Topic	**Cloud Storage**	
Class Description	**Student learns about dropbox cloud storage. Student builds a small python program/tool to upload his/her files to dropbox**	
Class	**C101**	
Class time	**45 mins**	
Goal	● Create dropbox account and Install dropbox using pip ● Using howdoi to get the best solution to upload files to dropbox ● Building and customizing a python program to get file name or file path as input from user on command line	
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**

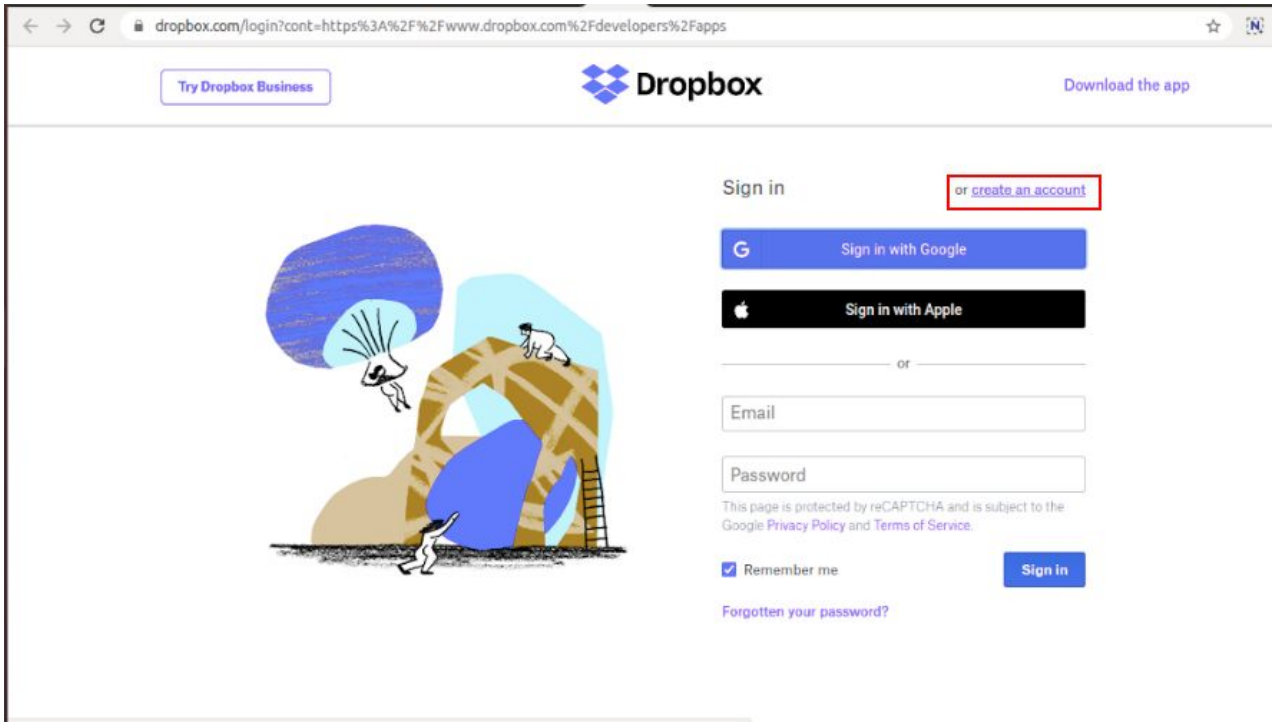
CONTEXT

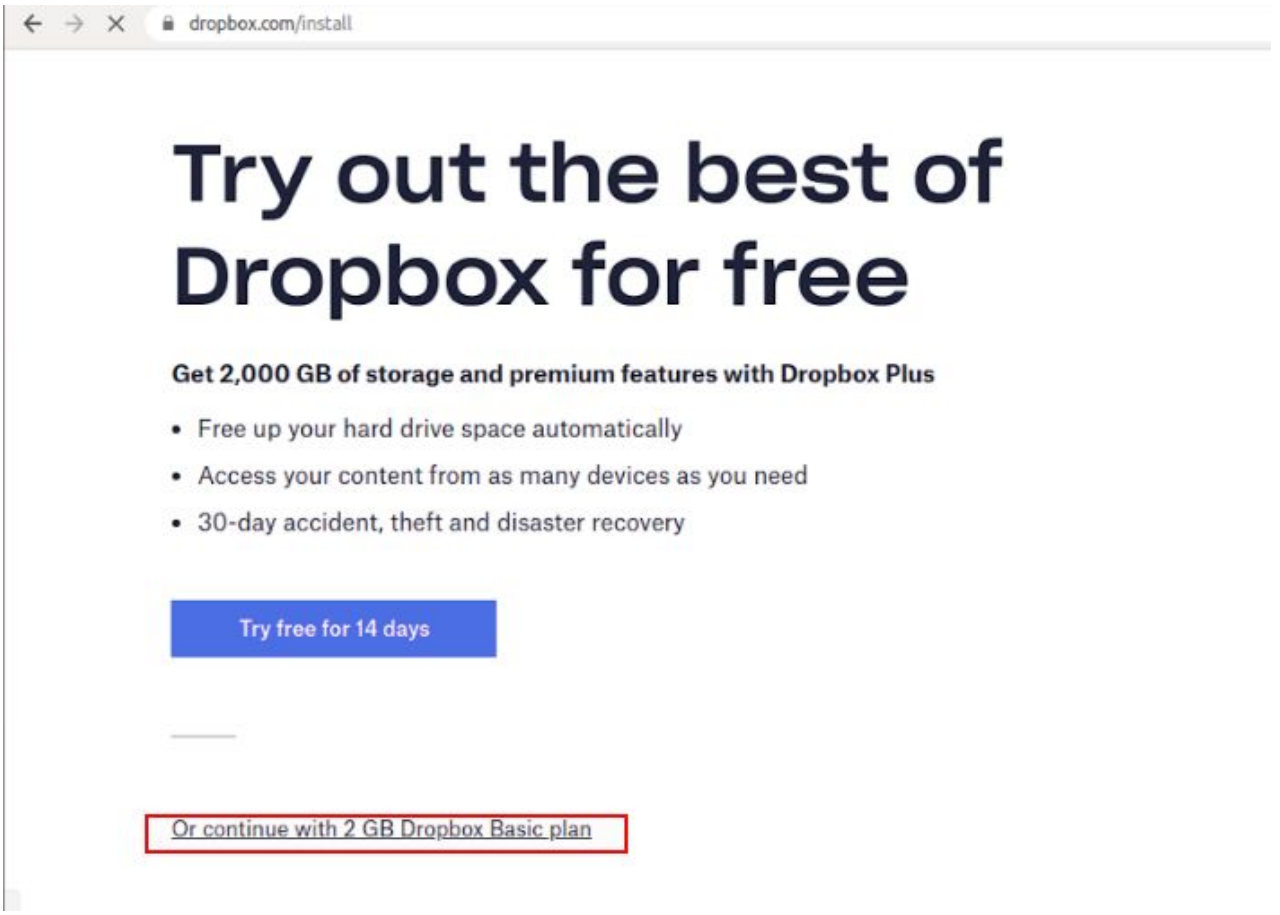
- **Talk about using cloud storage that student have already used like google drive or any other storage**

Class Steps	**Teacher Action**	**Student Action**

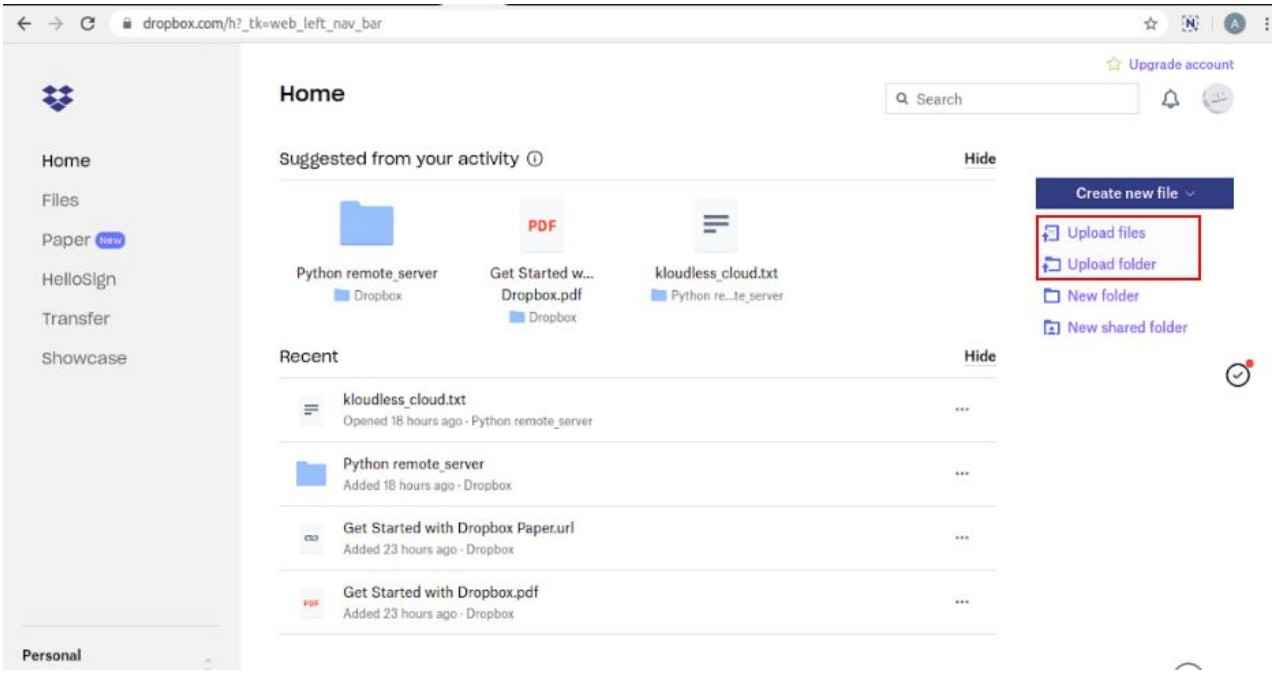
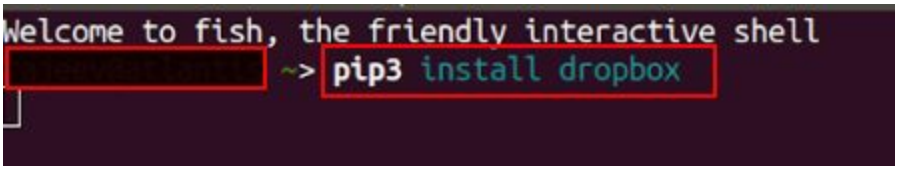
Step 1: **Warm Up** **(5 mins)**	Hi Do you remember what we learned in the last class?	ESR: We learned about functions. We also learned about functions defined for file object and use them to manipulate text inside files.
	Great! Remember the different libraries we used in javascript while designing 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 • **install dropbox library for python and create account on dropbox.** • **Customize python program to take file path as input from users**		
Step 2: Teacher-led Activity (15 min)	Have you ever used a google drive or any other cloud storage before?	ESR:- Google Drive / DropBox
	What is a cloud storage service?	A remote storage space which allows us to store our files remotely. These storage spaces allow us to access data from anywhere and from any device
	If you remember, in the last class we had progressed to create backup for our files. But we were backing up on our own system in a different storage location / drive. Wouldn't it be awesome if we can backup our files to remote cloud services like dropbox and google drive.	ESR: Yes!
	Let us learn to write a python program which backsup any file we want on a cloud storage service/ We will be using a popular cloud storage service called DropBox. Have you used dropbox?	ESR: Yes/No!
	Let's see how we can access the dropbox. <Teacher opens the link from the teacher activity 1> Let's create a account for dropbox.	<Student signup for the dropbox>

	(You can signup with google or fill the form and signup) <teacher chooses one of the options >	
		
	We'll see a screen which will ask to take free trial for 14 days or continue with basic 2GB Dropbox plan. Click on the basic 2GB plan. You have an account created	Student creates a dropBox account



	Now go to the home page of the dropbox. You can also manually upload the files to the dropbox. On the right side of the page you'll see the options to upload files and folder. Click on upload files if you want to upload a file or click on upload folder if you want to upload a folder . When you click on any of those option a box will open from where you can choose your file or folder to upload <Teacher demonstrates by uploading	Student tries by uploading the file on the dropbox manually

	any of the files to the dropbox also helps student do the same>	
		
	DropBox also has a python module / library. We can install it using pip Open terminal / cmd and run: pip3 install dropbox	Student installs dropbox on their system
		
	We had seen how to manually upload the file on DropBox. We will learn how to write a python script which will do that for you. We can also upload files using a python script. In previous class we installed a library/module called howdoi - to	Student observes

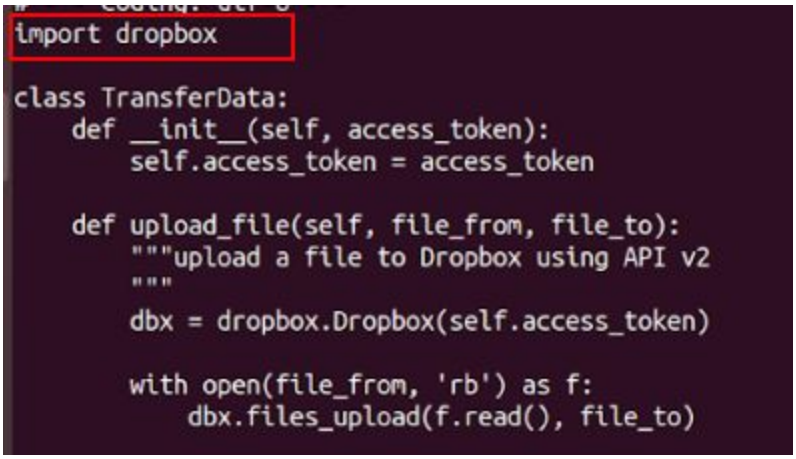
<table>
<tr><td></td><td>which we could ask how do I do anything. Let us see if we can ask it how to upload files using dropbox

<Teacher opens the terminal and write howdoi upload files to dropbox using python.>
You'll see a code to upload file to dropbox. In the code you can see that it needs an access token.
Access token is something by which you can gain access to your cloud storage on dropbox.</td><td></td></tr>
<tr><td colspan="3">

```
~> howdoi upload files to dropbox using python
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import dropbox

class TransferData:
    def __init__(self, access_token):
        self.access_token = access_token

    def upload_file(self, file_from, file_to):
        """upload a file to Dropbox using API v2
        """
        dbx = dropbox.Dropbox(self.access_token)

        with open(file_from, 'rb') as f:
            dbx.files_upload(f.read(), file_to)

def main():
    access_token = '******'
    transferData = TransferData(access_token)

    file_from = 'test.txt'
    file_to = '/test_dropbox/test.txt'  # The full path to upload the file to, including the file name

    # API v2
    transferData.upload_file(file_from, file_to)

if __name__ == '__main__':
    main()
~>
```

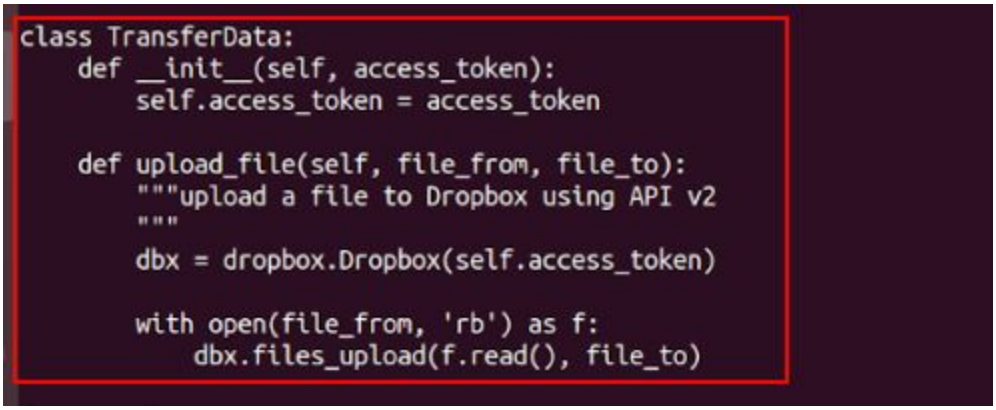
</td></tr>
<tr><td></td><td>Now getting back to the code.
Can you tell me what we are doing first here?

import dropbox</td><td>ESR:
To use dropbox in our code we need to import it first.
so importing dropbox
import dropbox</td></tr>
</table>



```
import dropbox

class TransferData:
    def __init__(self, access_token):
        self.access_token = access_token

    def upload_file(self, file_from, file_to):
        """upload a file to Dropbox using API v2
        """
        dbx = dropbox.Dropbox(self.access_token)

        with open(file_from, 'rb') as f:
            dbx.files_upload(f.read(), file_to)
```

	Very good. What is TransferData here?	ESR: TransferData is a class
	How is the class initialized?	ESR: def __init__(self, access_token): self.access_token = access_token
	Let me explain you what's next.	In this class a constructor (__init__) is used to initialize the object. The object accepts an access token which is passed in the init function
	In the next method dropbox is initialized and stored in variable dbx dbx = dropbox.Dropbox(self.access_token) with open(file_from, 'rb') as f: dbx.files_upload(f.read(), file_to) In this line with statement has been used to open a file as 'f'. with make the code readable and also handles any exception thrown when opening the file. The file is opened in r -> read	ESR: 0 and 1s

	mode and b-> binary mode. You know what is binary right? Yes, all information in computer is stored in binary - 0 and 1. These are called bits. It is easy to transmit information in 0s and 1s. As you can see, we use the dropbox's files_upload method to upload the file to the cloud destination	
 ``` class TransferData: def __init__(self, access_token): self.access_token = access_token def upload_file(self, file_from, file_to): """upload a file to Dropbox using API v2 """ dbx = dropbox.Dropbox(self.access_token) with open(file_from, 'rb') as f: dbx.files_upload(f.read(), file_to) ```		
	What' s this main function doing? def main(): access_token = '******' transferData = TransferData(access_token)	ESR: access_token variable is declared which has some string. Then a new transferData object is created using the class defined earlier and access_token is passed to it.

```
def main():
    access_token = '******'
    transferData = TransferData(access_token)
```

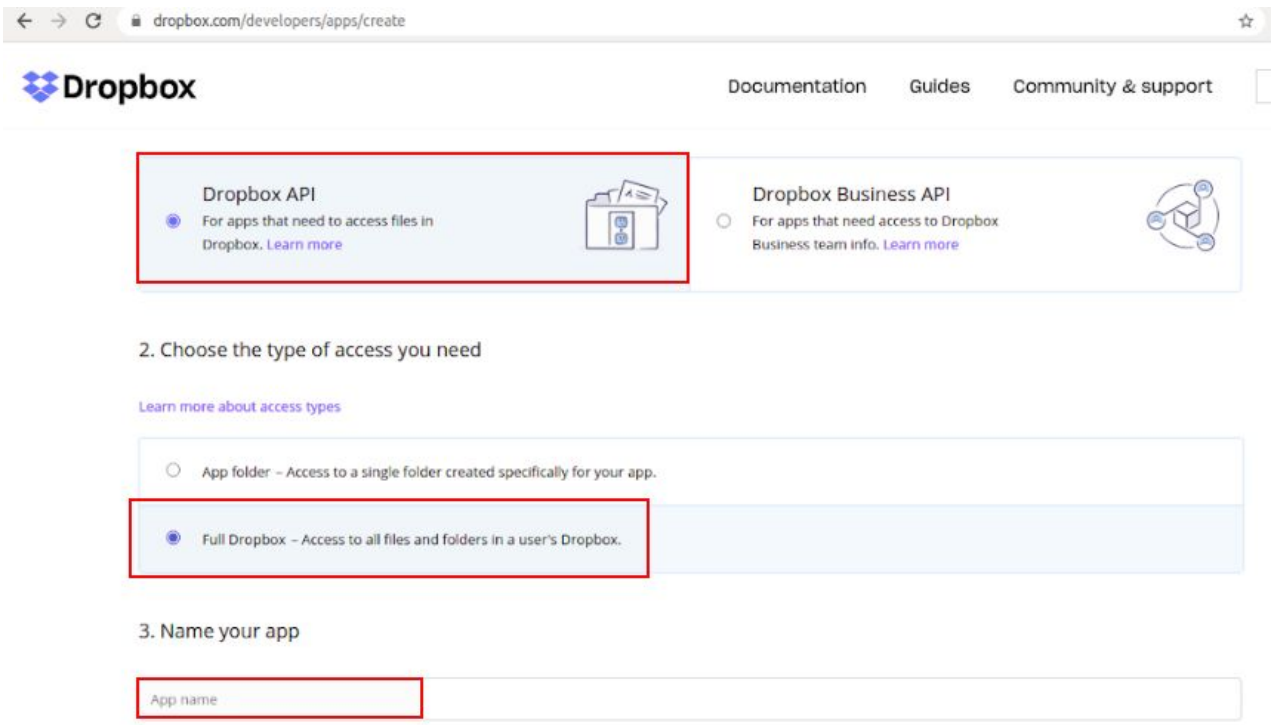
	Ok go on!! What's happening here? code- file_from = 'test.txt' file_to = '/test_dropbox/test.txt' # The full path to upload the file to, including the file name	ESR: After that a variable called file_from is declared which will have the path of the file or folder which we want to upload. below that file_to variable is declared which has the full path to upload the file to, including name that you wish the file to be called once uploaded.

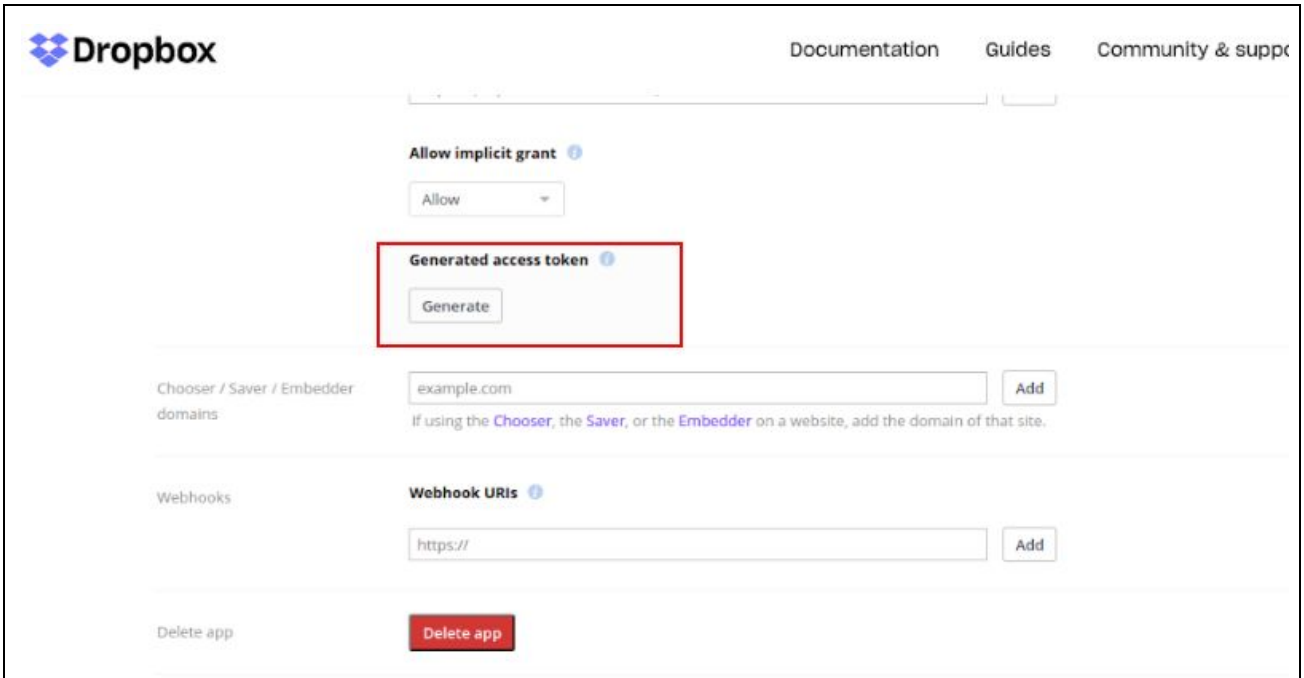
```
def main():
    access_token = '******'
    transferData = TransferData(access_token)

    file_from = 'test.txt'
    file_to = '/test_dropbox/test.txt'  # The full path to upload the file to, including the file name
```

	What's this code doing? code- transferData.upload_file(file_from, file_to)	ESR: Then upload_file function of the class is called and file_from and file_to is passed to it as arguments

```
def main():
    access_token = '******'
    transferData = TransferData(access_token)

    file_from = 'test.txt'
    file_to = '/test_dropbox/test.txt'  # The full path to upload the file to, including the file name

    # API v2
    transferData.upload_file(file_from, file_to)
```

	Do you know what this line means? code- if __name__ == '__main__': main() here the name of the file is set to main and main() is called. If the python interpreter is running that module (the source file) as the main program, it sets the special __name__ variable to have a value “__main__”. If this file is being imported from another module, __name__ will be set to the module’s name. Module’s name is available as value to __name__ global variable.	ESR: varied
		
	We have seen how we can push our files to dropbox. We still need access token so let's get ours To get the access token <teacher opens the link from teacher activity 2 and clicks on create app> Then in choose the Api -click on Dropbox Api option. In the choose the access type -click on Full Dropbox Insert the name of your app and click on create	Student follows the step along with the teacher

	<Teacher helps student to create a app>	
		
	On the settings screen you'll see a generate access token option and a generate button below. click on the generate button and copy the string that appears below. Now we have the access token to use our code. <teacher helps student to generate the token and copy it>	Student generates the token and copies it



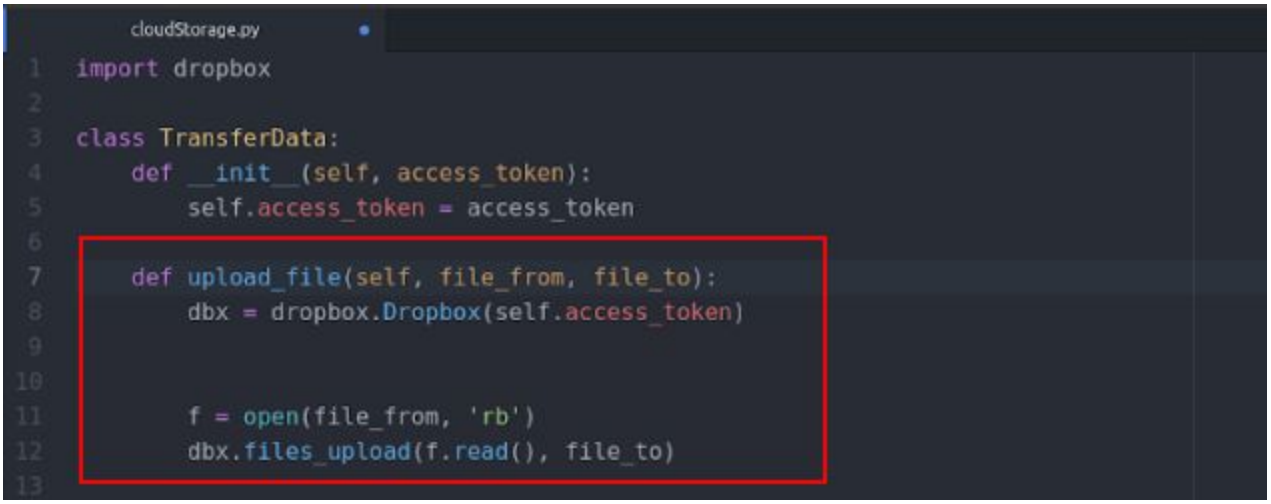
	Ok now we know how to write a script to upload file to dropbox . Let's make some custom changes to the script to accept the file path and the uploading path as input. Can you try doing that?	ESR:- Yes!!
Teacher Stops Screen Share		
	Now it's your turn. Please share your screen with me.	

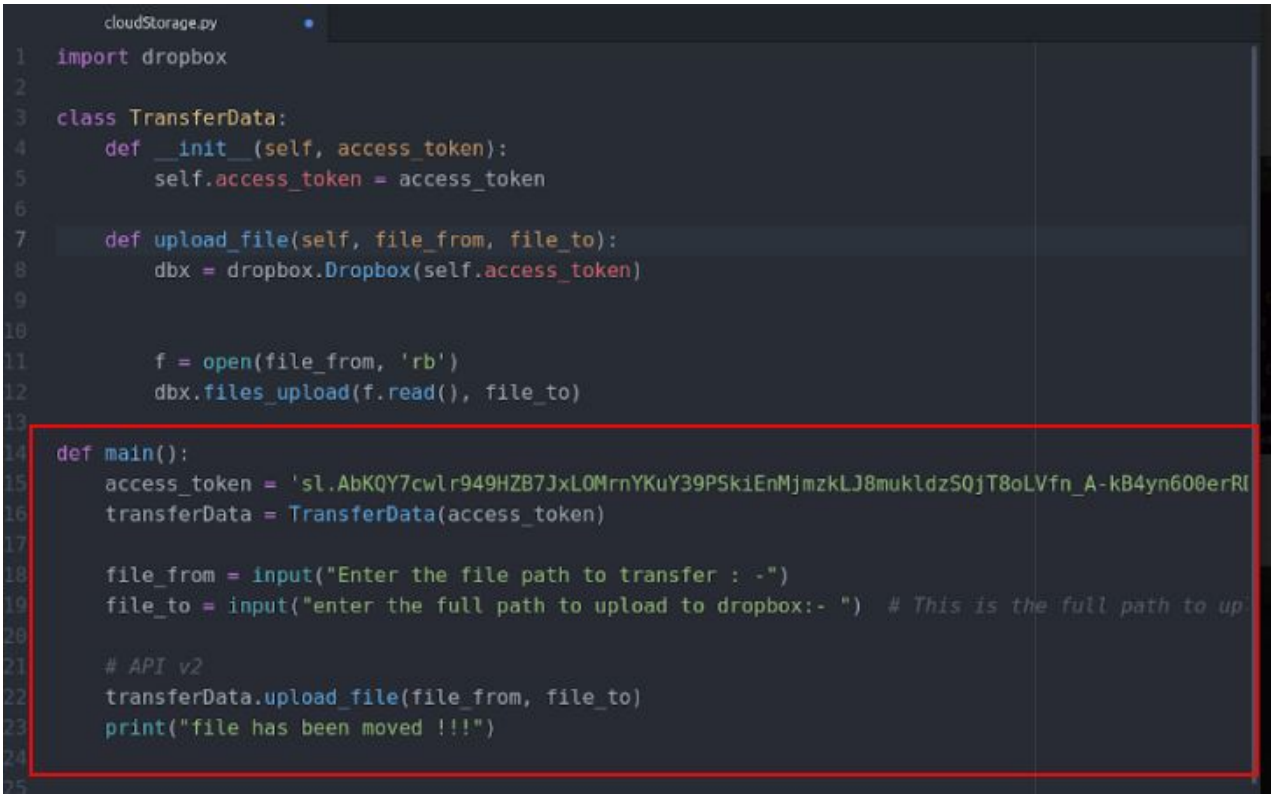
- **Ask Student to press ESC key to come back to panel**
- **Guide Student to start Screen Share**
- **Teacher gets into Fullscreen**

ACTIVITY

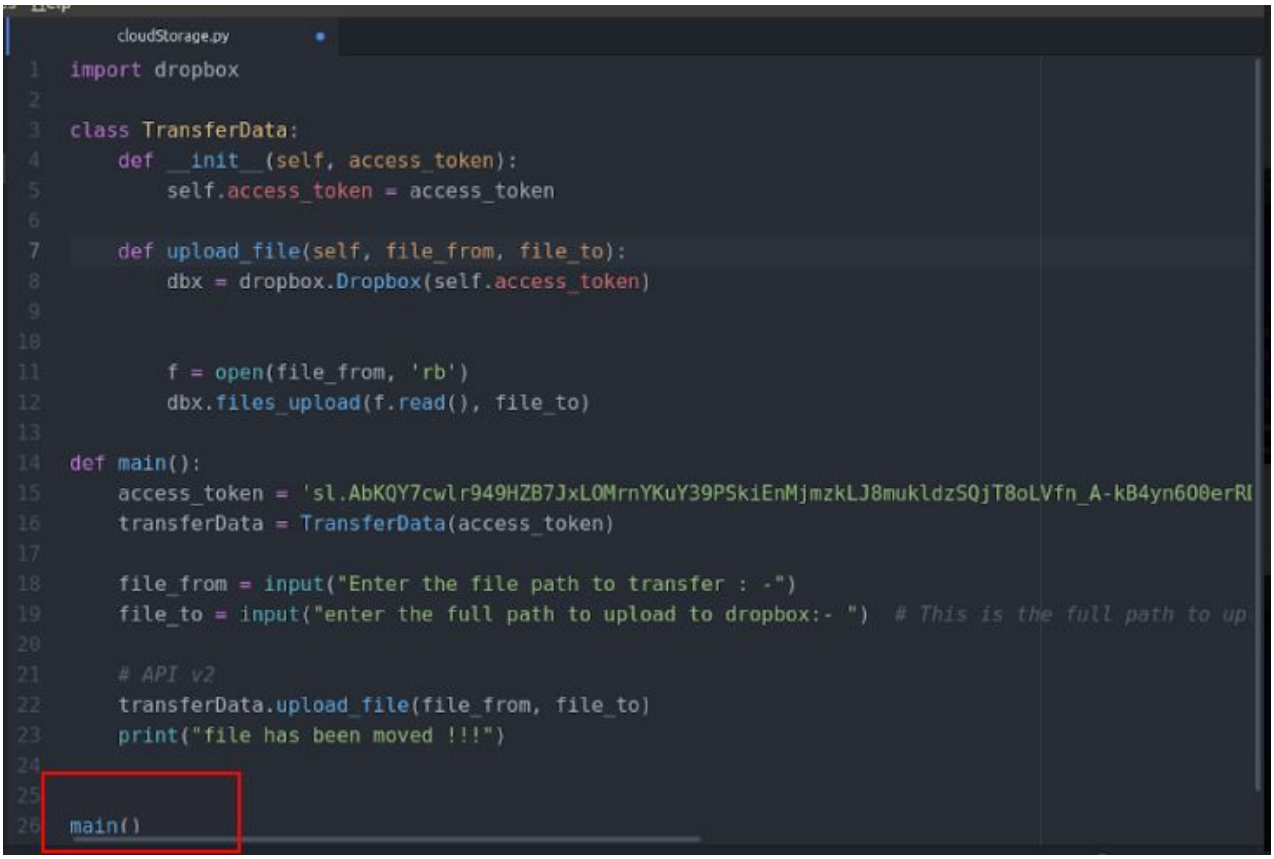
- **Build a program to upload files to dropbox**

<table>
<tr><td>Step 3:
Student-Led Activity
(15 min)</td><td>Let's open our editor and create a new python file with name cloudstorage.py.
<Teacher helps student create a cloudstorage.py file></td><td><student creates a cloudstorage.py file></td></tr>
<tr><td></td><td><Teacher helps importing dropbox></td><td><Student imports the dropbox></td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td></td><td><Teacher helps student to define class TransferData and using constructor to create a object which takes access token></td><td><Student defines a class TransferData and uses constructor to create a object which takes access token></td></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

	<Teacher helps student to define upload_file method in class which initializes dropbox , reads the file as binary and then upload it to dropbox>	< Student defines a upload_file method in class which initializes dropbox , reads the file as binary and then upload it to dropbox>
		
	<teacher helps student to define a main function which:- - has a access token stored in access_token variable. - creates a new object transferData using the class TransferData and passes access token to it. - Takes the file name as input from the user in command line and stores in variable file_from. -takes the path to upload the files on dropbox as input from user and store it in variable file_to. - calls the upload file method of the TransferData class and pass file_from and file_to it.	< student defines a main function which:- - has a access token stored in access_token variable. - creates a new object transferData using the class TransferData and passes access token to it. - Takes the file name as input from the user in command line and stores in variable file_from. -takes the path to upload the files on dropbox as input from user and store it in variable file_to. - calls the upload file

		method of the TransferData class and pass file_from and file_to it.
		
	<teacher helps student to call the main function and run and test the code and check if the file is getting uploaded on the dropbox or not>	<Student calls the main function and run and tests the code and checks if the file is getting uploaded on the dropbox>

```
cloudStorage.py
import dropbox

class TransferData:
    def __init__(self, access_token):
        self.access_token = access_token

    def upload_file(self, file_from, file_to):
        dbx = dropbox.Dropbox(self.access_token)


        f = open(file_from, 'rb')
        dbx.files_upload(f.read(), file_to)

def main():
    access_token = 'sl.AbKQY7cwlr949HZB7JxLOMrnYKuY39PSkiEnMjmzkLJ8mukldzSQjT8oLVfn_A-kB4yn6O0erR
    transferData = TransferData(access_token)

    file_from = input("Enter the file path to transfer : -")
    file_to = input("enter the full path to upload to dropbox:- ")  # This is the full path to up

    # API v2
    transferData.upload_file(file_from, file_to)
    print("file has been moved !!!")
```



cloudStorage.py

```
import dropbox

class TransferData:
    def __init__(self, access_token):
        self.access_token = access_token

    def upload_file(self, file_from, file_to):
        dbx = dropbox.Dropbox(self.access_token)


        f = open(file_from, 'rb')
        dbx.files_upload(f.read(), file_to)

def main():
    access_token = 'sl.AbKQY7cwlr949HZB7JxLOMrnYKuY39PSkiEnMjmzkLJ8mukldzSQjT8oLVfn_A-kB4yn6O0erRI
    transferData = TransferData(access_token)

    file_from = input("Enter the file path to transfer : -")
    file_to = input("enter the full path to upload to dropbox:- ")  # This is the full path to up

    # API v2
    transferData.upload_file(file_from, file_to)
    print("file has been moved !!!")


main()
```

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 dropbox module to upload files on cloud. We wrote a script which takes input file path from the user and makes a backup on the cloud storage.

	Awesome. Next class, we will be designing a project where you will create a security script which takes a snapshot of anyone who is using your laptop and uploads it on cloud!	-
Teacher Clicks 		
Additional Activities	Encourage the student to write reflection notes in their reflection journal using markdown. 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?	The student uses the markdown editor to write her/his reflection in a reflection journal.

Activity	Activity Name	Links
Teacher Activity 1	Dropbox login	https://www.dropbox.com/login?cont=https%3A%2F%2Fwww.dropbox.com%2Fh
Teacher Activity 2	access token	https://www.dropbox.com/developers/apps

Teacher Activity 3	final solution	https://github.com/whitehatjr/cloud_storage
Student Activity 1	Dropbox login	https://www.dropbox.com/login?cont=https%3A%2F%2Fwww.dropbox.com%2Fh
Student Activity 2	access token	https://www.dropbox.com/developers/apps