1. How do you distinguish between shutil.copy() and shutil.copytree()?

A. The shutil module in Python provides functions for copying files and directories. The **shutil.copy()** function is used to copy a single file from one location to another. On the other hand, the **shutil.copytree()** function is used to recursively copy an entire directory tree from one location to another.

The primary difference between these two functions is the level of granularity that they offer. **shutil.copy()** provides a basic way to copy a single file from one location to another, while **shutil.copytree()** provides a more powerful way to copy an entire directory tree, including all subdirectories and their contents.

In addition, **shutil.copy()** can be used to preserve file metadata, such as permissions and modification times, while **shutil.copytree()** does not offer this level of granularity. Finally, **shutil.copy()** can be used to copy files to a different name or location, while **shutil.copytree()** will always preserve the directory structure of the original directory tree.

2. What function is used to rename files??

A. The **shutil.move()** function can be used to rename files.

3. What is the difference between the delete functions in the send2trash and shutil modules?

A. The send2trash and shutil modules both provide functions to delete files and directories in Python, but there are some differences between them:

* The **send2trash** module moves files and directories to the system trash or recycle bin, depending on the platform, rather than deleting them permanently. This allows the user to recover the deleted items if necessary. The **send2trash** module provides a **send2trash()** function that takes a file or directory path as an argument.
* The **shutil** module provides functions to delete files and directories permanently. The **os.remove()** function can be used to delete a file, and the **shutil.rmtree()** function can be used to delete a directory and all its contents. These functions cannot be undone, and the deleted items cannot be recovered.

In summary, if we want to delete files or directories in a way that allows the user to recover them later, use the **send2trash** module. If we want to delete files or directories permanently, use the **shutil** module.

4.ZipFile objects have a close() method just like File objects’ close() method. What ZipFile method is equivalent to File objects’ open() method?

A. The equivalent of File objects' open() method for ZipFile objects is the **ZipFile()** function, which is used to open and read the contents of a zip file. The **ZipFile()** function returns a ZipFile object that provides access to the various contents of the zip file, such as its files and directories. It takes two arguments: the name of the zip file and the mode in which to open it (read, write, append, etc.). Once the ZipFile object is created, we can use its methods to access the contents of the zip file.

5. Create a programme that searches a folder tree for files with a certain file extension (such as .pdf or .jpg). Copy these files from whatever location they are in to a new folder.

A. def copy\_files\_with\_extension(src\_folder, dest\_folder, file\_extension):

for foldername, subfolders, filenames in os.walk(src\_folder):

for filename in filenames:

if filename.endswith(file\_extension):

src\_path = os.path.join(foldername, filename)

dest\_path = os.path.join(dest\_folder, filename)

shutil.copy(src\_path, dest\_path)