Topic	Backup	
Class Description	**Student learns to import modules in python. Student moves and copies files using functions available in os module and shutil modules available in python. Student builds a small python program/tool to organize their files into different folders. Student also writes a program which takes backup of all the files in a given folder**	
Class	**C99**	
Class time	**45 mins**	
Goal	● Import Python modules - os and shutil ● Use os and shutil module to create a file organizer program ● Build a program which backs up all the files in a given folder	
Resources Required	● Teacher Resources ○ Visual Code studio ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen ● Student Resources ○ Visual Code studio ○ Laptop with internet connectivity ○ Earphones with mic ○ Notebook and pen	
Class structure	**Warm Up Teacher-led Activity Student-led Activity Wrap up**	**5 mins 15 min 15 min 5 min**

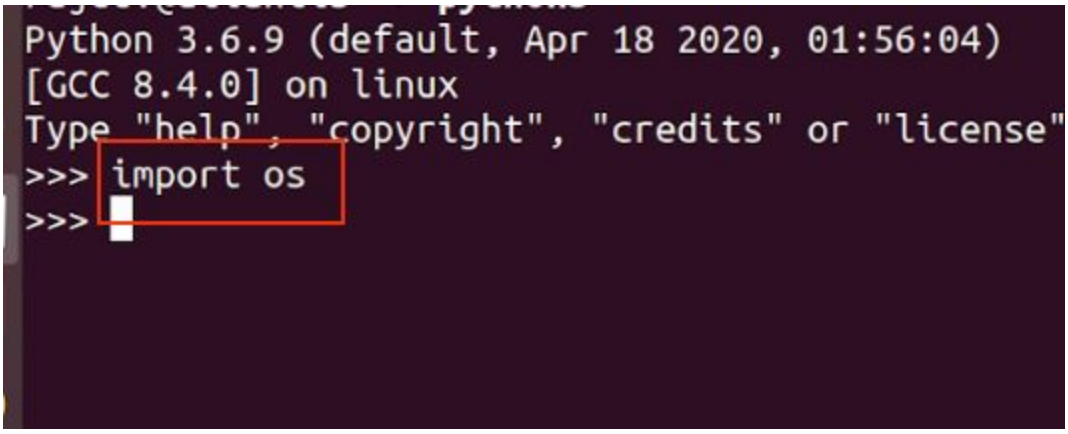
CONTEXT

- **Talk about using Python Libraries**

Class Steps	Teacher Action	Student Action

Step 1: **Warm Up** **(5 mins)**	Hi Do you remember what we learned in the last class?	ESR: We learned about functions. We also learned about functions defined for file object and use them to manipulate text inside files.
	Great! Remember the different libraries we used in javascript while designing games. Similar to Javascript, python has different libraries which have pre-written functions, objects etc. which we will start using in our class today.	-
	Libraries in python are called Modules. We will learn how to import modules into our project and use them in programming. We will learn about two specific modules - os module and shutil module. At the end of the class, we will have built two python tools which will automate - backing up any folder which we want. - organizing different kinds of files - images, videos, songs into separate folders	-
Teacher Initiates Screen Share		
CHALLENGE ● **Import Python modules - os and shutil** ● **Use os and shutil module to create a file organizer program**		

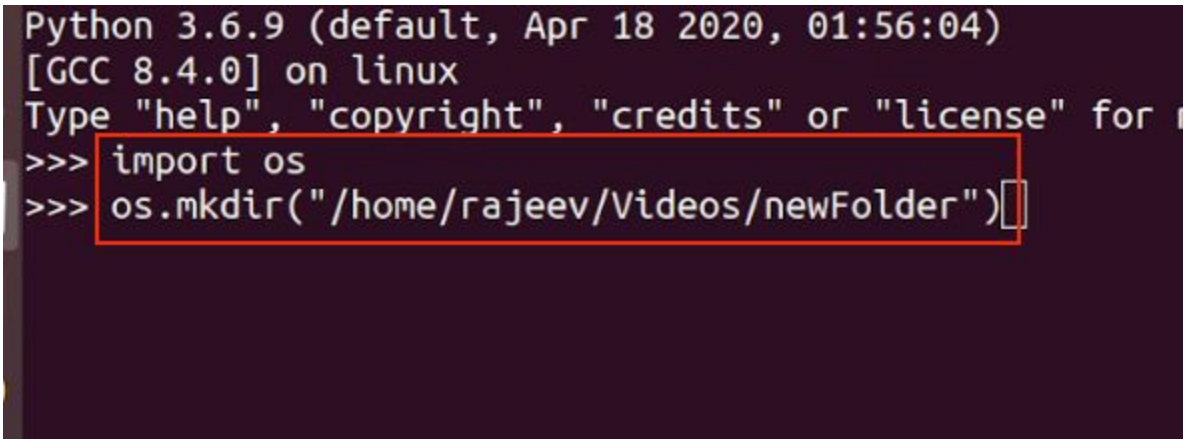
Step 2: Teacher-led Activity (15 min)	Do you know when you boot up your laptop/computer, it is actually running a program? Do you know what the program is called?	ESR: Operating System - Windows - Mac OS - Linux
	Yes! Operating System. Our Operating system provides us with different functionalities like moving files, copying files etc. The OS module in Python provides a way of using operating system dependant functionality. The functions that the OS module provides allows us to interface with the underlying operating system that Python runs on – be that Windows, Mac or. Linux.	-
	In python we use **import** to use a module/library into our project. This is how we will import OS module in python. example:- **import os** We will quickly use the os module to display date and time from the operating system. **os.system("date")** this will give you current date and time Teacher shows how to import and use os module in python shell	Student can import and use os module in their own python shell

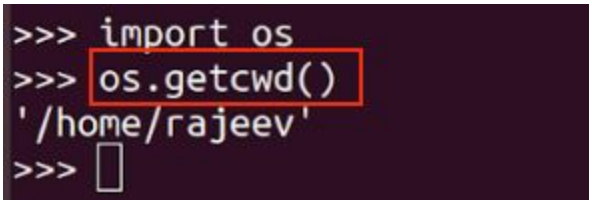
<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>We can also create new files and folders using the os module.
To do so use os.mkdir("path with the name of the file/folder to create")
example:-
folder to create - newfolder

os.mkdir("/home/rajeev/Documents/MovingFile/newFolder")

here the folder i'm creating is newFolder so i have added it at the end of the path.

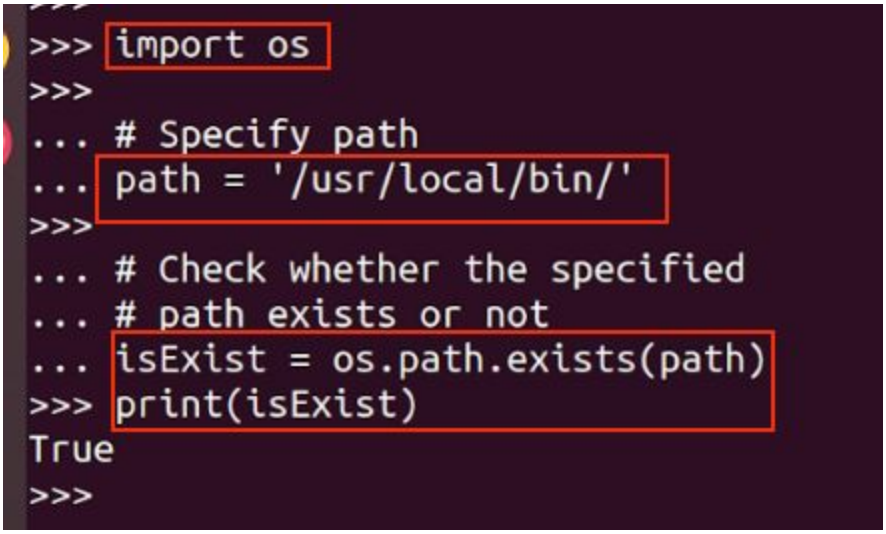
Teacher shows how to create a new directory on python shell</td><td>Student can experiment with the code on their python shell</td></tr>
</table>

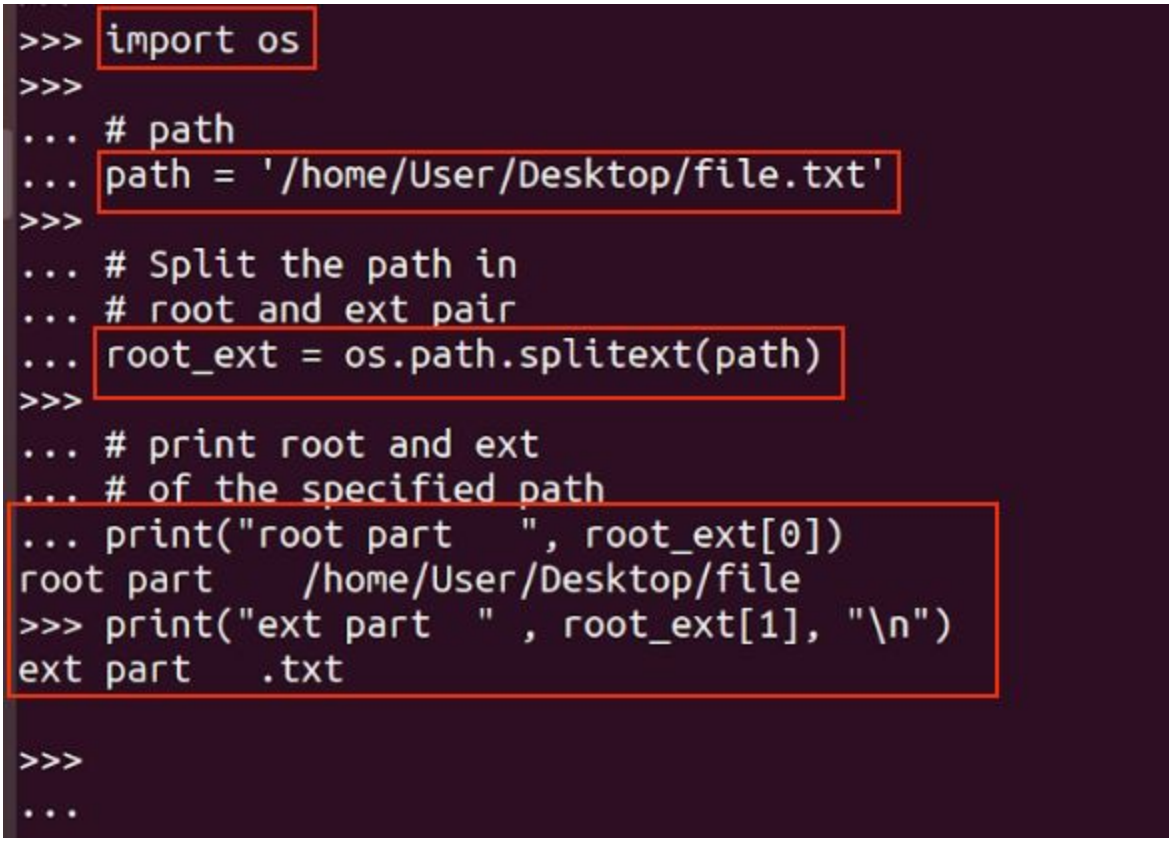
<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>We can also check our current directory by using
os.getcwd()

Run the command on python shell</td><td>-</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>To check if our file exists or not we can use another os module method called os.path.exists()
This method can be also used to check whether the given path refers to an open file descriptor or not.
example:-
# importing os module
import os

# Specify path
path = '/usr/local/bin/'

Check whether the specified
path exists or not</td><td>Student asks questions about tuples, file extensions etc.</td></tr>
</table>

	isExist = os.path.exists(path) **print(isExist)**	
		
	os.path.splitext() method in Python is used to split the path name into a pair root(folder name) and ext(folder extension). Here, ext stands for extension and has the extension portion of the specified path while root is everything except ext part. It return a tuple data type of python. Tuple is same as lists but it cannot be modified and it uses Parentheses instead of brackets eg:- ("file" , ".txt") ext is empty if the specified path does not have any extension. If the specified path has a leading period (‘.’), it will be ignored. example:- # importing os module **import os** # path	-

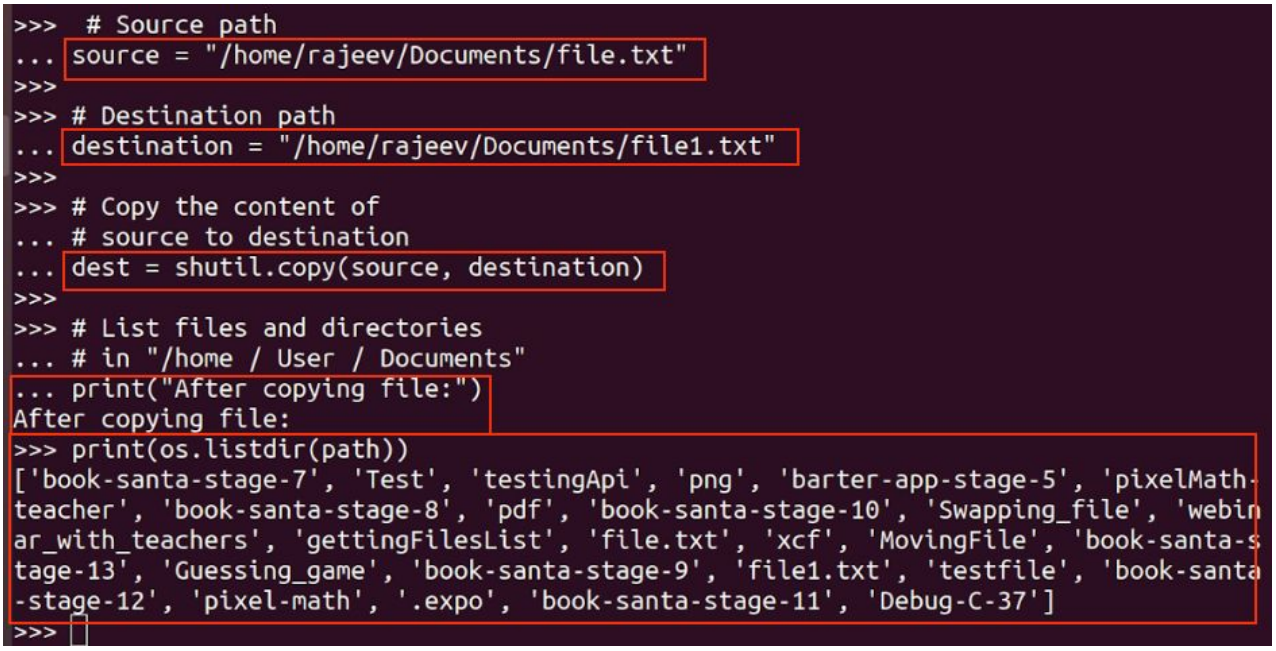
	path = '/home/User/Desktop/file.txt' # Split the path in # root and ext pair **root_ext = os.path.splitext(path)** # print root and ext # of the specified path **print("root part ", root_ext[0])** **print("ext part " , root_ext[1], "\n")**	
		
	To get all the files and folders from a folder os module has os.listdir()	-

	run the command on terminal os.listdir()	

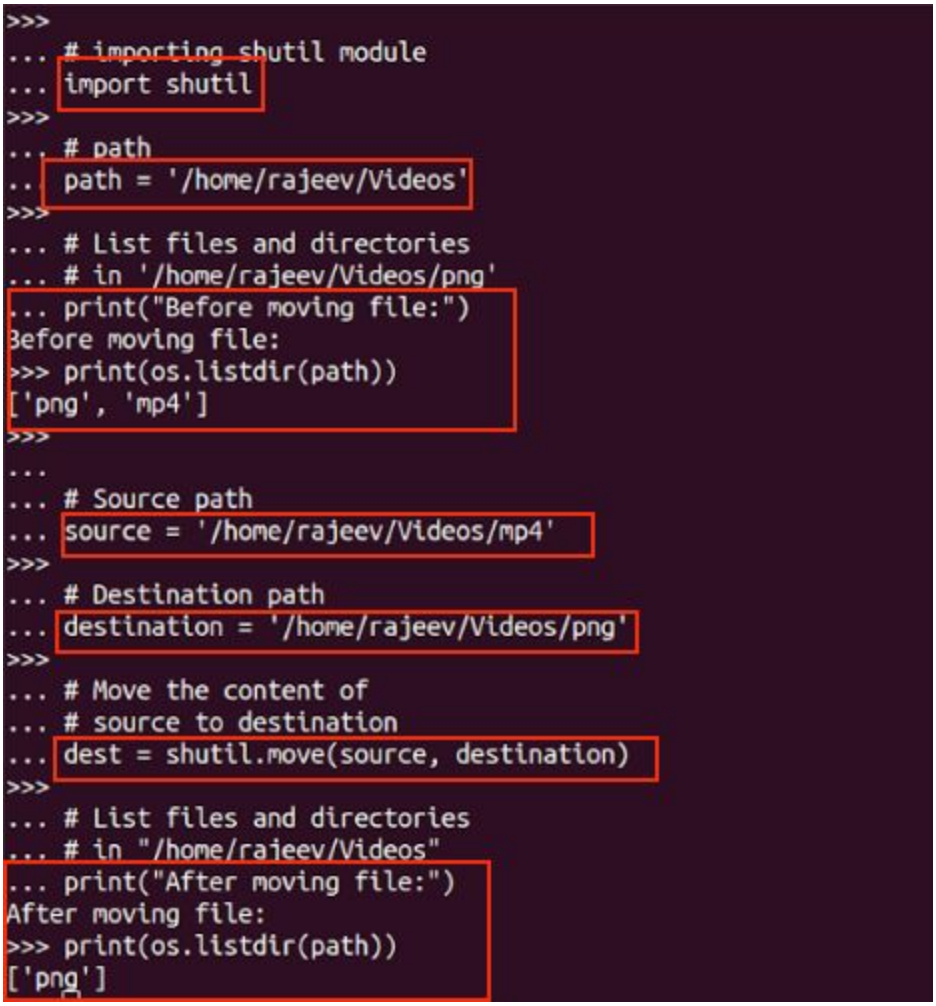
```
>>> os.listdir()
['.config', 'question.md~', 'student02.png', '.crashlytics', 'Public', 'testCardSwap'
, '.bash_logout', '.yarn', 'duet-jekyll-theme', '.viminfo', 'package-lock.json', 'tes
t.js', '.hplip', 'jrnl', '.swp', '.bashrc', '.node_repl_history', '.bash_history', 't
estGit', '.git-credentials', 'node_modules', 'bedTimeStories', 'myStory', 'pixelStick
er.tex', '.test.js.un~', '.thumbnails', '.pixelSticker.tex.un~', 'hp-check.log', 'web
site', '.thunderbird', 'Music', 'pixelMath-teacher', '.httrack.ini', '.lesshst', 'pix
elSticker.log', 'tutor', '.AndroidStudio3.5', 'pixelSticker2.tex', '.viminfz.tmp', 'S
hivam.png', '.react-native-cli', 'test', '.pki', '.java', '.yarnrc', 'tab-navigation.
tar.xz', '.task', 'testFolder', '.gnome', 'firestoreScreenShots', '.gvfs', '..vimrc.u
n~', 'pixelSticker.aux', 'projectDictionary', '.ReviewJS.md.swp', 'quicklisp', '.andr
oid', 'buggy-project', '.viminfo.tmp', '.pixelSticker2.tex.swp', 'texmf', 'vimQuestio
nEditor', 'example.csv', '.vscode', 'grade 4 level breakdown 3', '.mozilla', 'snap',
```

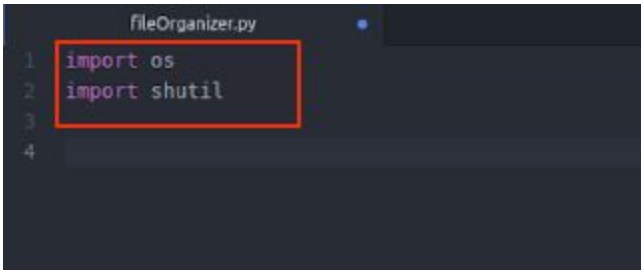
	There is another module **Shutil module** in Python provides many functions of high-level operations on files and collections of files. It comes under Python's standard utility modules. This module helps in automating the process of copying and removal of files and directories.
	
	Use **shutil.copy()** method to copy file from source to destination
	example:-
	
	`# importing os and shutil module`
	`import os`
	`import shutil`
	
	`# path`
	`path = '/home/rajeev/Documents'`
	
	`# List files and directories`
	`# in '/home/rajeev/Documents'`
	`print("Before copying file:")`
	`print(os.listdir(path))`
		-

```
# Source path
source = "/home/rajeev/Documents/file.txt"

# Destination path
destination = "/home/rajeev/Documents/file(copy).txt"

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

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

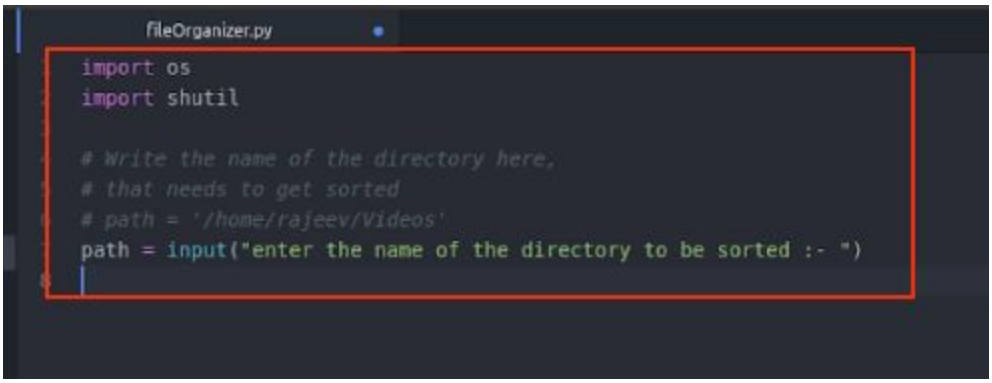


	shutil.move() method Recursively moves a file or directory (source) to another location (destination) and returns the destination. If the destination directory already exists then src is moved inside that directory. Using **shutil.move()** method to move file from source to destination # importing os and shutil module **import os** **import shutil** # path **path = '/home/rajeev/Videos'** # List files and directories # in "/home/rajeev/Videos" **print("Before moving file:")** **print(os.listdir(path))** # Source path **source = "/home/rajeev/Videos/mp4"** # Destination path **destination =** **"/home/rajeev/Videos/png"** # Move the content of # source to destination **dest = shutil.move(source,** **destination)** # List files and directories	

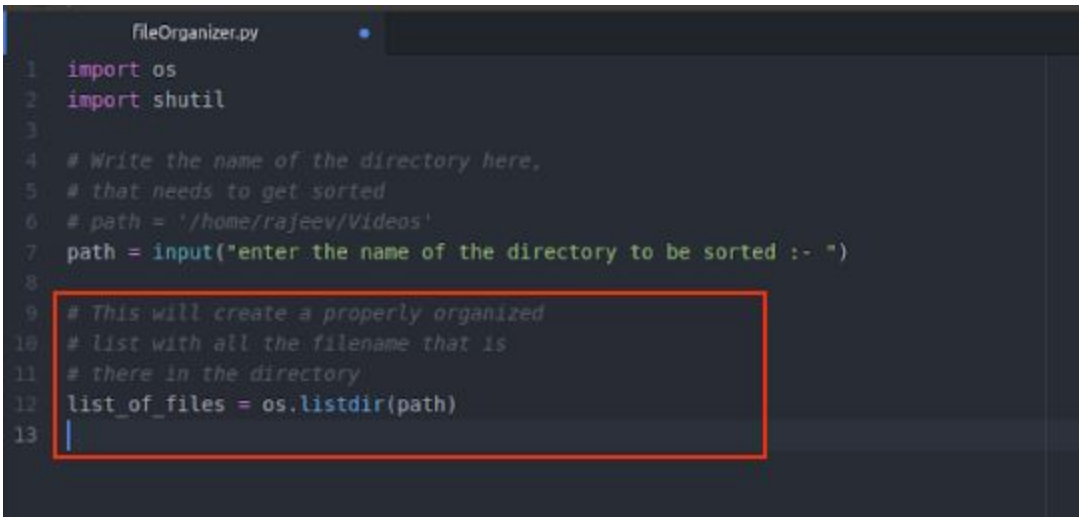
	# in ""/home/rajeev/Videos"	
		
	Now it's time to write some code. Let's write a code to organise our files with different extensions into different folders. <Teacher opens the editor and creates a new file called fileOrganizer.py>	
	First we need to import shutil and os to the file.	<student helps teacher to import the shutil and os>

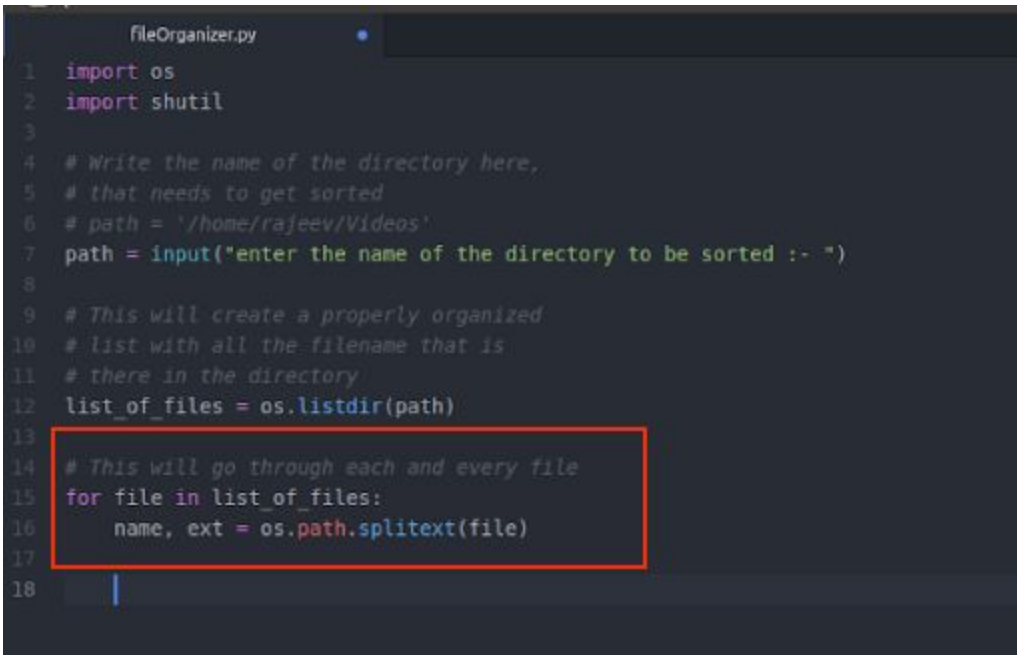
	<Teacher imports shutil and os to the file>	
		

	We will take the path of the folder from the which needs to be sorted. We'll use input() to take the path from the user.
	<teacher uses input function and stores the path in a path variable>
	code:-
	# Write the name of the directory here,
	# that needs to get sorted
	# path = '/home/rajeev/Videos'
	path = input("enter the name of the directory to be sorted :- ")
		<Student helps teacher to use input function and store the path in path variable>

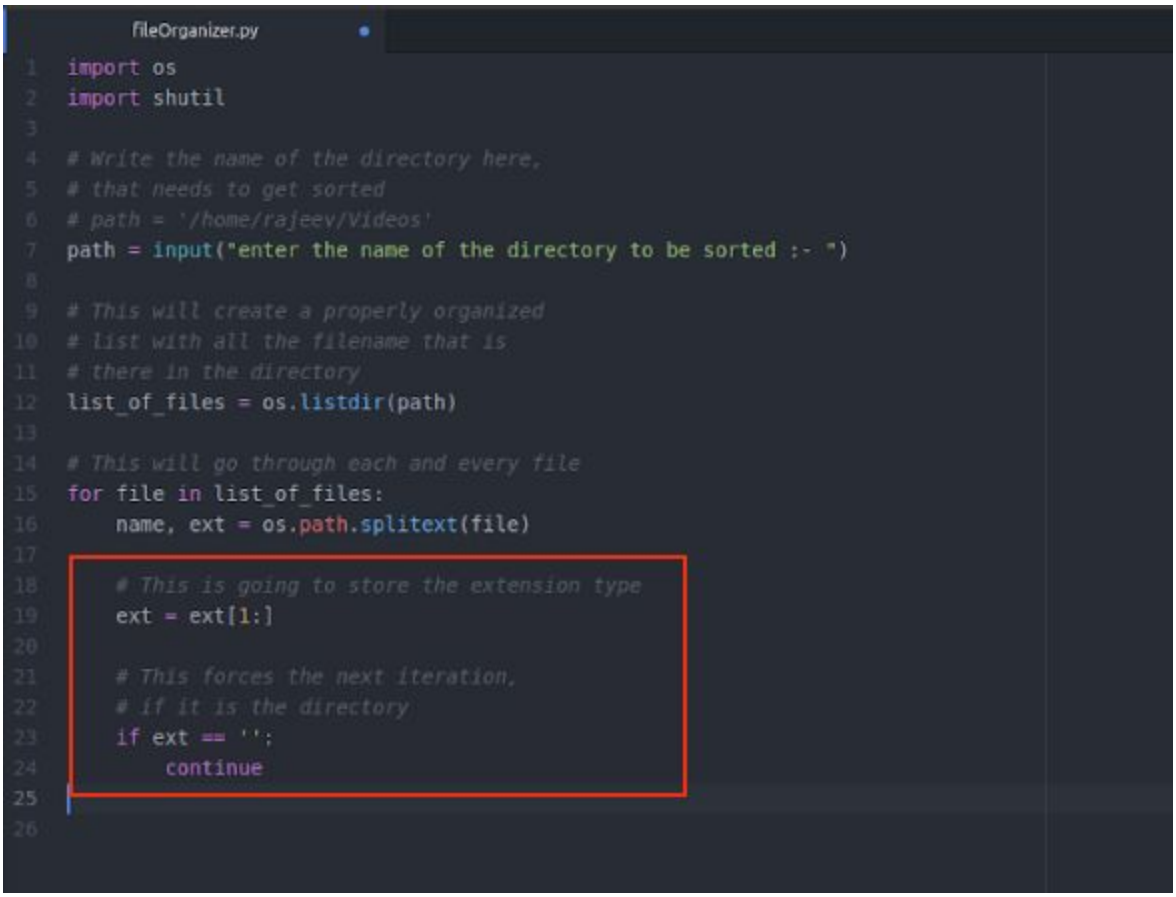
		
	Now we need to get all the files from the folder of the given path so we will use os.listdir() and pass the path to it in the parenthesis. <teacher uses the os.listdir() and stores	<Student helps teacher to use os.listdir() and pass the path to it. and store it in list_of_files variable>

	it in the list_of_files variable>
	code:-
	# This will create a properly organized
	# list with all the filename that is
	# there in the directory
	list_of_files = os.listdir(path)	

		
	we want to seperate the files by their extension types. So first we need to get all the extensions of the files. we will loop on the list_of_files and use os.path.splitext() to get the the extensions. and store them in ext vriable. <Teacher uses the for loop and uses os.path.splitext() to get the extensions and store them in ext variable.> code:- # This will go through each and every file for file in list_of_files: name, ext = os.path.splitext(file)	<Student helps teacher uses the for loop and uses os.path.splitext() to get the extensions and store them in ext variable.>

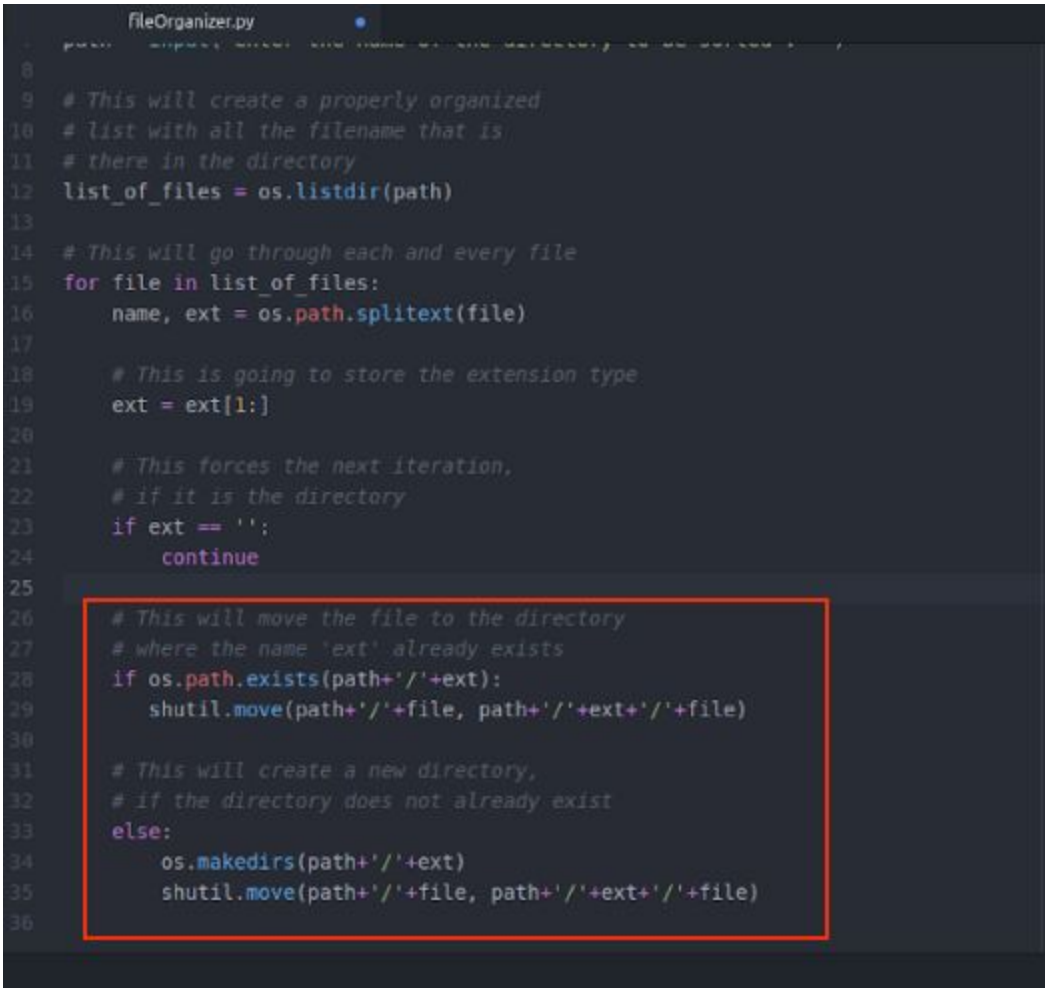


	There is an issue that if our folder contains the some folder then the loop will break as folders don't have any extensions. To avoid that we'll first store all the extensions at a place and then write a if condition to check if extension is empty . If extension is empty then write continue to keep going. <Teacher write code to store all the extensions in a ext variable and write the if condition> code:- # This is going to store the extension type **ext = ext[1:]** # This forces the next iteration, # if it is the directory **if ext == '':** **continue**	<Student helps teacher to write code to store all the extensions in a ext variable and write the if condition>

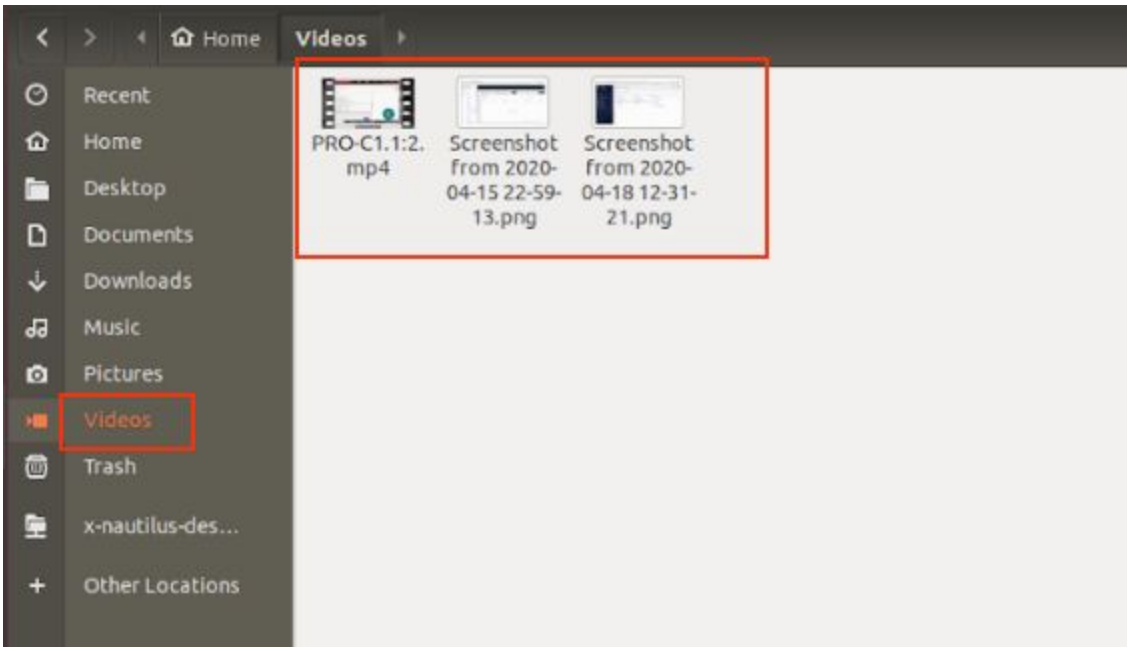


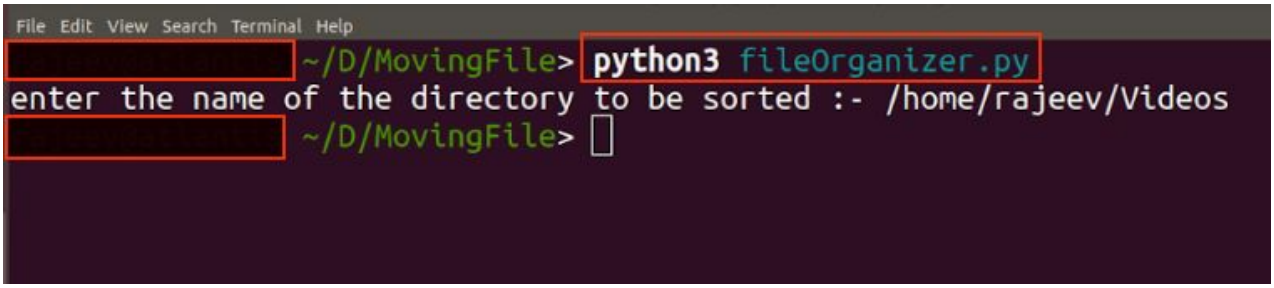
	If there is an existing folder with the extension name, then we want to move the files with that extension to that folder. Else we want to create a new folder with that extension name and move the files to that folder. <Teacher codes to check if there is a existing folder with the extension name then move the files to it, else create new folder with extension name and move files to it> code:- # This will move the file to the directory # where the name 'ext' already exists	<Student helps Teacher code to check if there is a existing folder with the extension name then move the files to it, else create new folder with extension name and move files to it>

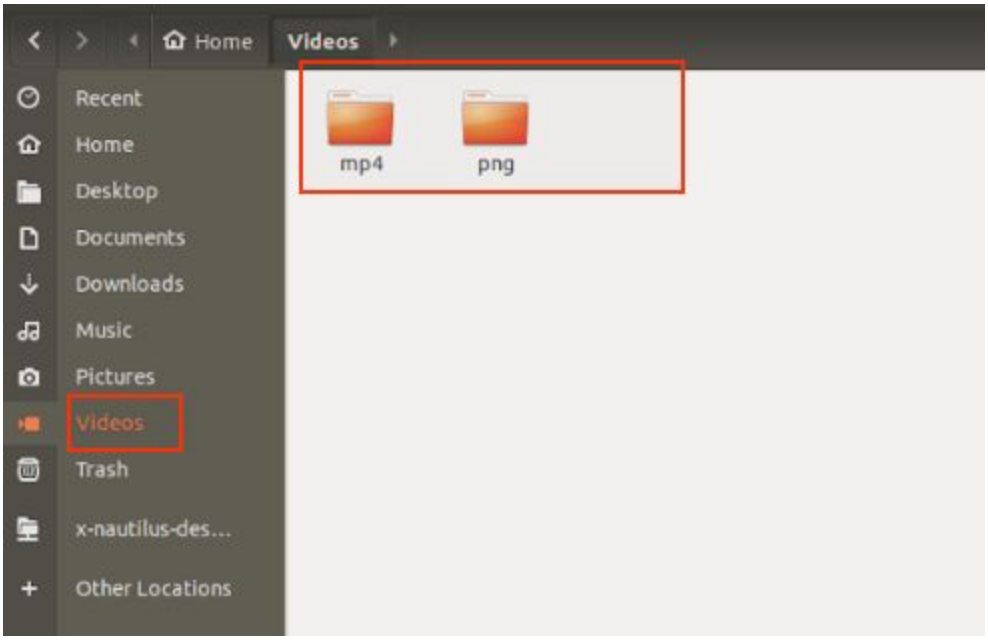
	if os.path.exists(path+'/'+ext): **shutil.move(path+'/'+file, path+'/'+ext+'/'+file)** # This will create a new directory, # if the directory does not already exist **else:** **os.makedirs(path+'/'+ext)** **shutil.move(path+'/'+file, path+'/'+ext+'/'+file)**	



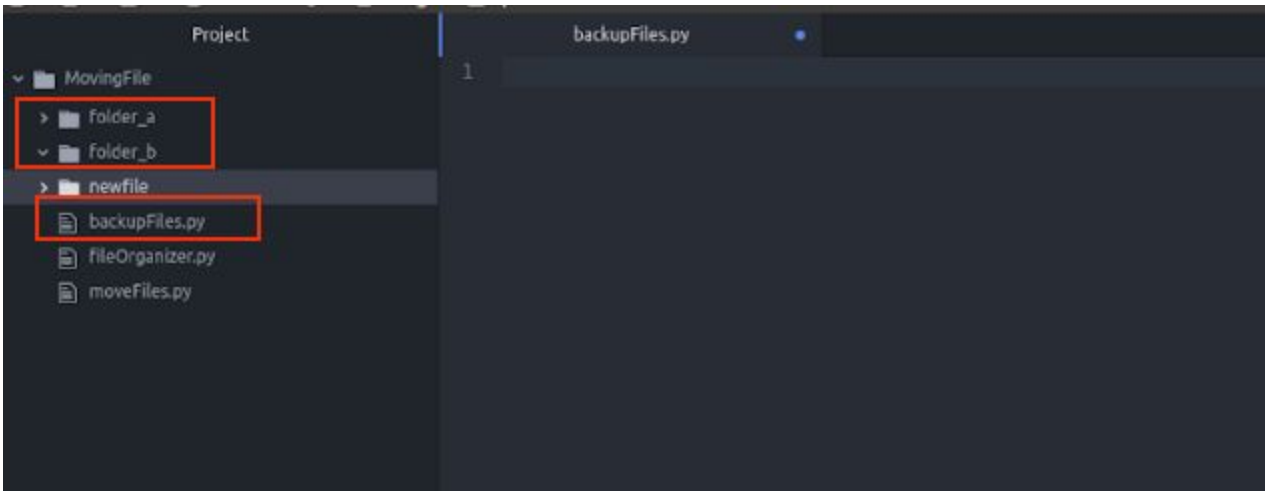
	let's run the code and test for the output. <Teacher navigates to the directory containing the orgaisingFile.py file on the terminal and then run the code using python3 organisingFile.py >	

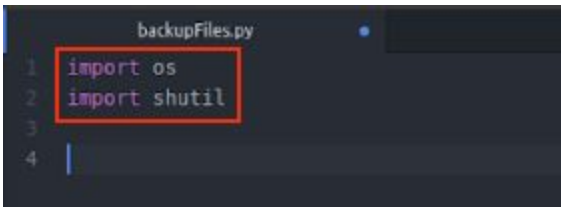






	Ok. Now here is a challenging activity for you. You need to create a program which will backup any folder given to it as input. It should make a copy of ALL the content in that file into a different destination folder. Can you do that?	
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	
• **Ask Student to press ESC key to come back to panel** • **Guide Student to start Screen Share** • **Teacher gets into Fullscreen**		
ACTIVITY • **Build a program which backs up all the files in a given folder**		
Step 3: Student-Led Activity (15 min)	We'll write a program which will take a backup of all our files and store in another folder.	-
	First create a python file named backFiles.py. Also create 2 folders Folder_a, Folder_b. Inside Folder_a create a sample.txt file with some text in it. <Teacher helps student create a the files and folders.>	<Student creates the files>

<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>Let's import os and shutil to our file

code:-
import os
import shutil
<Teacher helps student to import os and shutil></td><td><student imports os and shutil></td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td>We need to get the name of the folder from where the files have to be moved and also get the name of the folder to where the files will be stored.
Let's create two variables source and destination and use input() to get the names from the user.

code:-
source = input("enter source folder name:- ")</td><td><student uses input to get the names of the source and destination folders></td></tr>
</table>

	destination = input('enter destination folder name:- ') <Teacher helps student use input to get the names of the source and destination folders>	
backupFiles.py 1 import os 2 import shutil 3 4 source = input("enter source folder name:- ") 5 destination = input('enter destination folder name:- ') 6 7 8		
	After taking the input we will be adding the "/" at the end to complete the path. code:- **source = source + '/'** **destination = destination + '/'** <Teacher helps the student add "/" at the end of the folder name>	<student adds "/" at the end of the folder name>
backupFiles.py 1 import os 2 import shutil 3 4 source = input("enter source folder name:- ") 5 destination = input('enter destination folder name:- ') 6 7 source = source + '/' 8 destination = destination + '/' 9 10 11		

	Now we need to get all the files from the source folder> os module has a module called os.listdir() we'll use os.listdir() , pass source as path to it and store all the files in list_of_files variable. code:- **list_of_files = os.listdir(source)** <Teacher helps student use os.listdir() and store files in list_of_files variable>	<student uses os.listdir() and store files in list_of_files variable>
		
	Now we need to get the files one by one from the list_of_files and copy them to the destination folder. To do that we'll use for loop on the list of the files and usng shutil.copy we'll copy the files to the destination folder. code:- list_of_files = os.listdir(source) for file in list_of_files: shutil.copy((source+file), destination)	<student uses for loop on the list_of_files and copy the files to destination folder using shutil.copy>

	<Teacher helps student for loop on the list_of_files and copy the files to destination folder using shutil.copy>	
backupFiles.py 1 import os 2 import shutil 3 4 source = input("enter source folder name:- ") 5 destination = input('enter destination folder name:- ') 6 7 source = source + '/' 8 destination = destination + '/' 9 10 list_of_files = os.listdir(source) 11 for file in list_of_files: 12 shutil.copy((source+file), destination) 13		
	Now lets run and test the code.	<Student opens the terminal and goes to the directory which contains the backupFiles.py Using python3 backupFiles.py runs the code> In input puts Folder_a as the source and Folder_b as destination.
File Edit View Search Terminal Help [redacted] ~/D/MovingFile> python3 backupFiles.py enter source folder name:- folder_a enter destination folder name:- folder_b [redacted] ~/D/MovingFile>		
Teacher Guides Student to Stop Screen Share		

FEEDBACK • **Appreciate the student for their class** • **Get them to play around with different ideas, automations which they can build for their system using python**		
Step 4: Wrap-Up (5 min)	Let's quickly wrap up today's class. What did we learn?	ESR: We learned how to use other modules/ libraries in python. We learned about the two python modules - os and shutil and how we can use them to interact with os level functions like moving, copying files etc.
	You can actually think about automating a lot of other functions on your system using these modules. In the present class, we updated all our backup files on our same system. In the next class, we will learn how to take backup on a cloud. Interesting?	Yes!
	Looking forward to the next class then	
Teacher Clicks 		
Additional Activities	Encourage the student to write reflection notes in their reflection journal using markdown.	The student uses the markdown editor to write her/his reflection as a reflection journal.

	Use these as guiding questions: ● What happened today? ○ Describe what happened ○ Code I wrote ● How did I feel after the class? ● What have I learned about programming and developing games? ● What aspects of the class helped me? ● What did I find difficult?	

Activity	Activity Name	Links
Teacher Activity 1	Import module reference	https://www.geeksforgeeks.org/import-module-python/
Teacher Activity 2	os module in python	https://www.geeksforgeeks.org/os-module-python-examples/
Teacher Activity 3	Shutil reference dov	https://docs.python.org/3/library/shutil.html
Teacher Activity 4	solution	https://github.com/whitehatjr/Moving_File