Tecnologías Multimedia - Study Guide - Milestone 2: Installation and basic programming with Python

Vicente González Ruiz - Depto Informática - UAL

September 18, 2021

1. Description

The InterCom project [4] are a collection of Python modules written in Python [2], that use a number of packages. Therefore, you will need a Python interpreter and know how to develop/run Python code.

Most of the current Unix-based operating systems (Linux, FreeBSD and OSX) use Python for running some of their “daily tasks”, which means that a Python interpreter is already available. However, usually it is better to use our own interpreter because:

1. We can chose the version of Python and have more flexibility with the packages.
2. We can optimize the compilation of the interpreter depending on our needs (for example, including Tk support or not).
3. By default, all the Python packages will be installed in a different repository of the system packages, which eases the system/user Python-isolation and the removal of the interpreter.

In Windows you need to install Python, yes or yes, from the official website. However, notice that this “guide” only contemplates the installation of Python in Unix-based OS machines, and specifically, in a Xubuntu computer.

As said before, it is very likely that your Unix-like OS has Python installed. However, in order to control the version of the Python interpreter, we will install a dedicated one.

2. What you have to do?

1. Installation of Python.

   (a) Go to YAPT/01-hello_world/02-installation.ipynb [3] and follow the instructions to install the lastest stable version CPython, and create a new virtual environment called `tm` with `pyenv`. Basically (example for `Python 3.9.7`):

   i. Install package dependencies for compiling Python:

   ```
   #sudo apt-get install -y build-essential lib
   sudo apt-get update; sudo apt-get install ma
   ```

   ii. Download Pyenv:

   ```
   #curl https://pyenv.run | bash
   git clone https://github.com/pyenv/pyenv.git
   ```

   iii. (Optional) Compile a dynamic Bash extension to speed up Pyenv.

```
cd ~/.pyenv && src/configure && make -C src ;
```

iv. Define the PYENV_ROOT environment variable to point to the path where you cloned the Pyenv repo. For this, put these lines into ~/.profile before the part that sources ~/.bashrc.

```
export PYENV_ROOT="$HOME/.pyenv"
export PATH="$PYENV_ROOT/bin:$PATH"
```

v. Run:

```
echo 'eval "$(pyenv init --path)"' >> ~/.pr
echo 'eval "$(pyenv init -)"' >> ~/.bashrc
```

to put these lines at the bottom of ~/.profile and ~/.bashrc to enable autocompletion and all subcommands.

vi. Restart your login session for the changes to take effect. E.g. if you're in a GUI session, you need to fully log out and log back in.

vii. List the available Python interpreters:

```
pyenv install —list
```

viii. Install the Python interpreter (and some basic tools such as pip):

```
pyenv install –v 3.9.7
```

ix. Check what it is currently available:

```
pyenv versions
```

You should get something such as:

```
* system (set by /home/<your_home_sir_here>
  3.9.7
```

x. Create a virtual environments for TM

```
pyenv virtualenv 3.9.7 TM
```

```
# Install Python:
```

```
cat << EOF >> ~/.bashrc
export PATH="\$HOME/.pyenv/bin:$PATH"
eval "\$(pyenv init -)"
EOF
source ~/.bashrc
```

(b) Remember that you will need to active it when you want to work in this project:

```
pyenv activate tm
```

It is a good idea to append this to the `~/.bashrc` file.

(c) Install an IDE for programming with Python. I recommend Thonny if you are not used to any other.

```
pip install thonny
```

2. Alternatively (but reducing the chances of solving any possible issue), you can use the Python interpreter shipped with your OS. In this case, still it is strongly recommended to use an specific Python environment for the InterCom project.

3. Python programming.

    (a) You don't need to master Python to follow this course, but it is convenient for you to follow some Python programming tutorial, such as The Python Tutorial [1] if you realize that the language is a setback for you. If you need to start with Python from scratch, an introduction to Python such as this workshop of YAPT [3] could also be helpful. See also ZetCode's Python Tutorial.

3. Timming

There is not time limit for finishing this milestone. Develop it at your own pace. However, notice that we will be using Python very soon.

4. Deliverables

None.

5. Resources

[1] The Python Foundation. The Python Tutorial.

[2] The Python Foundation. The Python Website.

[3] V. González Ruiz. YAPT.

[4] The students of Tecnologías Multimedia at the UAL. The InterCom project.