(python) Programming for APC

Lecture 2 : Python 101

Tommaso Ronconi

MA PER SEGUIR VIRTUTE E CANOSCENZA

SISSA

Outline for today

Informed Python usage

- ➔ What is Python?
- ➔ How to set-up a working environment with **conda**
- ➔ A tour of the very basics of the language

$ python -c ‘import stuff’

PYTHON

AH PYTHON, THE INSTAGRAM MODEL OF PROGRAMMING LANGUAGES. YOU'RE A CONNOISSEUR OF SUCCINCT BEAUTY, INDENTATION AND SEMI-COLONLESS CODE MAKES YOU TINGLE. YOU DON'T GET THOSE OLD-SCHOOL DEVS ALWAYS STRESSING OUT ABOUT TYPES AND BASH SCRIPTS. AREN'T THERE IMPORTS FOR THAT? IT'S THREE O'CLOCK AND YOU'RE ALREADY DONE WITH THAT MILESTONE? WHO CARES IF YOU'VE FORGOTTEN HOW ANY OTHER LANGUAGE WORKS. AT LEAST YOU STILL HAVE ALL YOUR HAIR.

The Python programming language

- **high-level**: it has a strong abstraction from the details of the computer. It uses natural language elements, is easy to use and automates significant areas of computing systems like memory management
- **interpreted**: no program compilation action is needed by the user
- **object-oriented**: organised around data, or objects, rather than functions and logic
- **dynamic semantics**: its variables are dynamic (i.e. not *static*) and can change memory size and values during execution

Typical uses

- **Rapid Application Development**: a fast development and testing of ideas and prototype, with less emphasis on planning

Warning

Python may **not** be **the best** programming language **for a big and complex fail-proof application**

- **Scripting**: writing small programs or scripts that control other programs
- **Glue Language**: it is able to deal with libraries compiled with different languages and use them

Main PROs

- Simple, easy to learn **syntax**, **readable friendly** , steep learning curve. Provides a community-written **standard library** with enough flexible implementations for casual usage
- **support of modules and packages** customizable and available to the community from the community

Working in python: the prompt

- **Standard Python prompt**:

```
$ python
Python 3.10.10 (main, Mar 21 2023, 18:45:11) [GCC 11.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

available with any python distribution

- **Interactive Python prompt**:

```
$ ipython
Python 3.10.12 | packaged by conda-forge | (main, Jun 23 2023, 22:40:32) [GCC 12.3.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.14.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
```

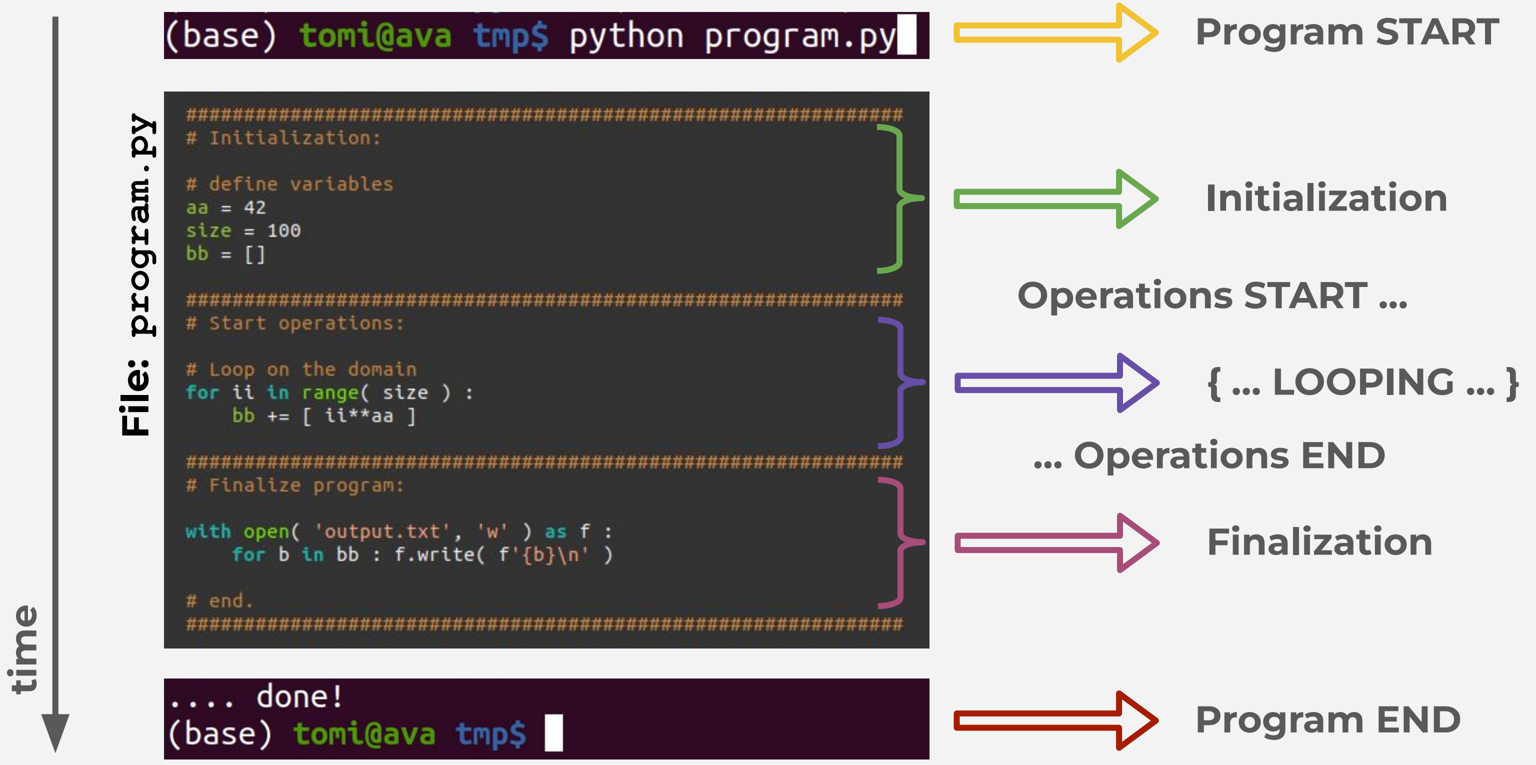
it's a bit better than the standard but you'll have to install it

To **close them**:

Linux:	`Ctrl+d, Ctrl+d`
MacOS:	`Cmd+d, Cmd+d`

Used for:
run simple commands/sets of instructions by **directly typing them in the prompt**: useful for quick checks but
IT DOES NOT REPLACE ACTUAL PROGRAMMING

Working in python: scripts



```
(base) tomi@ava tmp$ python program.py
```

```python
##############################################################
# Initialization:

# define variables
aa = 42
size = 100
bb = []

##############################################################
# Start operations:

# Loop on the domain
for ii in range( size ) :
    bb += [ ii**aa ]

##############################################################
# Finalize program:

with open( 'output.txt', 'w' ) as f :
    for b in bb : f.write( f'{b}\n' )

# end.
##############################################################
```

```
.... done!
(base) tomi@ava tmp$
```

do you remember Shebangs? **#! /bin/python**

Install packages: PyPI & CONDA

- The Python Package Index (PyPI): is a resource to find, install and publish packages, **developed by the community**.

  You can install new packages using the **pip** command:

  ```
  $ pip install something
  ```

  "pip" stands for "pip installs packages"

- CONDA: is a **cross-platform**, **open-source**, **package management system**

  It deals with
  - ➔ download of sources
  - ➔ management of dependencies
  - ➔ **generation of environments**: this aspect here is very useful if we want to escape the …

… DEPENDENCY HELL

We want to install **Miniconda 3** (if you do not have it already, check it!!):

https://docs.conda.io/projects/miniconda/en/latest/

Working in Python: notebooks

What are you still doing here?Move to the terminal.

And that’s all folks! (for today)