Day 37: Python Virtual Environments & SHUTIL

What is a Virtual Environment?

A **virtual environment** in Python is an isolated environment on your computer, where you can run and test your Python projects.

It allows you to manage project-specific dependencies without interfering with other projects or the original Python installation.

Think of a virtual environment as a separate container for each Python project. Each container:

- Has its own Python interpreter
- Has its own set of installed packages
- Is isolated from other virtual environments
- Can have different versions of the same package

Creating a Virtual Environment

```
python -m venv .auto
# '.' here signifies a hidden folder
```

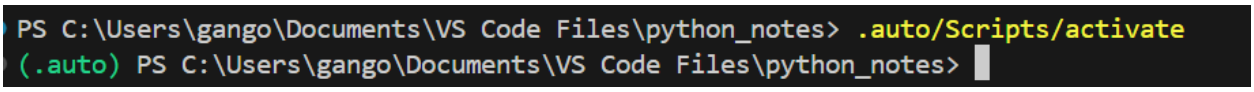
The folder structure will look like this:

```
.auto
  |-Include
  |-Lib
  |-Scripts
  |-.gitignore
  |-pyvenv.cfg
```

Activate Virtual Environment

```
.auto/Scripts/activate
```

Result:



Install packages

```
pip install cowsay
```

Using package

```
# test.py
import cowsay

cowsay.cow("Good Mooooorning!")
```

Delete Virtual Environment

```
# PowerShell
Remove-Item -Recurse -Force .auto

# CMD
rmdir /s /q .auto
```

Shutil

Shutil module offers high-level operation on a file like a copy, create, and remote operation on the file. It comes under Python's standard utility modules. This

module helps in automating the process of copying and removal of files and directories.

You can perform following operation using Shutil:

1. Copying Files to another directory
2. Copying the Metadata along with File
3. Copying the content of one file to another
4. Replicating complete Directory
5. Removing a Directory
6. Finding files

Copying Files to another directory

- `shutil.copy()` method in Python is used to copy the content of the source file to the destination file or directory.
- It also preserves the file's permission mode but other metadata of the file like the file's creation and modification times is not preserved.
- The source must represent a file but the destination can be a file or a directory.
- If the destination is a directory then the file will be copied into the destination using the base filename from the source.

Syntax: shutil.copy(source, destination, *, follow_symlinks = True)

Parameter:

- source: A string representing the path of the source file.
- destination: A string representing the path of the destination file or directory.
- follow_symlinks (optional) : The default value of this parameter is True.
  If it is False and source represents a symbolic link then destination will be created as a symbolic link.

Return Type: This method returns a string which represents the path of newly created file.

```
# Python program to explain shutil.copy() method

# importing shutil module
import shutil

source = r"C:\Users\gango\Documents\VS Code Files\python_notes\python_automation\test.py"
destination = r"C:\Users\gango\Documents\VS Code Files\python_notes\python_automation\TestFolder"

# Copy the content of
# source to destination
dest = shutil.copy(source, destination)

# Print path of newly
# created file
print("Destination path:", dest)
```

Copying the Metadata along with File

- `shutil.copy2()` method in Python is used to copy the content of the source file to the destination file or directory.
- This method is identical to **shutil.copy()** method but it also tries to preserve the file's metadata.

```
Syntax: shutil.copy2(source, destination, *, follow_symlinks = True)

Parameter:
- source: A string representing the path of the source file.
- destination: A string representing the path of the destination file or directory.
```

- follow_symlinks (optional) : The default value of this parameter is True.
    If it is False and source represents a symbolic link then it attempts to copy a
ll metadata from the source symbolic link
    to the newly-created destination symbolic link. This functionality is platform
dependent.

Return Type: This method returns a string which represents the path of newly created file.

```
# Python program to explain shutil.copy2() method

# importing os module
import os

# importing shutil module
import shutil

# path
path = 'csv/'

# List files and directories
# in '/home/User/Documents'
print("Before copying file:")
print(os.listdir(path))


# Source path
source = "csv/main.py"

# Print the metadeta
# of source file
metadata = os.stat(source)
print("Metadata:", metadata, "\n")

# Destination path
```

```
destination = "csv/gfg/check.txt"

# Copy the content of
# source to destination
dest = shutil.copy2(source, destination)

# List files and directories
# in "/home / User / Documents"
print("After copying file:")
print(os.listdir(path))

# Print the metadata
# of the destination file
matadata = os.stat(destination)
print("Metadata:", metadata)

# Print path of newly
# created file
print("Destination path:", dest)
```