

# Environnements virtuels en Python

Francis Wolinski

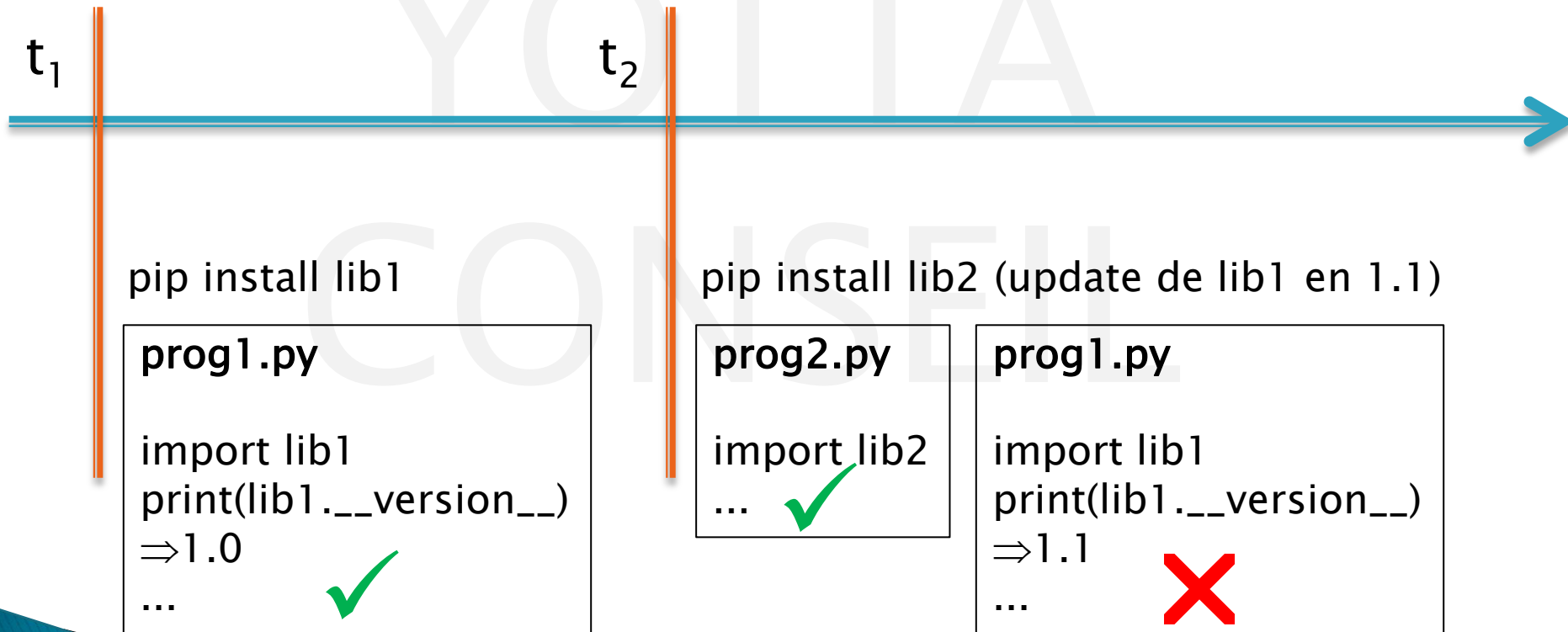
Master 2 TIDE – avril 2020

# Agenda

1. Problématique
2. Solutions
3. Rappel sur pip
4. Solution Python : venv
5. Solution Python : virtualenv
6. Solution Anaconda : conda
7. Références

# 1. Problématique

## Dépendances et versioning des librairies



## 2. Solutions

- ▶ Gérer différents environnements virtuels Python
- ▶ Chaque environnement possède son jeu propre de librairies avec leurs versions respectives
- ▶ En général les environnements sont matérialisés par des répertoires dédiés
- ▶ Il existe différentes solutions techniques :
  - Python Standard Library : venv
  - PyPI/PyPA : virtualenv
  - Anaconda : conda

# 3. Rappels sur pip

- ▶ pip : package manager de Python
- ▶ pour installer une librairie
  - `pip install <module>`
  - `pip install <module>==<version>`
- ▶ pour obtenir la liste des modules installés
  - `pip list`
- ▶ pour obtenir la liste des modules au format requirements.txt
  - `pip freeze > requirements.txt`
- ▶ pour installer les modules à partir d'un fichier requirements.txt
  - `pip install -r < requirements.txt`

## 4. Python venv (Python Standard Library)

- ▶ `python -m venv ENV_DIR`
- ▶ Ensuite dans le chemin `ENV_DIR`, il y a un sous-répertoire 'bin' ou 'Scripts' selon l'OS dans lequel il y a un exécutable 'activate' qui permet d'activer l'environnement et 'deactivate' qui permet de le désactiver
  - [linux] `source ENV_DIR/bin/activate`
  - [Windows] `ENV_DIR\Scripts\activate.bat`
- ▶ Une fois un environnement activé, toutes les installations et modifications avec pip s'effectuent dans l'environnement sans impacter les autres

# 5. Python virtualenv (PyPI, PyPA)

- ▶ Installer virtualenv
  - `pip install virtualenv`
- ▶ Création d'un nouvel environnement
  - `virtualenv ENV_DIR`
- ▶ Ensuite fonctionnement identique à venv
- ▶ Différences avec venv :
  - venv est un sous-ensemble de virtualenv
  - donc virtualenv est plus puissant

# 6. Anaconda conda

- ▶ création d'un environnement
  - `conda create --name <env_name>`
  - `conda create --name <env_name> python==<version>`
- ▶ pour activer un environnement
  - `conda activate <env_name>`
- ▶ liste des environnements existants
  - `conda info --envs`
- ▶ pour installer une librairie dans l'environnement courant
  - `conda install <module>`
  - `conda install <module>==<version>`
- ▶ pour obtenir la liste des modules installés
  - `conda list`
- ▶ pour obtenir la liste des modules au format requirements.txt
  - `conda list -e > requirements.txt`
- ▶ pour installer les modules à partir d'un fichier requirements.txt
  - `conda create --name <env_name> --file requirements.txt`



# 7. Références

- ▶ pip: [https://pip.pypa.io/en/stable/user\\_guide/](https://pip.pypa.io/en/stable/user_guide/)
- ▶ venv: <https://docs.python.org/fr/3/library/venv.html>
- ▶ virtualenv: <https://virtualenv.pypa.io/en/latest/>
- ▶ conda: <https://docs.conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html>
- ▶ voir aussi : <https://python-guide-pt-br.readthedocs.io/fr/latest/dev/virtualenvs.html>