

# PYTHON

---

Présenté par :  
**Xavier TABUTEAU**

# Gestionnaire de paquets PIP

## Gestionnaire de paquets

Dans l'invite de commande de Windows :

```
pip install mon_module      # Installation d'un module (appelé aussi paquet ou dépendance) dans
                             # l'environnement actuel (virtuel ou non).

pip freeze                  # Liste les modules installés.

pip freeze > requirements.txt # Liste les packages de l'environnement actuel dans le fichier
                             # « requirements.txt ».

pip install -r requirements.txt # Installation des paquets listés dans le fichier « requirements.txt » dans
                                # l'environnement actuel.
```

Remarque : Les dépendances ne sont pas automatiquement mises à jour. Si on fait une mise à jour des modules il faut refaire la commande « pip » pour sauvegarder dans un fichier la liste des modules à jour.

« pip » est le successeur du gestionnaire de paquets « easy\_install ».

Ces gestionnaires de paquets se base sur le dépôt de paquets PyPi (Python Package Index).

Si aucun environnement virtuel n'est actif, alors les commandes « pip » fonctionnent dans l'environnement réel général de Python installé sur le système d'exploitation. Il est possible de mettre un chemin relatif ou complet avant le « requirements.txt » ainsi que de changer son nom.

Si une librairie/module n'est pas disponible sur PyPi alors généralement c'est des fichiers sources fourni avec une documentation pour savoir comment les installer dans l'environnement Python. Souvent c'est « python setup.py install ».

# Créer un exécutable avec PyInstaller

## Gestionnaire de paquets

Lignes de commande :

```
pip install PyInstaller # installation de PyInstaller
```

```
PyInstaller -F chemin_complet_ou_relatif\nom_fichier.py # -F pour faire un exécutable indépendant
```

Le fichier exécutable sera dans un dossier « dist » créer à l'emplacement du script.

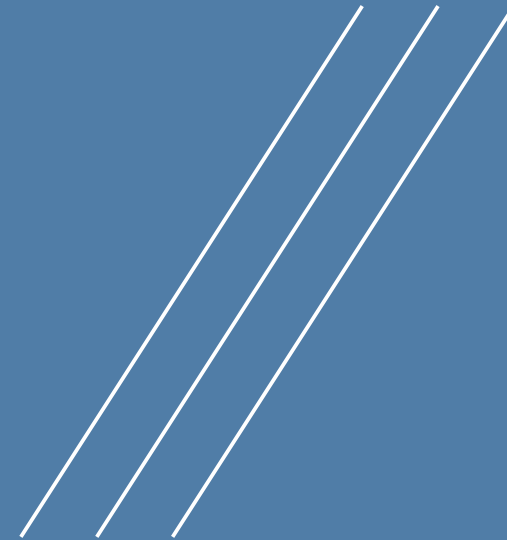
--distpath chemin : permet de changer le dossier où se situe le fichier exécutable.

--noconsole : permet de ne pas lancer la console quand on exécute le programme (utile pour les interfaces graphiques)

--onefile ou -F : permet de créer un exécutable intégrant tout, même python.

Il n'y a pas de différence entre -F et --onefile. -F est l'option courte. --onefile est l'option longue mais plus explicite.

# PYTHON



Présenté par  
**Xavier TABUTEAU**