

Лабораторна робота – Дослідження інструментів розробки Python

Цілі та задачі

Частина 1: Запуск віртуальної машини DEVASC

Частина 2: Огляд встановленого середовища Python

Частина 3: Віртуальні середовища Python і PIP

Частина 4: Спільний доступ до віртуального середовища

Довідкова інформація / Сценарій

У цьому курсі передбачається, що студенти мають певну підготовку з Python і практичний досвід програмування. Лабораторна практика з Python розвиває у ваших пальцях «м'язову пам'ять» для більш ефективної роботи на Python. Однак спочатку вам важливо знати, як підтримувати середовище розробки Python.

У цій лабораторній роботі ви ознайомитеся зі встановленим середовищем Python, віртуальними середовищами Python та PIP.

Необхідні ресурси

- Один ПК із вибраною вами операційною системою
- Virtual Box або VMWare
- Віртуальна машина DEVASC

Інструкції

Частина 1: Запуск віртуальної машини DEVASC

Якщо ви ще не виконали **Лабораторну роботу – Встановлення віртуального лабораторного середовища**, зробіть це зараз. Якщо ви вже завершили дану лабораторну роботу, запустіть **BM DEVASC**.

Частина 2: Огляд встановленого середовища Python

За допомогою цих команд ви зможете дізнатися більше про локальне середовище Python.

- a. У BM DEVASC можна перевірити вже встановлену версію Python за допомогою команди **python3 -V**.

```
devasc@labvm:~$ python3 -V
Python 3.8.2
devasc@labvm:~$
```

- b. Щоб переглянути каталог для локального середовища Python, скористайтеся командою **which python3**.

```
devasc@labvm:~$ which python3
/usr/bin/python3
devasc@labvm:~$
```

Частина 3: Віртуальні середовища Python і PIP

PIP розшифровується як Pip Installs Packages. Багато людей спочатку дізнаються про PIP і починають використовувати команди **pip3 install** для загальносистемного встановлення Python. Коли ви запускаєте команду **pip3 install**, можете ввести конкуруючі залежності до вашої системи, які можуть і не знадобитись для всіх проектів Python. Тому найкращий спосіб - увімкнути віртуальне середовище Python. Потім інстальуйте в цьому віртуальному середовищі тільки ті пакети, які потрібні для проекту. Таким чином, ви точно знатимете, які пакети встановлені в даному налаштуванні. Ви можете легко змінювати ці залежності пакетів під час переходу до нового віртуального середовища, а не порушувати або створювати проблеми через конкуруючі версії програмного забезпечення.

Щоб встановити віртуальне середовище Python, скористайтесь інструментом **venv** Python 3, а потім активуйте віртуальне середовище, як показано у наступних кроках.

Крок 1: Створіть віртуальне середовище Python 3.

- a. У віртуальній машині DEVASC змініть каталог **labs/devnet-src/python**.

```
devasc@labvm:~$ cd labs/devnet-src/python/  
devasc@labvm:~/labs/devnet-src/python$
```

- b. Введіть наведену нижче команду, щоб використати інструмент **venv** для створення віртуального середовища Python 3 з іменем **devfun**. Перемикач **-m** вказує Python, що потрібно запустити модуль **venv**. Ім'я вибирає програміст.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devfun  
devasc@labvm:~/labs/devnet-src/python$
```

Крок 2: Активуйте і протестуйте віртуальне середовище Python 3.

- a. Активуйте віртуальне середовище. Запрошення командного рядка зміниться, вказуючи ім'я середовища, в якому ви зараз працюєте. У даному прикладі це **devfun**. Тепер, коли ви використовуєте форму команди **pip3 install**, система буде встановлювати пакети тільки для активного віртуального середовища.

```
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate  
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

- b. Виконайте команду **pip3 freeze**, щоб переконатися, що у середовищі **devfun** не встановлено додаткових пакетів Python.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze  
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

- c. Тепер встановіть пакет **requests** Python у середовищі **devfun**.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 install requests  
Collecting requests  
  Downloading requests-2.23.0-py2.py3-none-any.whl (58 kB)  
    |████████████████████████████████████████| 58 kB 290 kB/s  
Collecting urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1  
  Downloading urllib3-1.25.9-py2.py3-none-any.whl (126 kB)  
    |████████████████████████████████████████| 126 kB 1.7 MB/s  
Collecting idna<3,>=2.5  
  Downloading idna-2.9-py2.py3-none-any.whl (58 kB)  
    |████████████████████████████████████████| 58 kB 18.3 MB/s  
Collecting certifi>=2017.4.17  
  Downloading certifi-2020.4.5.1-py2.py3-none-any.whl (157 kB)  
    |████████████████████████████████████████| 157 kB 19.8 MB/s
```

```
Collecting chardet<4,>=3.0.2
  Downloading chardet-3.0.4-py2.py3-none-any.whl (133 kB)
    |████████████████████████████████████████| 133 kB 59.2 MB/s
Installing collected packages: urllib3, idna, certifi, chardet, requests
Successfully installed certifi-2020.4.5.1 chardet-3.0.4 idna-2.9 requests-2.23.0
urllib3-1.25.9
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

- d. Знову введіть команду **pip3 freeze**, щоб побачити пакети, встановлені зараз у середовищі **devfun**.

Примітка: Ваш перелік пакетів і номери версій можуть дещо відрізнятись.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze
certifi==2020.4.5.1
chardet==3.0.4
idna==2.10
requests==2.24.0
urllib3==1.25.9
(devfun) devasc@labvm:~/labs/devnet-src/python$
```

- e. Щоб деактивувати віртуальне середовище і повернутися до своєї системи, введіть команду **deactivate**.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$
```

Крок 3: Перевірте поточні пакети, встановлені в системному середовищі.

- a. Введіть загальносистемну команду **python3 -m pip freeze**, щоб побачити, які пакети встановлені у системному середовищі.

Примітка: Оскільки Python 3 викликається за допомогою наступної команди, ви використовуєте лише **pip** замість **pip3**.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze
ansible==2.9.4
apache-libcloud==2.8.0
appdirs==1.4.3
argcomplete==1.8.1
astroid==2.3.3
bcrypt==3.1.7
blinker==1.4
certifi==2019.11.28
<output omitted>
xmldict==0.12.0
zipp==1.0.0
devasc@labvm:~/labs/devnet-src/python$
```

- b. Якщо ви хочете швидко знайти версію встановленого пакета, передайте результати виконання команді **grep**. Введіть наведене нижче, щоб переглянути версію пакета запитів, який встановлено.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m pip freeze | grep requests
requests==2.22.0
requests-kerberos==0.12.0
requests-ntlm==1.1.0
requests-toolbelt==0.9.1
```

```
requests-unixsocket==0.2.0
devasc@labvm:~/labs/devnet-src/python$
```

Частина 4: Спільний доступ до віртуального середовища

Результат виконання команди **pip3 freeze** виводиться у специфічному форматі з певної причини. Ви можете використовувати всі перераховані залежності, щоб інші користувачі, які хочуть працювати над тим самим проектом, що і ви, могли отримати те ж саме середовище, що і ваше.

Розробник може створити файл вимог, наприклад **requirements.txt**, за допомогою команди **pip3 freeze > requirements.txt**. Тоді інший розробник з іншого активованого віртуального середовища може використати команду **pip3 install -r requirements.txt**, щоб встановити потрібні проекту пакети.

- a. Повторно активуйте віртуальне середовище **devfun**.

```
devasc@labvm:~/labs/devnet-src/python$ source devfun/bin/activate
(devfun) devasc@labvm:
```

- b. Виведіть результати виконання команди **pip3 freeze** у текстовий файл під назвою **requirements.txt**.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze >
requirements.txt
```

- c. Деактивуйте віртуальне середовище **devfun**. Ви можете скористатись командою **ls**, щоб побачити, що файл **requirements.txt** знаходиться в каталозі **/python**.

```
(devfun) devasc@labvm:~/labs/devnet-src/python$ deactivate
devasc@labvm:~/labs/devnet-src/python$ ls
devfun file-access-input.py if-acl.py requirements.txt
devices.txt file-access.py if-vlan.py while-loop.py
devnew hello-world.py person-info.py
```

- d. Створіть і активуйте нове віртуальне середовище Python під назвою **devnew**.

```
devasc@labvm:~/labs/devnet-src/python$ python3 -m venv devnew
devasc@labvm:~/labs/devnet-src/python$ source devnew/bin/activate
(devnew) devasc@labvm:~/labs/devnet-src/python$
```

- e. Використайте команду **pip3 install -r requirements.txt**, щоб встановити ті ж пакети, які встановлені у віртуальному середовищі **devfun**.

```
(devnew) devasc@labvm:~/labs/devnet-src/python$ pip3 install -r
requirements.txt
Collecting certifi==2020.4.5.1
  Using cached certifi-2020.4.5.1-py2.py3-none-any.whl (157 kB)
Collecting chardet==3.0.4
  Using cached chardet-3.0.4-py2.py3-none-any.whl (133 kB)
Collecting idna==2.9
  Using cached idna-2.9-py2.py3-none-any.whl (58 kB)
Requirement already satisfied: pkg-resources==0.0.0 in ./devnew/lib/python3.8/site-
packages (from -r requirements.txt (line 4)) (0.0.0)
Collecting requests==2.23.0
  Using cached requests-2.23.0-py2.py3-none-any.whl (58 kB)
Collecting urllib3==1.25.9
  Using cached urllib3-1.25.9-py2.py3-none-any.whl (126 kB)
Installing collected packages: certifi, chardet, idna, urllib3, requests
Successfully installed certifi-2020.4.5.1 chardet-3.0.4 idna-2.9 requests-2.23.0
urllib3-1.25.9
```

```
(devnew) devasc@labvm:~/labs/devnet-src/python$
```

- f. При введенні **pip3 freeze** у середовищі **devnew** ви повинні побачити наступне:

```
(devnew) devasc@labvm:~/labs/devnet-src/python$ pip3 freeze  
certifi==2020.4.5.1  
chardet==3.0.4  
idna==2.9  
pkg-resources==0.0.0  
requests==2.23.0  
urllib3==1.25.9  
(devnew) devasc@labvm:~/labs/devnet-src/python$
```

- g. Деактивуйте віртуальне середовище **devnew**.

```
(devnew) devasc@labvm:~/labs/devnet-src/python$ deactivate  
devasc@labvm:~/labs/devnet-src/python$
```