38

Day 38: SHUTIL

Contents:

1. Copying the content of one file to another

- **shutil.copyfile()** method in Python is used to copy the content of the source file to the destination file.
- The metadata of the file is not copied.
- Source and destination must represent a file and destination must be writable. If the destination already exists then it will be replaced with the source file otherwise a new file will be created.

```
# Python program to explain shutil.copyfile() method
# importing shutil module
import shutil

# Source path
source = "C:/Users/gango/Documents/VS Code Files/python_notes/python_automation/test.py"

# Destination path - Folder 'TestFolder_2' has to be created first
destination = "C:/Users/gango/Documents/VS Code Files/python_notes/python_automation/TestFolder_2/test_2.py"
```

```
dest = shutil.copyfile(source, destination)

print("Destination path:", dest)
```

Difference between `.copy()` , `.copy2()` & `.copyfile()`

Feature	`copyfile()`	`copy()`	`copy2()`
File contents	✅	✅	✅
Basic metadata (mode, timestamps)	❌	✅	✅
Extended metadata (creation time, flags, xattrs)	❌	❌	✅ (where supported)
Destination can be a directory	❌	✅	✅
Purpose	for raw content copy, lightweight	for typical file copy	for backups, archiving or anything where you need a faithful clone of the original file.

2. Replicating complete Directory

- **shutil.copytree()** method recursively copies an entire directory tree rooted at source to the destination directory. The destination directory, named by must not already exist. It will be created during copying.

```
Syntax:
shutil.copytree(src, dst, symlinks = False, ignore = None, copy_function = cop
y2, ignore_dangling_symlinks = False)

- ignore (optional) : If ignore is given, it must be a callable that will receive as i
ts arguments the directory being visited by copytree(), and a list of its content
s, as returned by os.listdir().
- copy_function (optional): The default value of this parameter is copy2. We c
an use other copy function like copy() for this parameter.
```

```
- ignore_dangling_symlinks (optional) : This parameter value when set to True
is used to put a silence on the exception raised if the file pointed by the symlin
k doesn’t exist.
```

```
# Python program to explain shutil.copytree() method
# importing os module
import os

# importing shutil module
import shutil

# path
path = os.getcwd() + '\\python_automation'
# os.getcwd() - gives current working directory
# In '\\', the first '\' has been used as an escape character for second '\'.
# We can also use '/python_automation'
print(path)

print("Before copying file:")
print(os.listdir(path))

# Source path
src = path + '\\TestFolder_2'

# Destination path
dest = path + '\\TestFolder_3'

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

print("After copying file:")
print(os.listdir(path))

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

3. Removing a Directory

- **shutil.rmtree()** is used to delete an entire directory tree, the path must point to a directory (but not a symbolic link to a directory).

```
Syntax: shutil.rmtree(path, ignore_errors=False, onerror=None)

Parameters:
- path: A path-like object representing a file path. A path-like object is either a
string or bytes object representing a path.
- ignore_errors: If ignore_errors is true, errors resulting from failed removals w
ill be ignored.
- oneerror: If ignore_errors is false or omitted, such errors are handled by calli
ng a handler specified by onerror.
```

```
import shutil
import os

# location
location = os.getcwd()

# directory
dir = "TestFolder_3"



# path
path = os.path.join(location, dir)
print(path)

# removing directory
# Giving error Access is denied, because in the background VS code has this
```

```
file opened
shutil.rmtree(path)
```

4. Finding files

- **shutil.which()** method tells the path to an executable application that would be run if the given **cmd** was called.
- This method can be used to find a file on a computer which is present on the PATH.

```
Syntax: shutil.which(cmd, mode = os.F_OK | os.X_OK, path = None)

Parameters:
- cmd: A string representing the file.
- mode: This parameter specifies mode by which method should execute. os.
F_OK tests existence of the path and os.X_OK Checks if path can be executed
or we can say mode determines if the file exists and executable.
- path: This parameter specifies the path to be used, if no path is specified th
en the results of os.environ() are used
- Return Value: This method returns the path to an executable application
```

```
# importing shutil module
import shutil

# file search
cmd = 'shutil_1'

# Using shutil.which() method
locate = shutil.which(cmd)

# Print result - Have to chenge the folder to current to get the result
print(locate)
```

5. Copying files from one os to another

- Python's `shutil` doesn't inherently "copy between operating systems".
- What matters is whether both source and destination paths are accessible from the same Python process.

```
import shutil

# Source file in WSL
src = r"C:\Users\gango\Documents\VS Code Files\python_notes\python_auto
mation\test.py"

# Destination on Windows drive (mounted under /mnt)
dst = r"\\wsl.localhost\FedoraLinux-42\home\gangoliya\Documents\vs_code_fi
les"

shutil.copy(src, dst)
print("File copied to WSL from Windows")
```