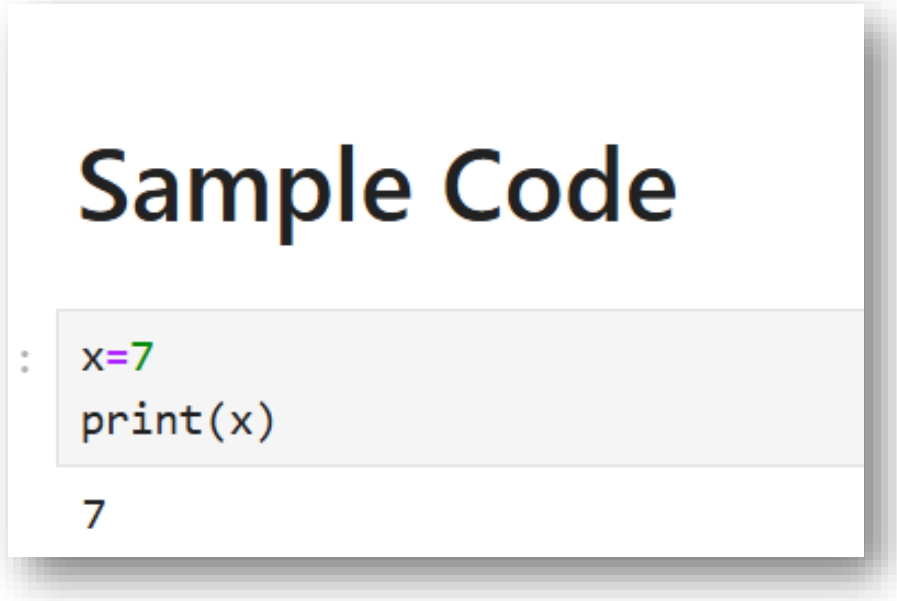
- Anaconda Navigator >>> Select Documents Folder >>> New Folder (name it python_notebooks) >>>New Notebook >>>Select Python3 Kernel >>Rename notebook(Introduction)

Colab Notebook

- Search “Colab notebook” in google

Working with notebooks

- Add a text cell
- Add a code cell



Check Python Version

```
import sys
print("Python version")
print(sys.version)
```

```
import sys
print("Python version")
print(sys.version)
```

```
Python version
3.10.12 (main, Jul 29 2024, 16:56:48) [GCC 11.4.0]
```

Anything above 3.10 is good

Before you code

- Python is case sensitive
- Be careful while using the Variable names and Function names
  - `Sales_data` is not same as `sales_data`
  - `Print()` is not same as `print()`

Basic Commands in Python

Basic Commands

```
print(180+180)
```

```
360
```

```
print("Venkat")
```

```
Venkat
```

```
print(30*12)
```

```
360
```

```
#Division Exampleis given below
print(365/12)
```

```
30.416666666666668
```

What an error looks like?

```
Print(300+900)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-24-ec3057a7da45> in <cell line: 1>()
----> 1 Print(300+900)

NameError: name 'Print' is not defined
```

Typically the error messages are lengthy, look at the end of the message

What an error looks like?

```
67-'37'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-26-8c38e530be0e> in <cell line: 1>()
----> 1 67-'37'

TypeError: unsupported operand type(s) for -: 'int' and 'str'
```

Typically the error messages are lengthy, look at the end of the message

Assigning and Naming convention

Assignment operator

```
income = 12000
print(income)
```

```
12000
```

```
x=20
print(x)
```

```
20
```

```
x="march"
print(x)
```

```
march
```

```
del(x)
```

```
print(x)
```

```
---------------------------------
NameError
<ipython-input-31-fc17d851ef81> in
----> 1 print(x)

NameError: name 'x' is not defined
```

del(x) deletes the x

Assignment operator

```
x=20
```

```
y=30
```

```
z=x*y
```

```
print(z)
```

```
600
```

```
k=50
```

```
z=x*k
```

```
print(z)
```

```
1000
```

Naming convention

- Must start with a letter (A-Z or a-z)
- Can contain letters, digits (0-9), and/or underscore “_”

```
1x =20
```

```
x1=20
```

```
x.1=30
```

```
x_1=20
```

```
x 1=20
```

```
x$20
```

Packages

Packages

- A package is collection of python functions. A properly structured and complied code. A package may contain many sub packages.
- Many python functions are only available via “packages” that must be imported.
- For example to find value of log(10) we need to first import match package that has the log function in it

```
sqrt(1729)
```

```
---------------------------------------
NameError
<ipython-input-44-cee7fedb28fd> in <ce
----> 1 sqrt(1729)

NameError: name 'sqrt' is not defined
```

xt steps: Explain error

```
import math as mt
mt.sqrt(1729)
```

```
41.58124577258358
```

Packages

- To be a good data scientist on python, on need to be very comfortable with below packages
  - numpy
  - scipy
  - pandas
  - scikit-Learn
  - matplotlib
  - nltk

Important Packages- NumPy

- NumPy is for fast operations on vectors and matrices, including mathematical, logical, shape manipulation, sorting, selecting.
- It is the foundation on which all higher level tools for scientific Python packages are built

```
import numpy as np

income = np.array([9000, 8500, 9800, 12000, 7900, 6700, 10000])
print(income)
print(income[0])

expenses=income*0.65
print(expenses)

savings=income-expenses
print(savings)
```

We will have a dedicated session on Numpy

```
import numpy as np

income = np.array([9000, 8500, 9800, 12000, 7900, 6700, 10000])
```

```
print(income)
```

```
[ 9000  8500  9800 12000  7900  6700 10000]
```

```
print(income[0])
```

```
9000
```

```
expenses=income*0.6525
print(expenses)
```

```
[5872.5  5546.25 6394.5  7830.   5154.75 4371.75 6525.  ]
```

```
savings=income-expenses
print(savings)
```

```
[3127.5  2953.75 3405.5  4170.   2745.25 2328.25 3475.  ]
```

Important Packages- Pandas

- Data frames and data handling
- Pandas has Data structures and operations for manipulating numerical tables and time series.

```
import pandas as pd
bank= pd.read_csv('C:\\Users\\venk\\Google
Drive\\Training\\Datasets\\Bank Tele
Marketing\\bank_market.csv')

print(bank)
```

We will have a dedicated session on Pandas

Important Packages- Matplotlib

```
import matplotlib as mp
mp.pyplot.scatter(bank["age"],bank["balance"])
```

We will have a dedicated session on Matplotlib

Important Packages- Scikit-Learn

- Machine learning algorithms made easy

```
import sklearn as sk
import pandas as pd

air = pd.read_csv("D:\\Google
Drive\\Training\\Datasets\\AirPassengers\\AirPassengers.csv")
air

from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(air[["Promotion_Budget"]], air[["Passengers"]])

#Coefficients
print(lr.coef_)
print(lr.intercept_)
```

We will be using this in ML sessions

Get help on Python

- Chat-GPT
- Colab code generation
- Stack Overflow
- Official Documentation

Conclusion

Conclusion

- In this session we got basic introduction to Python.
- We tried some basic commands in Python