

Jour-23 Virtual Environnement

Configuration des environnements virtuels

Pour commencer par Project, il serait préférable d'avoir un environnement virtuel. L'environnement virtuel peut nous aider à créer un environnement isolé ou séparé. Cela nous aidera à éviter les conflits des dépendances entre les projets. Si vous écrivez PIP Freeze sur votre terminal, vous verrez tous les packages installés sur votre ordinateur. Si nous utilisons VirtualEnv, nous accéderons uniquement à des packages spécifiques à ce projet. Ouvrez votre terminal et installez VirtualEnv

```
asabeneh @ asabeneh: ~ $ pip install virtualenv
```

À l'intérieur du dossier de 30 joursofpython, créez un dossier flask_project.

Après avoir installé le package VirtualEnv, accédez à votre dossier de projet et créez un Env virtual en écrivant:

Pour Mac / Linux:

```
asabeneh @ asabeneh: ~ / Desktop / 30daysofpython / flask_project \ $ virtualenv Venv
```

Pour Windows:

```
C: \ Users \ User \ Documents \ 30daysofpython \ flask_project > python -m venv venv
```

Je préfère appeler le nouveau projet Venv, mais n'hésitez pas à le nommer différemment. Vérifions si le VENV a été créé en utilisant la commande LS (ou Dir pour la commande Windows).

```
asabeneh @ asabeneh: ~ / Desktop / 30daysofpython / flask_project $ ls venv /
```

Activons l'environnement virtuel en écrivant la commande suivante dans notre dossier de projet.

Pour Mac / Linux:

```
Asabeneh @ Asabeneh: ~ / Desktop / 30daysofpython / flask_project $ source Venv / bin / activater
```

L'activation de l'environnement virtuel en Windows peut très sur Windows Power Shell et Git Bash.

Pour Windows Power Shell:

```
C: \ Users \ User \ Documents \ 30daysofpython \ flask_project > Venv \ scripts \
activer
```

Pour windows git bash:

```
C: \ Users \ User \ Documents \ 30daysofpython \ flask_project > Venv \ scripts \.
activer
```

Après avoir rédigé la commande d'activation, votre répertoire de projet commencera par V ENV. Voir l'exemple ci-dessous.

```
(Venv) Asabeneh @ Asabeneh: ~ / Desktop / 30daysofpython / flask_project $
```

Maintenant, vérifions les packages disponibles dans ce projet en écrivant PIP Freeze. Vous ne verrez aucun package.

Nous allons faire un petit projet Flask, alors installons le package Flask sur ce projet.

```
(venv) asabeneh @ asabeneh: ~ / Desktop / 30daysofpython / flask_project $ pip install flask
```

Maintenant, écrivons PIP Freeze pour voir une liste de packages installés dans le projet:

```
(Venv) asabeneh @ asabeneh: ~ / dektop / 30daysofpython / flask_project $ pip Freeze click
== 7.0 flask == 1.1.1 itsdangerous == 1.1.0 jinja2 == 2.10.3 MarkupSafe == 1.1.1 Werkzeug
== 0.16.0
```

Lorsque vous terminez, vous devez dactiver le projet actif en utilisant **deactivate**.

```
(Venv) Asabeneh @ Asabeneh: ~ / Desktop / 30daysofpython $ désactiver
```

Les modules nécessaires pour fonctionner avec Flask sont installés. Maintenant, votre répertoire de projet est prêt pour un projet FLASK. Vous devez inclure le Venv sur votre fichier .gitignore pour ne pas le pousser vers GitHub.

Exercices: Jour 23

1. Créez un répertoire de projet avec un environnement virtuel basé sur l'exemple ci-dessus.

Félicitations!