

## Programmation et projet encadré - L7TI005

Environnements virtuels avec uv

---

Yoann Dupont [yoann.dupont@sorbonne-nouvelle.fr](mailto:yoann.dupont@sorbonne-nouvelle.fr)

Pierre Magistry [pierre.magistry@inalco.fr](mailto:pierre.magistry@inalco.fr)

2025-2026

Université Sorbonne-Nouvelle  
INALCO  
Université Paris-Nanterre

## Environnement virtuel ?

Un environnement virtuel, dans le cadre de Python, est un environnement de travail isolé pour installer des librairies et logiciels.

Il y a différents avantages à utiliser un environnement virtuel en Python :

1. évite les modifications du Python système → installation système reste légère, mais surtout *stable*.
2. permet de créer des dépendances reproductibles à vos projets en évitant (généralement) les conflits
3. dans le pire des cas, on peut juste le supprimer et recommencer sans problème
4. ... et Python force un peu à faire comme ça de toute façon !

On peut faire le parallèle avec une machine virtuelle pour lancer un système d'exploitation.

## pip, venv, uv

Pour gérer les environnements virtuels, on va utiliser trois programmes :

1. pip qui est le programme pour installer des librairies Python
2. venv qui permet de créer un environnement virtuel où installer ces programmes
3. uv qui a sa propre version de ces deux programmes, un peu comme git a des sous-commandes

Si tout va bien, vous avez installé uv lors de l'install party, sinon retournez voir les diapos de l'install party : <https://plurital.org/install-party.pdf>

## Usage en PPE, et peut-être ailleurs

Pour PPE1, on utilisera les environnements virtuels pour installer le programme de nuages de mots qui permettra de visualiser les mots fréquents de votre corpus.

Pour PPE2, on installera quelques bibliothèques et programmes.

Cela pourra également vous servir pour d'autres cours.

## Sommaire des commandes qu'on détaillera

```
uv venv $HOME/venvs/plurital
```

```
source $HOME/venvs/plurital/bin/activate
```

```
uv pip install wordcloud
```

```
wordcloud_cli --help
```

On exemplifie en installant wordcloud et en fixant le lieu de l'environnement virtuel, il n'y a pas besoin de beaucoup plus pour installer d'autres paquets.

## Créer son environnement virtuel

La création d'un environnement virtuel est à faire une fois avant d'installer des programmes. Il s'agit d'une "copie" isolée de votre Python système qu'on peut modifier indépendamment. On peut utiliser environnement "général" pour installer de nombreux programmes, un spécifique gérer les dépendances spécifiques d'un projet, etc. selon les besoins.

Ici, on va utiliser un environnement "général" qu'on pourra utiliser "tous les jours".

```
uv venv $HOME/venvs/plurital
```

Cela va créer un environnement virtuel dans le dossier \$HOME/venvs/plurital.

## Activer son environnement virtuel

Activer un environnement virtuel va donner la priorité à la copie qui a son dossier isolé de votre installation principale. Testez les commandes `whereis python` et `which python` avant et après avoir activé votre environnement virtuel.

Il faut activer son environnement à chaque terminal nouvellement ouvert.

```
source $HOME/venvs/plurital/bin/activate
```

Si on veut déactiver une environnement virtuel on lance :

```
deactivate
```

## Installer un paquet (bibliothèque / programme)

Une fois l'environnement virtuel activé, on peut installer un paquet si on a son nom.

L'exemple suivant installe le paquet wordcloud qui permet de créer des nuages de mots :

```
uv pip install wordcloud
```

Cela installera notamment l'exécutable `wordcloud_cli` dont on peut faire la démonstration.

```
wordcloud_cli --help
```

# Où trouver les programmes à installer ?

Il y a deux aspects à cette question :

1. quels programmes installer pour que la bibliothèque/programme xyz marche ?  
→ quand les choses sont bien faites, un fichier `requirements.txt` donnera les dépendances à d'autres paquets Python.<sup>1</sup>
2. où trouver des programmes que je peux installer ?  
→ les paquets Python sont stockés sur le *PYthon Package Index* (PyPI) :  
<https://pypi.org>.<sup>2</sup>

---

<sup>1</sup>Il peut y avoir des dépendances autres que Python, mais elles n'apparaîtront pas dans ce fichier.

<sup>2</sup>Avec, on vous le souhaite : description, instructions d'installation et lien vers le dépôt.

## Bonus : uv et jupyter lab

Jupyter lab peut en théorie gérer les environnements virtuels pour éviter les installations multiples de jupyter... En pratique, il est plus simple d'installer jupyter lab dans un environnement virtuel existant<sup>3</sup> :

```
uv pip install jupyterlab
```

puis on peut lancer :

```
jupyter lab
```

on peut alors installer simplement les autres paquets requis.

---

<sup>3</sup>On aurait sans doute dû faire comme ça à l'install party.

## En résumé :

- Utilisation des environnements virtuels pour gérer les paquets Python d'une manière efficace et moderne avec uv.
- On fait un test : on propose un environnement de "base" de Python pour plurital, on espère que cet environnement pourra vous suivre jusqu'au M2 !