Introduction to python

ICTP/Udine ATLAS group
25-29/07/2022

Summary

- Python engine
  - Basic components and setup
- Python language
  - Data types, object oriented programming
- Numpy package
  - Computation with multi dimensional arrays
- Pandas package
  - Tabular data and data preprocessing

What is python?

Python (programming language)

From Wikipedia, the free encyclopedia

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.[31]

What is python?

Python (programming language)

From Wikipedia, the free encyclopedia

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.[31]

+ High level language
  - Rather than dealing with registers, memory addresses, and call stacks, high-level languages deal with variables, arrays, objects, complex arithmetic or boolean expressions, subroutines and functions, loops, etc..

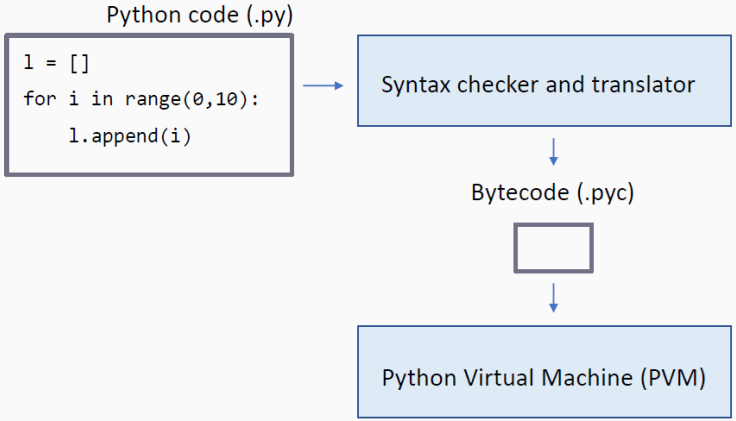
What is python?

Python (programming language)

From Wikipedia, the free encyclopedia

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.[31]

+ Interpreted language
  - Code is not compiled to machine language
  - However, the source code is compiled to an intermediate level, called bytecode
  - For this reason, to run Python programs, you need an interpreter that can execute the bytecode



What is python?

Python (programming language)

From Wikipedia, the free encyclopedia

Python is a high-level, interpreted, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.[31]

+ Minimal syntax
  - No semi-colons to end instructions
  - No braces to define if clauses and for loops
  - No need to specify variables type
  - Etc.

```
struct st *mySt = malloc(sizeof(struct st) +
len * sizeof(mySt->items[0]));
```

```
items = []
```

How to install python

DISCLAIMER: on lxplus@CERN, the setup is a little bit different (see next slides)

+ Ubuntu

Via apt-get: see this guide

- Need sudo rights

TL;DR tip:

```
$ sudo apt-get update
$ sudo apt-get install python3
$ sudo apt get install python3 pip
```

Optional:

```
$ pip3 install ipython
$ pip3 install jupyter
```

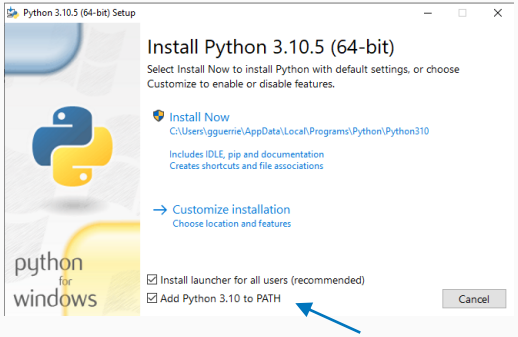
+ Windows

Via .exe file: see this guide

- Normal installation of an executable file

TL;DR tip:

Download the correct release, install the .exe file and add Python to PATH environmental variables.



+ MacOS

Via homebrew: see this guide

- Very similar to Linux

TL;DR tip:

Before installing Homebrew, install CLT for Xcode:

```
$ xcode-select –install
```

Then, after homebrew is installed:

```
$ brew install python
```

How to install python

For the speedrunners:

ANACONDA®

click on the logo

WARNING:
Being easier does not mean being better

Instead of installing separately Python and libraries you can use Anaconda (it may download many files, but it provides an easier installation)

+ https:// www.anaconda.com
    - It will install Python3, iPython , Jupyter and many common Python packages for data science

How to install packages

+ Python is equipped with several useful libraries

Install them with the pip3 command

- pip3 install numpy
- pip3 install pandas
- pip3 install matplotlib
- pip3 install scikit learn

Or with conda

- conda install numpy
- conda install pandas
- conda install matplotlib
- conda install scikit learn

Where do I install those packages?

Normally, you install them locally. Is it the best option?

NO.

How to install packages: python and conda virtual environments

documentation

Create and activate the environment:

```
python3 -m venv myenv
source myenv/bin/activate
```

Install your stuff:

```
pip install -r requirements.txt
pip install numpy
```

…

Deactivate and delete the environment:

```
deactivate
rm -rf myenv/
```

documentation

Create and activate the environment:

```
conda create --name myenv
conda activate myenv
```

Install your stuff:

```
conda install numpy
conda install pandas
```

…

Deactivate and delete the environment:

```
conda deactivate
conda remove --name myenv --all
```

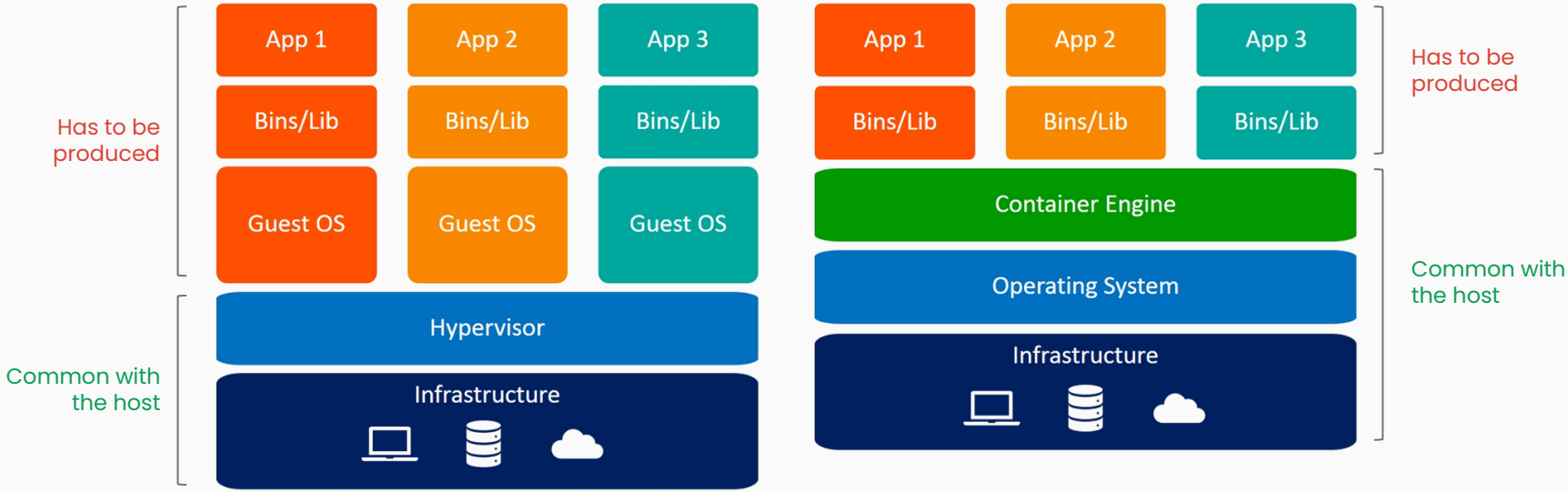
Requirements

Mandatory
Optional

Make sure you have:

ROOT
uproot3
uproot
matplotlib
pandas
numpy
pydot
awkward
vector
scikit-learn
lmfit
jupyterlab_latex
tables

Lazy option: docker containers



Virtual machines

Docker containers

Lazy option: docker containers

1. Install Docker Ubuntu Windows MacOS

Easy
Medium
Hard

+ Ubuntu
  - Set up the initial repo
  - Install Docker engine OR Docker desktop, your choice

+ Windows
  - Install WSL2 distribution for Windows (tip: always use an admin powershell to run commands)
  - Install Docker Desktop

+ MacOS
  - Install from the command line or by downloading package

Lazy option: docker containers

Mandatory
Optional

1. Install Docker Ubuntu Windows MacOS
2. Find and run the container of your interest:

If you are running a notebook:

remove after usage — Ports to forward — name of the docker image — tag

```
sudo docker run -it --rm -v /home:/home -p8888:8888 atlasopendata/root_notebook:codata22
```

interactive — volume to mount

To simply start a bash shell:

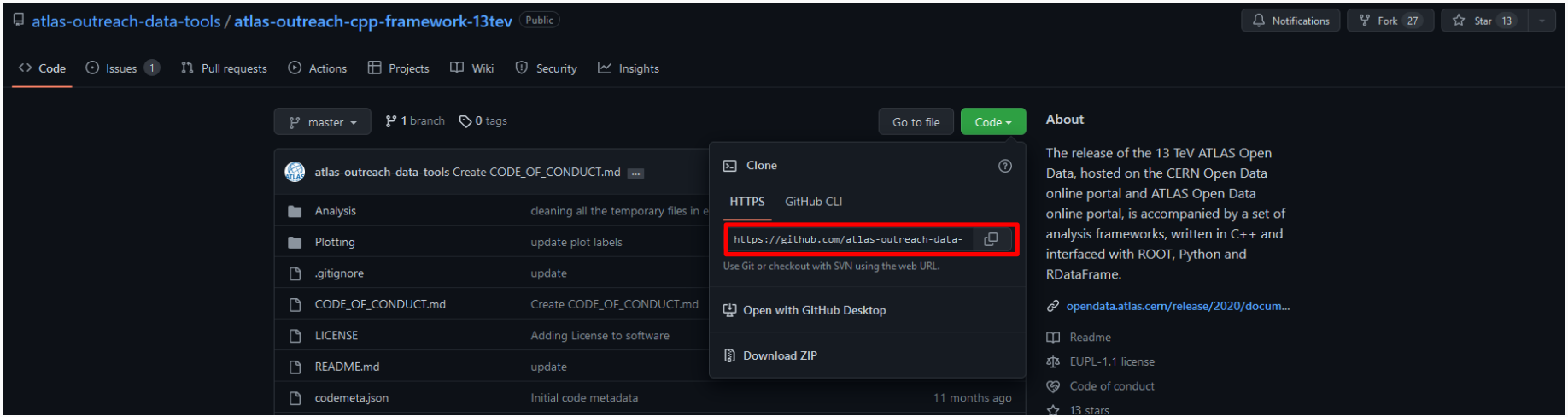
Command to execute at the start

```
sudo docker run -it -v /home:/home atlasopendata/root_notebook:codata22 bash
```

3. Tip: always use the option –v to have your outputs easily accessible on your pc
(also, clone the git repos directly there)

```
-v /scratch/your_host_folder:/home/jovyan/work/random_name
```

Lazy option: docker containers



C++ analysis (today):

```
git clone https://github.com/atlas-outreach-data-tools/atlas-outreach-cpp-framework-13tev.git
```

Python notebooks:

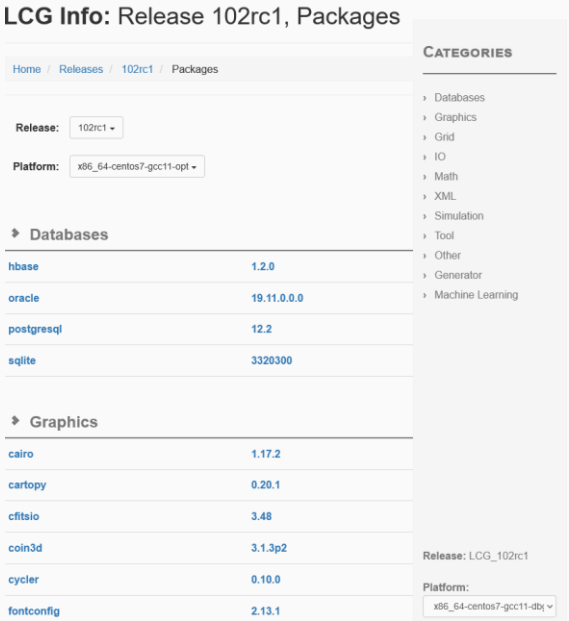
```
git clone https://github.com/atlas-outreach-data-tools/notebooks-collection-opendata.git
```

Even lazier (yet best) option: lxplus@CERN

+ In lxplus the environment can be loaded via LCG:

```
source /cvmfs/sft.cern.ch/lcg/views/LCG_102rc1/x86_64-centos7-gcc11-opt/setup.sh
```

- Instantly load a vast set of libraries
- Access the versions of the packages
- Choose the best combination of features
- Can always add custom libs within a venv



Hands on Python

Notebooks available here