



UL HPC School

[Advanced] Prototyping with Python

UL High Performance Computing (HPC) Team

C. Parisot

University of Luxembourg (UL), Luxembourg

<http://hpc.uni.lu>

Latest versions available on Github:



UL HPC tutorials:

<https://github.com/ULHPC/tutorials>

UL HPC School:

<http://hpc.uni.lu/hpc-school/>

Tutorial sources:

<https://github.com/ULHPC/tutorials/tree/devel/advanced/Python/>





Summary

- 1 Introduction
- 2 Python for [Fast] Scientific Prototyping
- 3 Using Python on UL HPC Clusters



Main Objectives of this Session

- Run Python code on the cluster
- Install and use your own **Python packages**
 - ↪ Create a **virtual environment** to use several version of the same package
- Compile your code in **C** to have better performances
- Use **Scoop** to distribute your code on the cluster



Summary

- 1 Introduction
- 2 Python for [Fast] Scientific Prototyping**
- 3 Using Python on UL HPC Clusters



Python / Pip

- **pip**: Python package manager
 - ↪ “nice” python packages: `mkdocs...`
 - ↪ Windows: install via [Chocolatey](#)

```
$> pip install <package>
```

```
# install <package>
```



Python / Pip

- **pip**: Python package manager
 - ↪ “nice” python packages: `mkdocs...`
 - ↪ Windows: install via [Chocolatey](#)

```
$> pip install <package> # install <package>
```

```
$> pip install -U pip # upgrade on Linux/Mac OS
```



Python / Pip

- **pip**: Python package manager
 - ↪ “nice” python packages: `mkdocs...`
 - ↪ Windows: install via [Chocolatey](#)

```
$> pip install <package> # install <package>
```

```
$> pip install -U pip # upgrade on Linux/Mac OS
```

- Dump python environment to a requirements file

```
$> pip freeze -l > requirements.txt # as Ruby Gemfiles
```




Pyenv / VirtualEnv / Autoenv

- **pyenv**: \simeq RVM/rbenv for Python
- **virtualenv** \simeq RVM Gemset
- (optional) **autoenv**
 - ↪ Directory-based shell environments
 - ↪ easy config through `.env` file. **Ex:**

```
Terminal -- ash -- 90x23
❯ pyenv versions
2.7.10
* 3.5.0 (Set by /Users/jyu/.pyenv/version)
miniconda3-3.16.0
pypy-2.6.0
❯ python --version
Python 3.5.0
❯ pyenv global pypy-2.6.0
❯ python --version
Python 2.7.9 (209ee98b69288471b0fc2e8de82ce5209eb98b, Jun 01 2015, 17:38:13)
[PyPy 2.6.0 with GCC 4.9.2]
❯ cd /Volumes/treasuredata/jupyter
❯ source /Volumes/treasuredata/jupyter/master/.pyenv/version
miniconda3-3.16.0 (Set by /Volumes/treasuredata/.python-version)
❯ python --version
Python 3.4.3 :: Continuum Analytics, Inc.
❯
```

(rootdir)/.env : autoenv configuration file

`pyversion='head .python-version'`

`pvenv='head .python-virtualenv'`

`pyenv virtualenv --force --quiet ${pyversion} ${pvenv}-${pyversion}`

activate it

`pyenv activate ${pvenv}-${pyversion}`



Summary

- 1 Introduction
- 2 Python for [Fast] Scientific Prototyping
- 3 Using Python on UL HPC Clusters**



Virtualenv

- Install virtualenv on the cluster using pip
- Create your own **virtual environment** to install packages inside it



Use several version of Python

There are **several versions** of Python available on the cluster.
They have been build against **several toolchains**.
The goal of this part is to compare the different versions available on the cluster.



Scoop / Cython

Optimize your code for execution on the HPC cluster

- parallelisation using [Scoop](#)
- compile your Python code in C for faster execution with [Pythran](#) or [Cython](#)
- use [Numpy](#) package to optimize your code



Thank you for your attention...

Questions?

<http://hpc.uni.lu>

The UL High Performance Computing (HPC) Team

University of Luxembourg, Belval Campus:
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu



- 1 Introduction
- 2 Python for [Fast] Scientific Prototyping
- 3 Using Python on UL HPC Clusters