

# PYTHON

---

Présenté par :  
**Xavier TABUTEAU**

## Environnements virtuels

# Environnements virtuels

Installer Python c'est simple. Mais installer un environnement homogène et performant devient laborieux (un environnement scientifique par exemple) :

- Il faut commencer par isoler son environnement de travail du système.
- Il existe une multitude de librairies et gérer leurs dépendances peut s'avérer difficile.
- Il faut les compiler spécifiquement pour votre système si vous souhaitez exploiter toute la puissance de calcul des processeurs modernes.

Les environnements virtuels Python permettent d'avoir des installations de Python isolées du système et séparées les unes des autres. Cela permet de gérer plusieurs projets sur sa machine de développements, certains utilisant des modules de versions différentes, voir même des versions différentes de Python.

Nous utiliserons le module virtualenv pour la suite du cours.

Installation du package virtualenv :

```
pip install virtualenv
```

Création d'un environnement virtuel :

```
python -m venv monvenv      # avec venv intégrer a Python
```

```
virtualenv monvenv          # avec virtualenv
```

```
# avec virtualenv en spécifiant une version de Python déjà installée sur le PC.
```

```
virtualenv -p "c:\python311\python.exe" monvenv
```

## Environnements virtuels

# Environnements virtuels

Activation d'un environnement virtuel :

Sous windows :

```
.\monvenv\Scripts\activate
```

Sous linux :

```
. monvenv/bin/activate
```

le « . » sous linux demande à exécuter la commande dans le shell courant au lieu d'un autre shell pour ne pas perdre les variables d'environnements.

Remarque :

lorsque l'environnement virtuel est actif, nous avons le nom de l'environnement virtuel entre parenthèses en début de ligne de console.

Désactivation d'un environnement virtuel :

Sous windows :

```
deactivate
```

Sous linux :

```
. monvenv/bin/deactivate
```

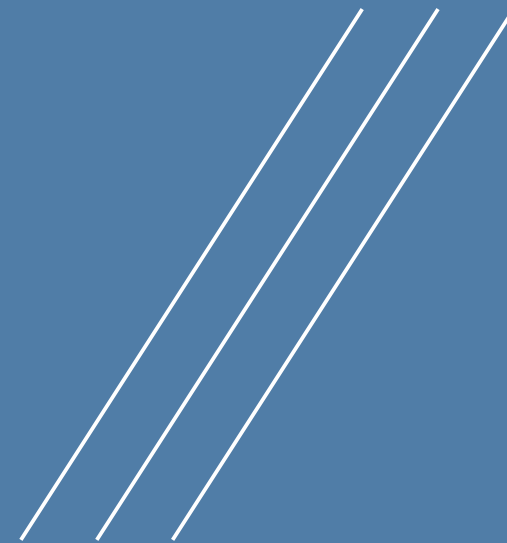
Remarque :

PyCharm créer directement un environnement virtuel à la création d'un nouveau projet.

Menu, file, settings, python interpreter : la fenêtre indique les librairies installées et leurs versions.

Cela indique aussi si une nouvelle version est disponible.

# PYTHON



Présenté par  
**Xavier TABUTEAU**