First steps in

Me

Algeciras

2011
Aerospace
Engineering

Internship
Image Processing

2013
M. Sc. in
Electronics, signal
Treatment and
Communications

Internship
Mechanical
testing

2018
PhD in Structural
Health Monitoring

R&D Engineer in
Material and
Processes

Reverse Engineering
Consultant

2018
Telemetry systems
specialist

2020
Data
Scientist

What about you?

Name, e-mail and brief description of background (Coding?).

Course roadmap

1) **Theory and system configuration (1d)**
What
Why
How

2) Practical sessions with Python (3d)
Basic concepts in Python
Common 3rd party packages
Advanced topics in Python

3) Python Project (1d)

Objective

Overview of Python

Increase students' interest in the language

Increase general programming knowledge

Course roadmap

1) **Theory and system configuration (1d)**

What

Why

How

2) Practical sessions with Python (3d)

Basic concepts in Python

Common 3rd party packages

Advanced topics in Python

3) Python Project (1d)

What is Python?

What is Python? Executive Summary*

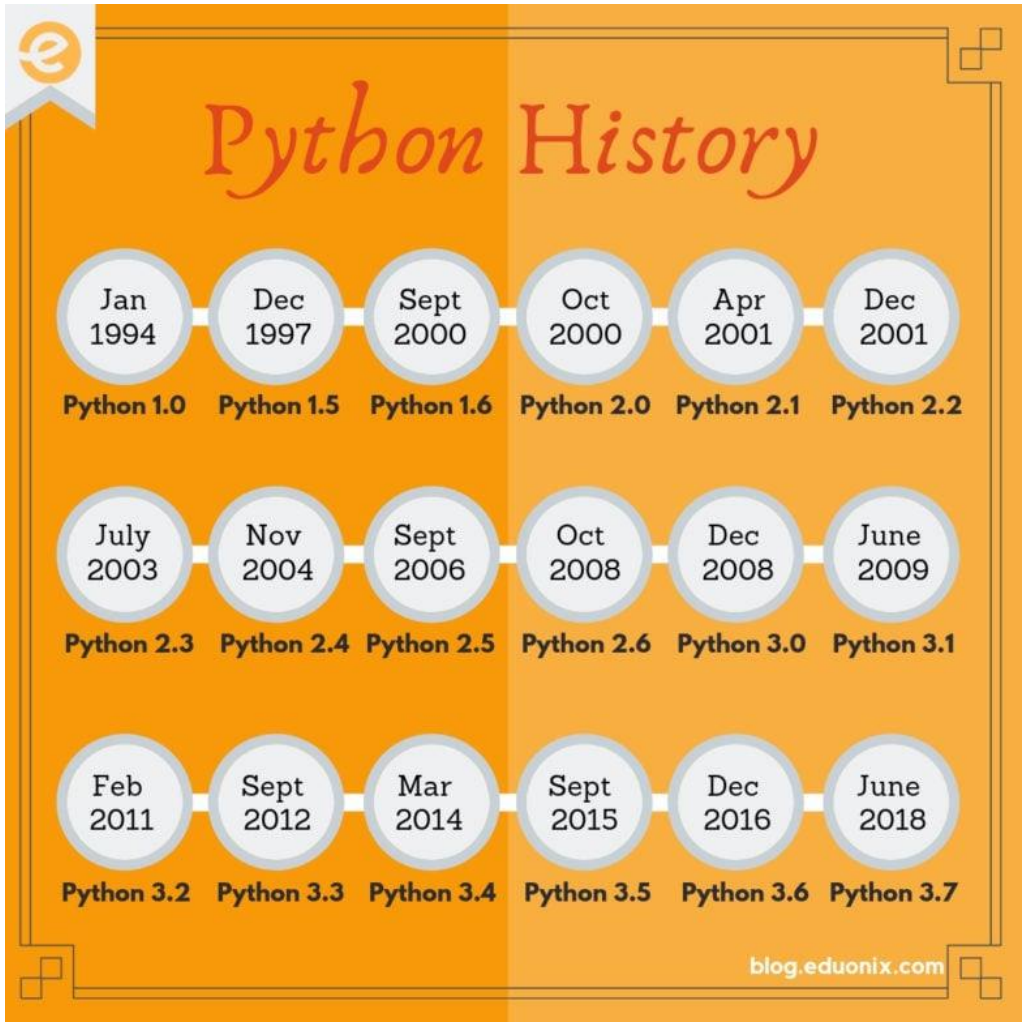
What is Python? Executive Summary

Python is an **interpreted, object-oriented, high-level programming language with dynamic semantics**. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's **simple, easy to learn syntax** emphasizes readability and therefore **reduces the cost of program maintenance**. 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.**

Developer/creator: Guido van Rossum

*https://www.python.org/doc/essays/blurb/

What is Python?



Oct 2019
Python 3.8

Oct 2020
Python 3.9

Oct 2021
Python 3.10

Oct 2022
Python 3.11

*https://blog.eduonix.com/software-development/python-popular-among-programmers/

Course roadmap

1) **Theory and system configuration (1d)**

What

Why

Why

How

2) Practical sessions with Python (3d)

Basic concepts in Python

Common 3rd party packages

Advanced topics in Python

3) Python Project (1d)

Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

Why Python?

High level language

Easy to read, learn and write

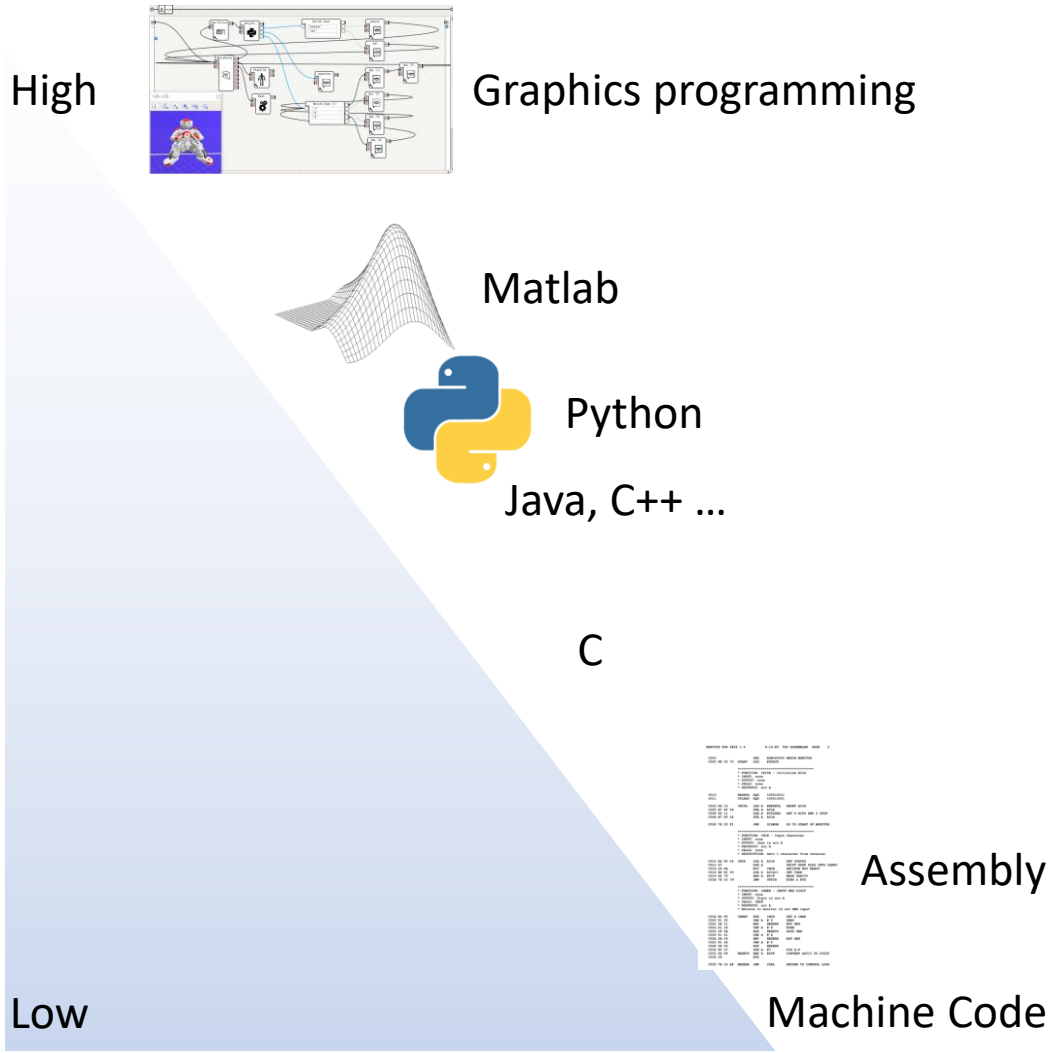
Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science



Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

Clean syntax and indentation structure

```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n’)
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```

Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

*Study by SlashData

The size of programming language communities in Q3 2020 is as follows:

1. JavaScript
Active developers: 12.4 million
Most popular in: Web, Cloud
Least popular in: Data Science, Machine Learning, AR/VR

2. Python
Active developers: 9 million
Most popular in: Data Science, Machine Learning, IoT
Least popular in: Mobile, Web

3. Java
Active developers: 8.2 million
Most popular in: Mobile, Cloud
Least popular in: Data Science, Machine Learning, Web

4. C/C++
Active developers: 6.3 million
Most popular in: IoT, AR/VR
Least popular in: Web, Cloud, Mobile

5. PHP
Active developers: 6.1 million
Most popular in: Web, Cloud
Least popular in: Data Science, Machine Learning, Mobile

6. C#
Active developers: 6.0 million
Most popular in: Games, AR/VR, Desktop
Least popular in: Data Science, Machine Learning, Mobile

*https://content.techgig.com/10-programming-languages-with-the-biggest-community-support/articleshow/78908491.cms

Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

TIOBE Index

May 2022	May 2021	Change	Programming Language	Ratings	Change
1	2	^	Python	12.74%	+0.86%
2	1	v	C	11.59%	-1.80%
3	3		Java	10.99%	-0.74%
4	4		C++	8.83%	+1.01%
5	5		C#	6.39%	+1.98%
6	6		Visual Basic	5.86%	+1.85%
7	7		JavaScript	2.12%	-0.33%
8	8		Assembly language	1.92%	-0.51%
9	10	^	SQL	1.87%	+0.16%
10	9	v	PHP	1.52%	-0.34%
11	17	^^	Delphi/Object Pascal	1.42%	+0.22%
12	18	^^	Swift	1.23%	+0.08%
13	13		R	1.22%	-0.16%
14	16	^	Go	1.11%	-0.11%
15	12	v	Classic Visual Basic	1.03%	-0.38%
16	21	^^	Objective-C	1.03%	+0.24%
17	19	^	Perl	0.99%	-0.05%
18	37	^^	Lua	0.98%	+0.64%
19	11	vv	Ruby	0.86%	-0.64%
20	15	vv	MATLAB	0.82%	-0.41%

PYPL Index

Worldwide, Jan 2023 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	27.93 %	-0.9 %
2		Java	16.78 %	-1.3 %
3		JavaScript	9.63 %	+0.5 %
4	↑	C#	6.99 %	-0.3 %
5	↓	C/C++	6.9 %	-0.5 %
6		PHP	5.29 %	-0.8 %
7		R	4.03 %	-0.2 %
8	↑↑↑	TypeScript	2.79 %	+1.0 %
9		Swift	2.23 %	+0.3 %
10	↓↓	Objective-C	2.2 %	-0.1 %
11	↑↑	Go	1.94 %	+0.7 %
12	↑↑↑	Rust	1.9 %	+0.9 %
13	↓	Kotlin	1.81 %	+0.1 %
14	↓↓↓↓	Matlab	1.63 %	-0.1 %
15	↑	Ruby	1.13 %	+0.3 %
16	↓↓	VBA	1.03 %	-0.0 %
17		Ada	0.89 %	+0.2 %
18	↑↑↑	Dart	0.86 %	+0.5 %

Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

Interpreted

```
# Python 3: Simple output (with Unicode)
>>> print("Hello, I'm Python!")
Hello, I'm Python!

# Input, assignment
>>> name = input('What is your name?\n’)
>>> print('Hi, %s.' % name)
What is your name?
Python
Hi, Python.
```

VS

Compiled

```
public void InitAnimations()
{
    viewportSizeY = GetViewportRect().Size.y;
    spawnY = viewportSizeY / 2.0f + 16f;

    // Player Anim
    Animation anim = GetNode<AnimationPlayer>("FlyInAnim").GetAnimation("FlyIn");
    anim.TrackSetKeyValue(0, 0, new Vector2(0, -viewportSizeY / 2.0f + -80f));
    anim.TrackSetKeyValue(0, 1, new Vector2(0, (-viewportSizeY / 2.0f) + viewportSizeY / 3.0f));
    player.Position = new Vector2(0, -viewportSizeY / 2.0f + -80f);

    anim = GetNode<AnimationPlayer>("BgChangeAnim").GetAnimation("BgChangeAnim");
    anim.TrackSetKeyValue(1, 0, new Vector2(0, 0));
    anim.TrackSetKeyValue(1, 1, new Vector2(0, -viewportSizeY / 2.0f - 200.0f));


    GetNode<Sprite>("SpaceBGOverlay").Position = new Vector2(0, 0);


    anim = GetNode<AnimationPlayer>("BackgroundAnim").GetAnimation("BackgroundAnim");
    GetNode<AnimatedSprite>("BackgroundSprite").Frame = random.RandiRange(0, 1);
    anim.TrackSetKeyValue(0, 0, new Vector2(0, -viewportSizeY / 2.0f + 225.0f));
    anim.TrackSetKeyValue(0, 1, new Vector2(0, +viewportSizeY / 2.0f - 225.0f));
    GetNode<AnimatedSprite>("BackgroundSprite").Position = new Vector2(0, -viewportSizeY / 2.0f + 225.0f);

    GetNode<Sprite>("PrototypLogo").Position = new Vector2(0, 32f);// new Vector2(0,viewportSizeY/2.0f-(viewportSizeY/3.0f));
}
```

*https://www.python.org/ and https://i.stack.imgur.com/EPgUV.png

Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

No royalties or licenses

Why Python?

High level language

Easy to read, learn and write

Great community support

Famous and gaining more popularity

Interpreted

Free and open source

Great for Data Science

NumPy

SciPy

pandas

matplotlib

TensorFlow

...

*https://www.python.org/ and https://i.stack.imgur.com/EPgUV.png

Coding for Data Science

Top 6 Data Science Programming Languages 2021 [Hand-Picked]

edX Blog

9 Top Programming Languages for Data Science

FLATIRON SCHOOL

The 10 Best Data Science Programming Languages to Learn in 2021

BIG DATA MADE SIMPLE

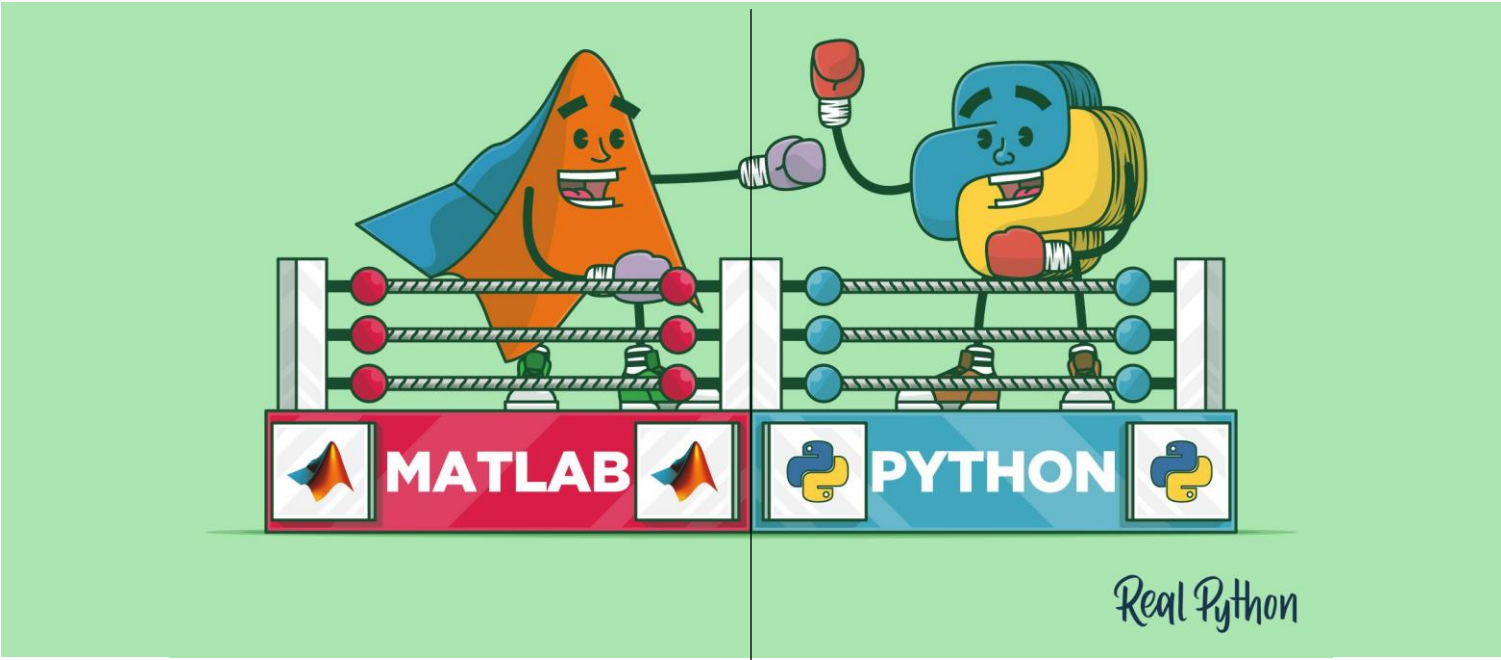
DATA SCIENCE

Top 8 programming languages every data scientist should master in 2019

1. Python	1. Python	1. Python	1. Python
2. JavaScript	2. R	2. JavaScript	2. R
3. Scala	3. SQL	3. Java	3. Java
4. R	4. Scala	4. R	4. SQL
5. SQL	5. Julia	5. C/C++	5. Julia
6. Julia	6. JavaScript	6. SQL	6. Scala
	7. Java	7. Matlab	7. Matlab
	8. C/C++	8. Scala	8. (Tensorflow)
	9. Matlab	9. Julia	
		10. SAS	

Python vs Matlab

Interpreted
High level and easy to use

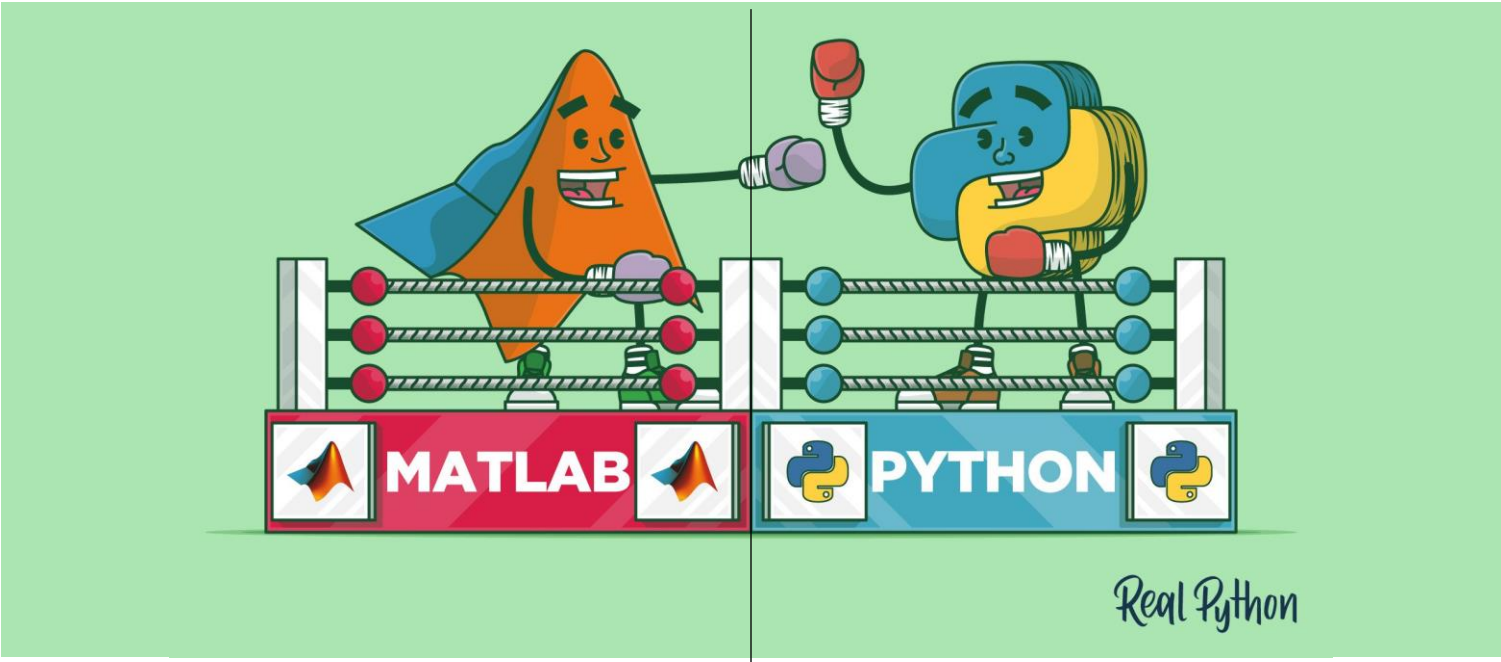


Slightly higher performing (out of the box)	**Free**
Proprietary (Pricey!)	**Open Source**
Closed source	**Good deployment options**
Poor deployment options	Integration with a huge amount of packages
Integration with Simulink	**King of Data Science**
King of simulation	Multiple IDEs and packages
Single IDE, Toolboxes agreed with MathWorks	Huge community support (and growing!)
Amazing help developed by MathWorks	

*https://realpython.com/matlab-vs-python/#:~:text=The%20biggest%20technical%20difference%20between,type%20of%20object%20called%20str%20.

Python vs Matlab II

Interpreted
High level and easy to use



Very scarce support for:
- formats: json, yaml
- Technologies: rest api, grpc, kafka…

Poor integration with other technologies.

*https://realpython.com/matlab-vs-python/#:~:text=The%20biggest%20technical%20difference%20between,type%20of%20object%20called%20str%20.

Course roadmap

1) **Theory and system configuration (1d)**

What

Why

Why

How

2) Practical sessions with Python (3d)

Basic concepts in Python

Common 3rd party packages

Advanced topics in Python

3) Python Project (1d)

Starting with Python

Bare Python

Python from python.org
Select an IDE
Link the IDE to the Python interpreter

ANACONDA®

Great for Data Science
GUI for environments
Easy import-export of env
Easy integration with IDEs
It can be used for Jupyter as well

jupyter

Nice interface to program and share.
Need Python first (Or docker)

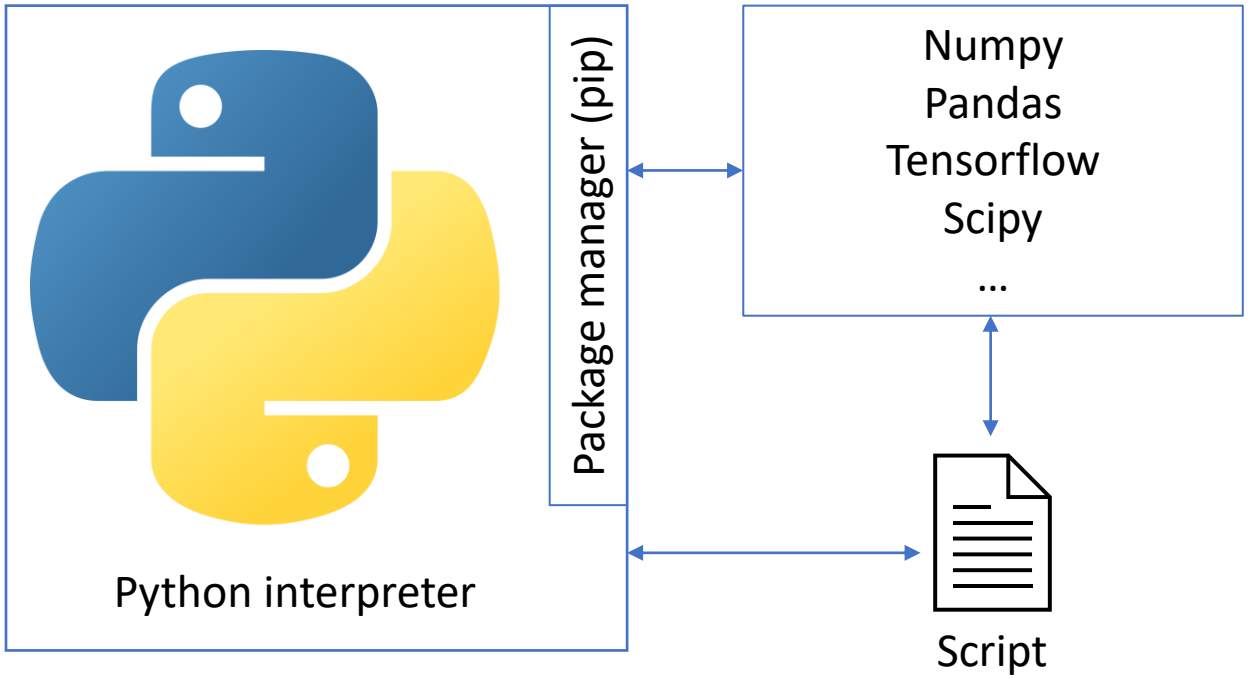
Virtual environments

Python from python.org

pipx
pipenv

- Env per project_1
- Env per project_2
- Env per project_3

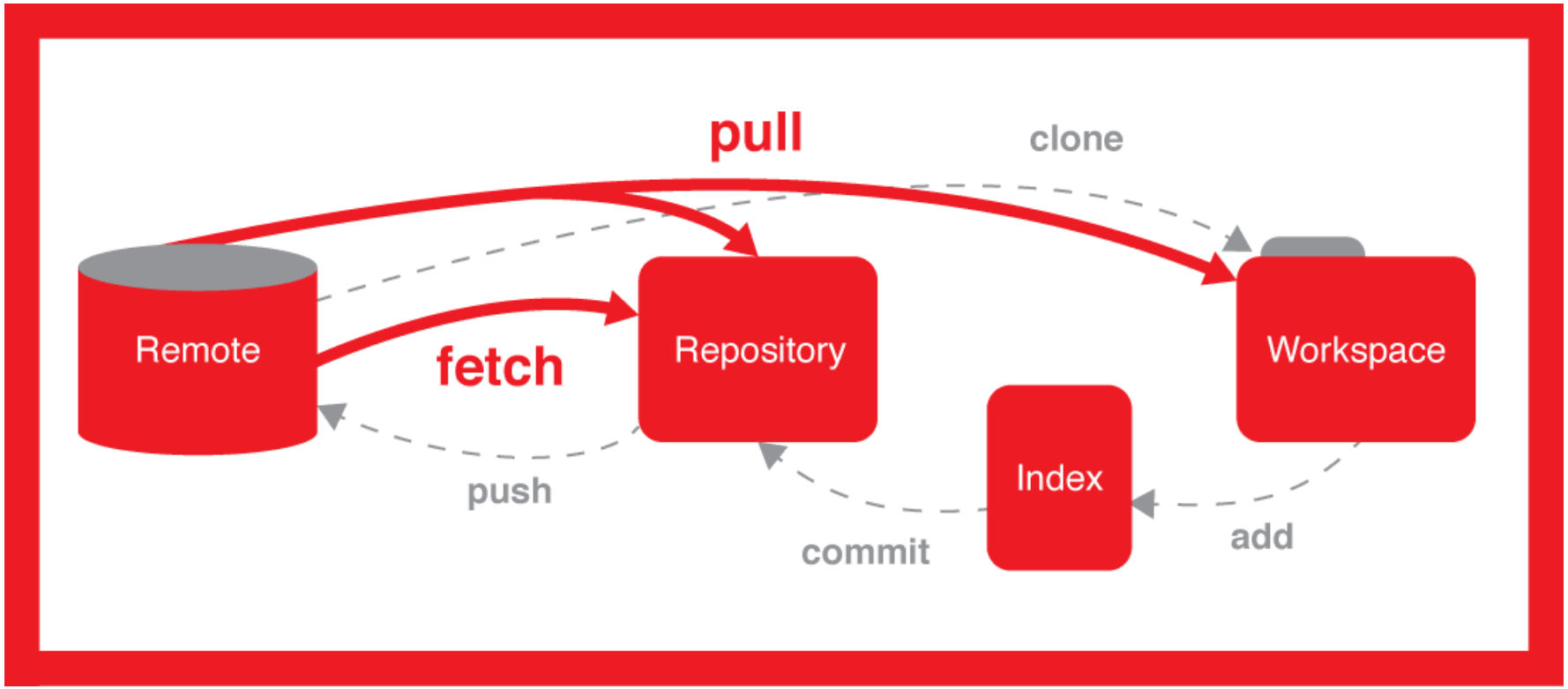
How python works in your system



Version Control

Git

Code collaboration, changes tracking



*https://www.quora.com/What-is-difference-between-push-and-pull-in-git

Installation and configuration guide

Python

1) Go to www.python.org
2) Downloads -> Platform -> Stable releases -> Python 3.9.13 (Select Windows installer version x86/x64)
3) Install it (add it to the Path!)
4) Open a console
    1) py –m pip install –upgrade pip
    2) pip install pipx
    3) pipx install pipenv
    4) pipx ensurepath
    5) Close console
5) In a new console
    1) pipenv –version
6) Create a folder to store your project.
7) Open a console, navigate to the folder
8) pipenv install there.

* Environments are installed under **%UserProfile%\.virtualenvs**

Installation and configuration guide

IDEs I -> Spyder

1) Go to https://www.spyder-ide.org/
2) Downloads -> download the latest
3) Install it
4) Open Spyder
5) Tools -> Preferences-> Python Interpreter -> Use the following python interpreter and look for the environment.

IDEs II -> PyCharm

1) Go to https://www.jetbrains.com/pycharm/
2) Downloads -> download community version
3) Install it
4) Open it

IDEs III -> VS Code

1) Go to https://code.visualstudio.com/
2) Downloads -> download the latest
3) Install it
4) Open VS Code
5) Configure the theme.
6) Open the folder of your project.
7) You will need an extension for Python development

Version control

1) Download git for windows (All options by default)
2) Install GitExtensions (msi package)

* You might need to install other libraries like .Net if required.

** Version control is not required for Python, but it is convenient you familiarize with version control.

Let’s start coding

Python version

Print

Open and write to files

Loop

… and that’s it for today!

QUESTIONS?