## Environnement virtuel python

**Un environnement virtuel en Python** est une installation de Python isolée qui permet de gérer des dépendances et des packages spécifiques pour chaque projet. L'utilisation d'un environnement virtuel est recommandée pour éviter les conflits entre les différentes versions de Python et les packages installés sur le système.

Avec un environnement virtuel, vous pouvez créer un environnement de développement propre pour chaque projet et installer les packages nécessaires sans affecter les autres projets ou le système global. Les environnements virtuels permettent également de gérer les versions de Python et les packages requis pour chaque projet. Il existe deux façons possibles pour gérer les environnements virtuels en python : virtualenv et venv

virtualenv et venv sont deux modules différents en Python qui fournissent une fonctionnalité similaire pour créer des environnements virtuels, mais il existe des différences entre eux.

virtualenv est un package tiers qui doit être installé séparément, tandis que venv est inclus dans la bibliothèque standard de Python3.3 et des versions ultérieures.

virtualenv peut être utilisé avec Python 2 et 3, tandis que venv n'est disponible que pour Python 3.

virtualenv offre plus de fonctionnalités et d'options de personnalisation que venv, telles que la possibilité de créer des environnements avec différentes versions de Python et l'option de spécifier l'emplacement où l'environnement virtuel doit être créé.

venv, en revanche, est un module plus simple qui est plus facile à utiliser pour les cas d'utilisation de base.

En général, si vous utilisez Python3 et que vous avez besoin d'un environnement virtuel de base, venv est un bon choix. Cependant, si vous avez besoin de plus d'options de personnalisation ou si vous devez travailler avec plusieurs versions de Python, virtualenv pourrait être un meilleur choix.

Si vous choisissez d’utiliser virtualenv au lieu de venv il faut alors l’installer en utilisant la commande : pip install virtualenv.

Voici les étapes pour créer et activer un environnement virtuel en Python :

1. Ouvrez un terminal ou une invite de commandes.  
  
3. Naviguez jusqu'au répertoire de votre projet où vous voulez créer l'environnement virtuel.  
  
4. Créez un environnement virtuel en utilisant la commande :

Python –m venv <nom\_de\_l’environnement> (si on utilize venv) ;  
 virtualenv <nom\_de\_l’environnement> (pour virtualenv).

5. Cette commande va créer un répertoire appelé <nom\_de\_l’environnement> qui contient un interpréteur Python et une copie du gestionnaire de paquets pip.  
  
6. Activez l'environnement virtuel en exécutant la commande :   
 Sous Windows : <nom\_de\_l’environnement>\Scripts\activate.bat (cmd);  
 <nom\_de\_l’environnement>\Scripts\Activate.ps1 (powershell).   
 Sous Unix ou Linux : source <nom\_de\_l’environnement>/bin/activate  
  
Une fois activé, vous verrez le nom de votre environnement virtuel dans l'invite de commandes.  
Vous pouvez maintenant installer les packages requis dans l'environnement virtuel en utilisant pip.

Par exemple, pip install numpy installera numpy dans l'environnement virtuel.  
Pour désactiver l'environnement virtuel, exécutez simplement la commande deactivate .  
Cela vous ramènera à l'environnement Python par défaut de votre système.  
  
N'oubliez pas d'activer l'environnement virtuel à chaque fois que vous travaillez sur votre projet pour vous assurer que vous utilisez les packages et les dépendances corrects.

**Autres commandes pratiques**

1. pip freeze > requirements.txt : Pour lister les différents packages de votre environnement virtuel et leurs versions dans un fichier nommé requirements.txt .  
  
2. pip install -r requirements.txt : Pour installer tous les packages contenus dans le fichier requirements.text .