

Python - Virtual Environment

Python Virtual Environment

Python virtual environments create a virtual installation of Python inside a project directory. Users can then install and manage Python packages for each project. This allows users to be able to install packages and modify their Python environment without fear of breaking packages installed in other environments.

What is Virtual Environment in Python?

A Python virtual environment is:

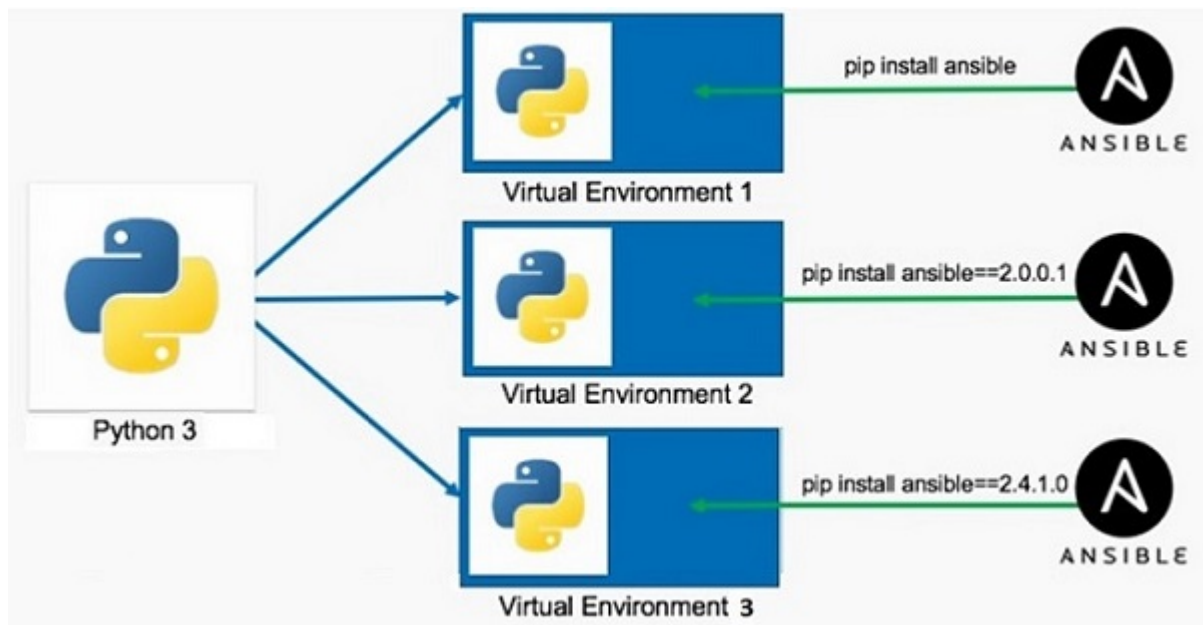
- Considered as disposable.
- Used to contain a specific Python interpreter and software libraries and binaries which are needed to support a project.
- Contained in a directory, conventionally either named `venv` or `.venv` in the project directory.
- Not considered as movable or copyable.

When you install Python software on your computer, it is available for use from anywhere in the filesystem. This is a system-wide installation.

While developing an application in Python, one or more libraries may be required to be installed using the pip utility (e.g., **`pip3 install somelib`**). Moreover, an application (let us say App1) may require a particular version of the library – say **`somelib 1.0`**. At the same time another Python application (for example App2) may require newer version of same library say **`somelib 2.0`**. Hence by installing a new version, the functionality of App1 may be compromised because of conflict between two different versions of same library.

This conflict can be avoided by providing two isolated environments of Python in the same machine. These are called virtual environment. A virtual environment is a separate directory structure containing isolated installation having a local copy of Python interpreter, standard library and other modules.

The following figure shows the purpose of advantage of using virtual environment. Using the global Python installation, more than one virtual environments are created, each having different version of the same library, so that conflict is avoided.



Learn **Python** in-depth with real-world projects through our **Python certification course**. Enroll and become a certified expert to boost your career.

Creation of Virtual Environments in Python using venv

This functionality is supported by **venv** module in standard Python distribution. Use following commands to create a new virtual environment.

```

C:\Users\Acer>md\pythonapp
C:\Users\Acer>cd\pythonapp
C:\pythonapp>python -m venv myvenv
  
```

Here, **myvenv** is the folder in which a new Python virtual environment will be created showing following directory structure –

```

Directory of C:\pythonapp\myvenv
22-02-2023 09:53 <DIR> .
22-02-2023 09:53 <DIR> ..
22-02-2023 09:53 <DIR> Include
22-02-2023 09:53 <DIR> Lib
22-02-2023 09:53 77 pyvenv.cfg
22-02-2023 09:53 <DIR> Scripts
  
```

The utilities for activating and deactivating the virtual environment as well as the local copy of Python interpreter will be placed in the scripts folder.

```

Directory of C:\pythonapp\myvenv\scripts
22-02-2023 09:53 <DIR> .
  
```

```

22-02-2023 09:53 <DIR> ..
22-02-2023 09:53 2,063 activate
22-02-2023 09:53 992 activate.bat
22-02-2023 09:53 19,611 Activate.ps1
22-02-2023 09:53 393 deactivate.bat
22-02-2023 09:53 106,349 pip.exe
22-02-2023 09:53 106,349 pip3.10.exe
22-02-2023 09:53 106,349 pip3.exe
22-02-2023 09:53 242,408 python.exe
22-02-2023 09:53 232,688 pythonw.exe

```

Activating Virtual Environment

To enable this new virtual environment, execute **activate.bat** in Scripts folder.

```

C:\pythonapp>myvenv\scripts\activate
(myvenv) C:\pythonapp>

```

Note the name of the virtual environment in the parentheses. The Scripts folder contains a local copy of Python interpreter. You can start a Python session in this virtual environment.

Checking If Python is Running Inside a Virtual Environment?

To confirm whether this Python session is in virtual environment check the **sys.path**.

```

(myvenv) C:\pythonapp>python
Python 3.10.1 (tags/v3.10.1:2cd268a, Dec 6 2021, 19:10:37) [MSC v.1929
64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import sys
>>> sys.path
['', 'C:\\Python310\\python310.zip', 'C:\\Python310\\DLLs',
'C:\\Python310\\lib', 'C:\\Python310', 'C:\\pythonapp\\myvenv',
'C:\\pythonapp\\myvenv\\lib\\site-packages']
>>>

```

The scripts folder of this virtual environment also contains pip utilities. If you install a package from PyPI, that package will be active only in current virtual environment.

Deactivating Virtual Environment

To deactivate this environment, run **deactivate.bat**.