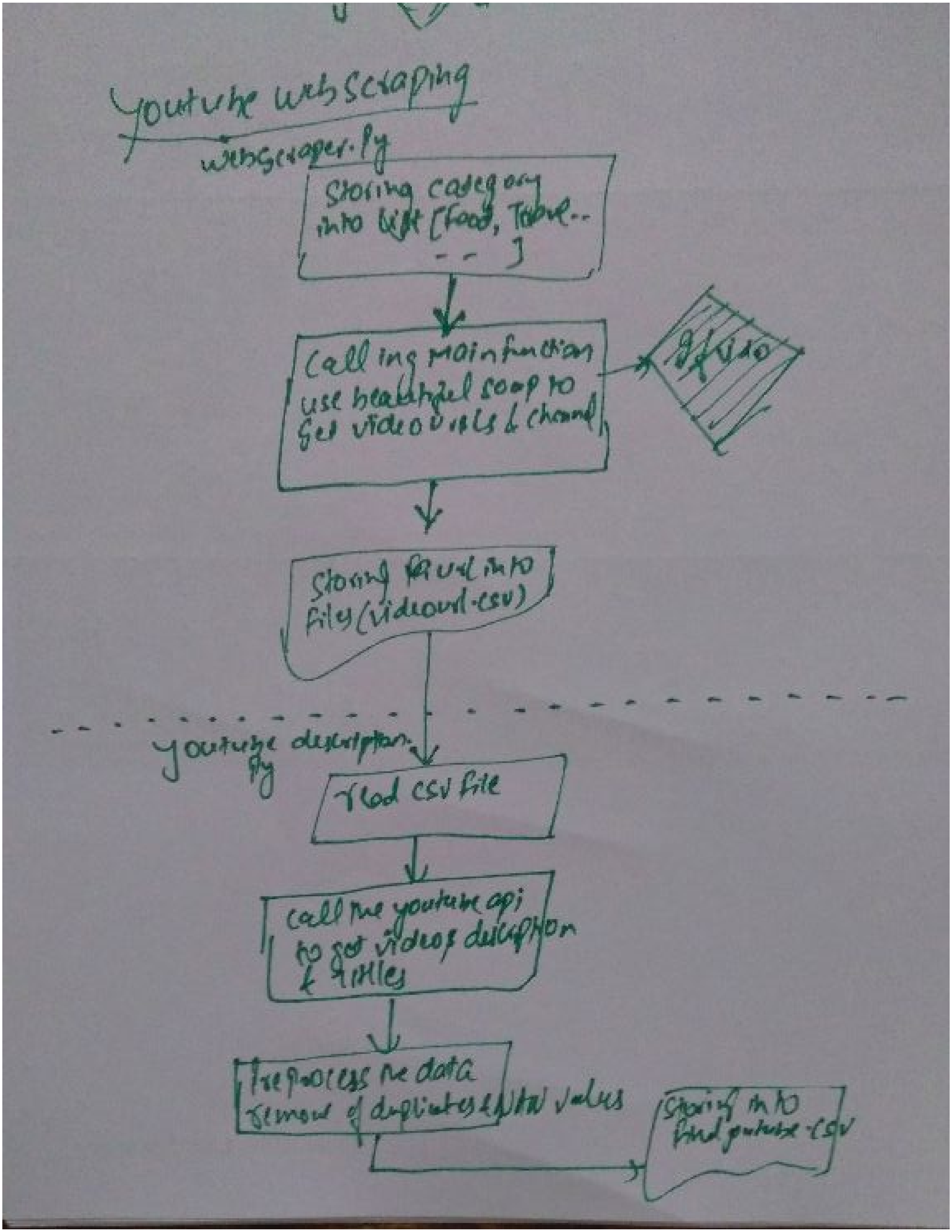
Youtube Webscraper

Youtube webscraper cantain 2 python file

1-WebScraper.py(For getting urls)

2-youtubeDescription.py(For getting title and descriptions)

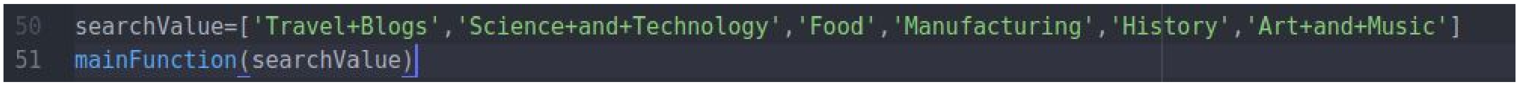

youtube webscraping
webscraper.py
Storing category into list [Food, Travel.. - -]
Calling mainfunction use beautiful soup to get video urls & channel
df video
Storing the url into file(videourl.csv)
youtube description.py
read csv file
call the youtube api to get videos description & titles
Preprocess the data removal of duplicates & NAN values
Storing into final youtube.csv

1.WebScraper.py

Webscraper contain one function mainFunction()

If we search Food on youtube we will get video,playlist and channel related to foods.so we have to get url of videos and channel names.

We create a list searchValue and store all categories and call the mainFunction()



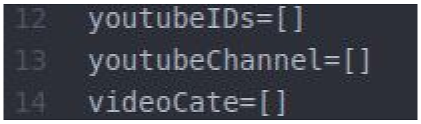
```
searchValue=['Travel+Blogs','Science+and+Technology','Food','Manufacturing','History','Art+and+Music']
mainFunction(searchValue)
```

Initilization lists

youtubeIDs is used to store video ids

youtubeChannel is used to store channel name

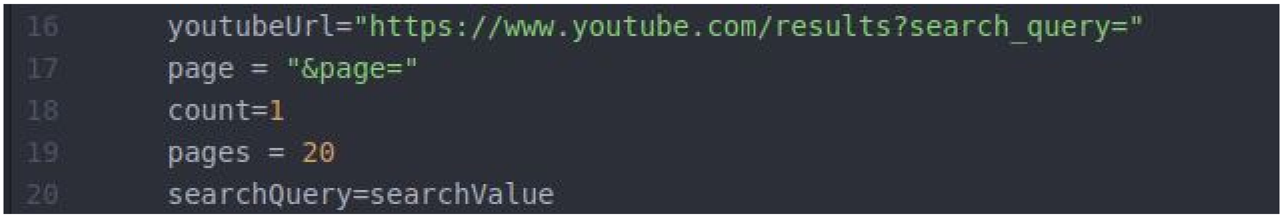
videoCate is used to store categories



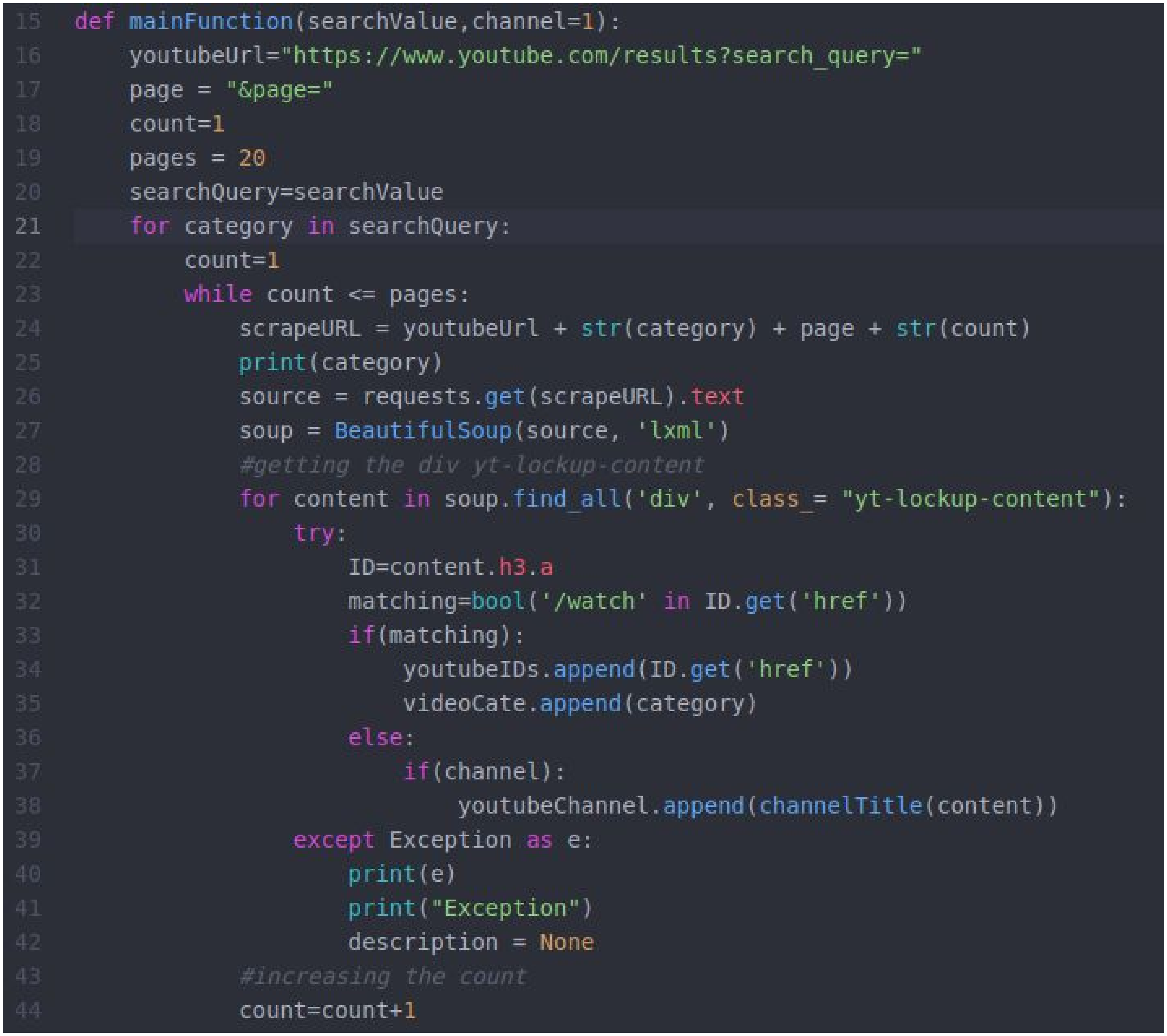
```
youtubeIDs=[]
youtubeChannel=[]
videoCate=[]
```

Inside mainfunction()

Initializing the youtubeurl,page and count(starting page) and pages(end page)



```
    youtubeUrl="https://www.youtube.com/results?search_query="
    page = "&page="
    count=1
    pages = 20
    searchQuery=searchValue
```



```python
def mainFunction(searchValue,channel=1):
    youtubeUrl="https://www.youtube.com/results?search_query="
    page = "&page="
    count=1
    pages = 20
    searchQuery=searchValue
    for category in searchQuery:
        count=1
        while count <= pages:
            scrapeURL = youtubeUrl + str(category) + page + str(count)
            print(category)
            source = requests.get(scrapeURL).text
            soup = BeautifulSoup(source, 'lxml')
            #getting the div yt-lockup-content
            for content in soup.find_all('div', class_ = "yt-lockup-content"):
                try:
                    ID=content.h3.a
                    matching=bool('/watch' in ID.get('href'))
                    if(matching):
                        youtubeIDs.append(ID.get('href'))
                        videoCate.append(category)
                    else:
                        if(channel):
                            youtubeChannel.append(channelTitle(content))
                except Exception as e:
                    print(e)
                    print("Exception")
                    description = None
            #increasing the count
            count=count+1
```

For every category in list we search on youtube using beautiful soap.then find all the div tag containing yt-lockup-content and store ID in hyper reference

If Id is video then

1-Video link store in YoutubeIDs list

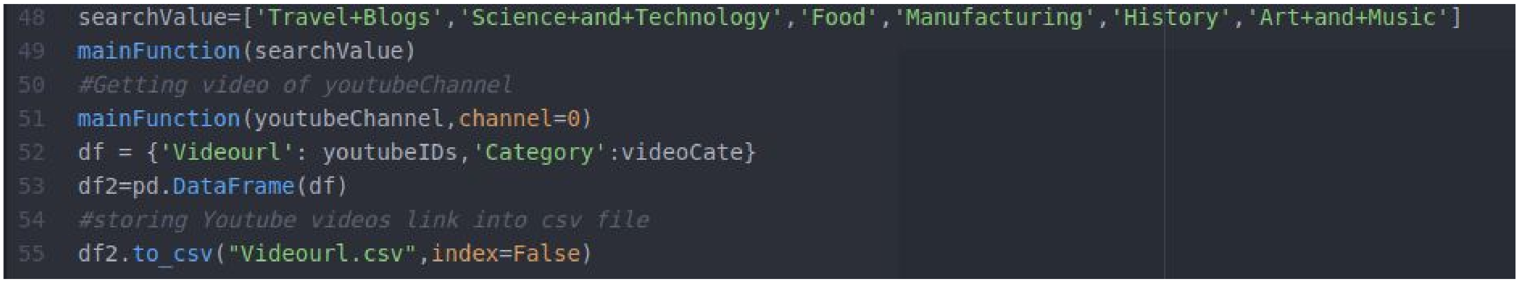
2-videoCate store category related to video

Else

Call the function channelTitle

Store the channel name in youtubeChannel list

After mainFunction finish then we again call mainFunction(youtubeChannel) to get video from youtube channel



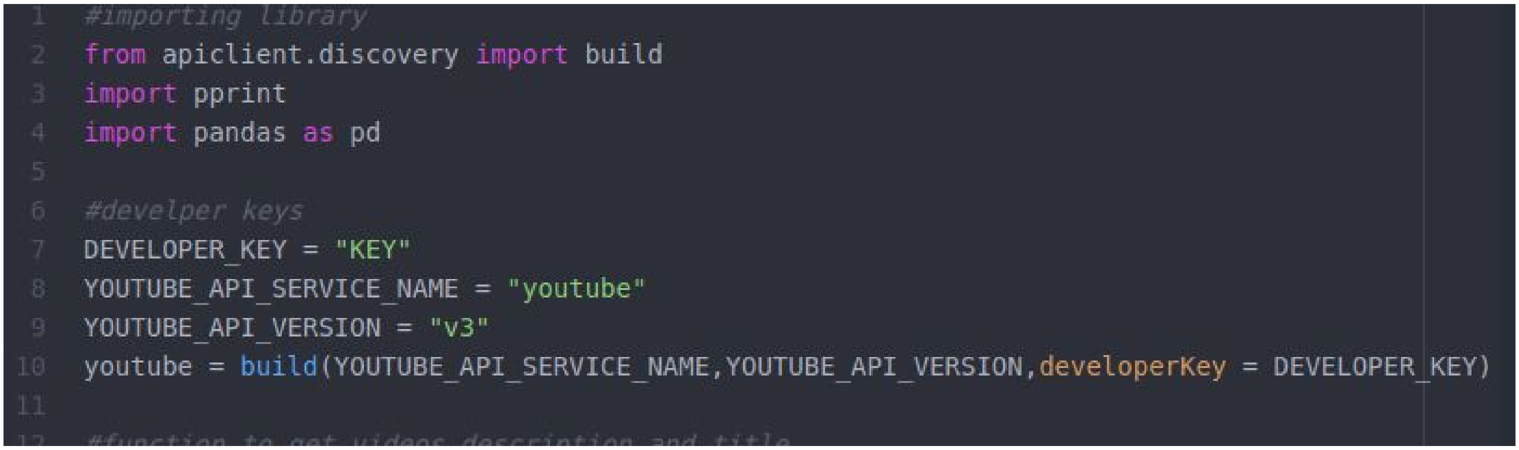
```python
searchValue=['Travel+Blogs','Science+and+Technology','Food','Manufacturing','History','Art+and+Music']
mainFunction(searchValue)
#Getting video of youtubeChannel
mainFunction(youtubeChannel,channel=0)
df = {'Videourl': youtubeIDs,'Category':videoCate}
df2=pd.DataFrame(df)
#storing Youtube videos link into csv file
df2.to_csv("Videourl.csv",index=False)
```

Storing the dataFrame into videourl.csv

2.youtubeDescription.py

Main function of of youtubeDescription is to get video youtubeDescription and title.i use youtubeapi to get description of videos

Initilized DEVELOPER_KEY



```python
#importing library
from apiclient.discovery import build
import pprint
import pandas as pd

#develper keys
DEVELOPER_KEY = "KEY"
YOUTUBE_API_SERVICE_NAME = "youtube"
YOUTUBE_API_VERSION = "v3"
youtube = build(YOUTUBE_API_SERVICE_NAME,YOUTUBE_API_VERSION,developerKey = DEVELOPER_KEY)

#function to get videos description and title
```

Reading the VideoUrl.csv file which contain videos url

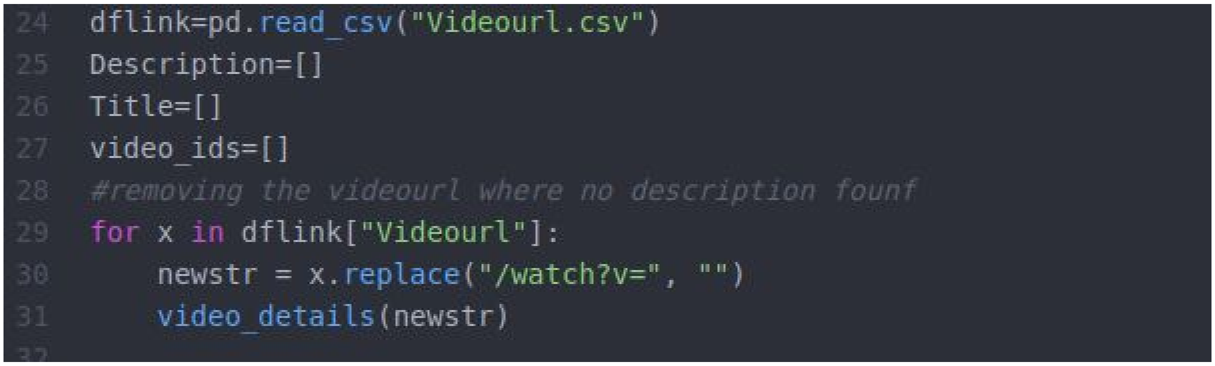
Initilization lists

Description is used to store video description

Title is used to store video titles

video_ids store video is where there is no description

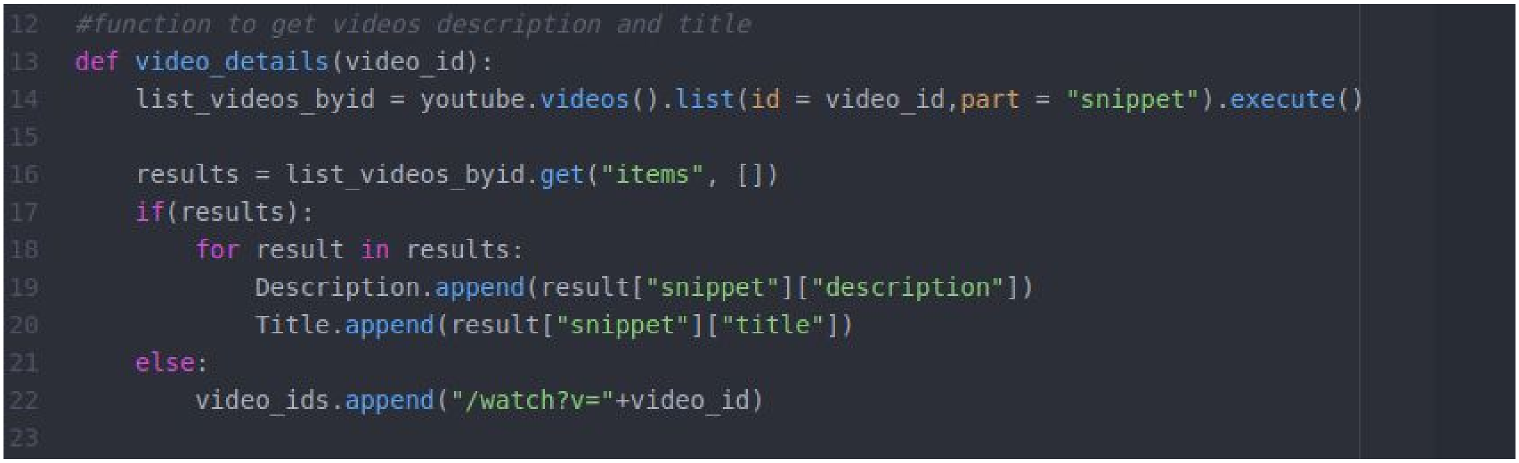
For very videoIds we call video_details function



```python
dflink=pd.read_csv("Videourl.csv")
Description=[]
Title=[]
video_ids=[]
#removing the videourl where no description founf
for x in dflink["Videourl"]:
    newstr = x.replace("/watch?v=", "")
    video_details(newstr)
```

video_detail call youtubeapi and get result of every video url

If result not found store the video url into video_ids list

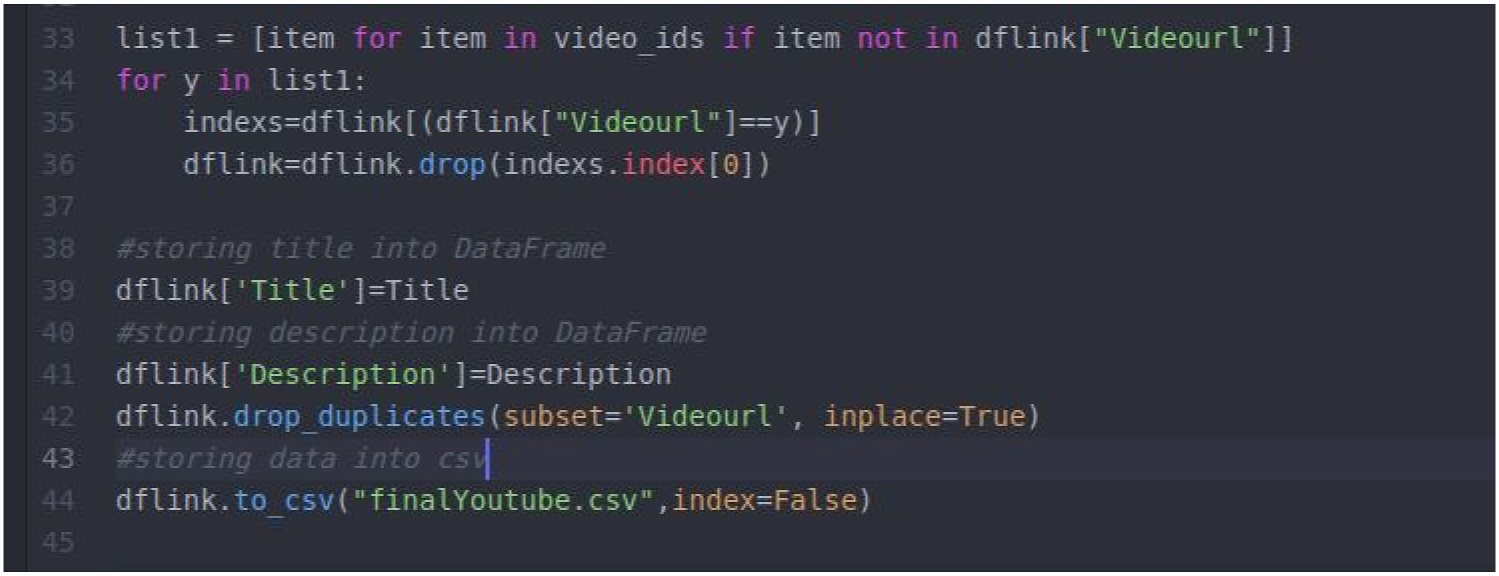


```python
#function to get videos description and title
def video_details(video_id):
    list_videos_byid = youtube.videos().list(id = video_id,part = "snippet").execute()

    results = list_videos_byid.get("items", [])
    if(results):
        for result in results:
            Description.append(result["snippet"]["description"])
            Title.append(result["snippet"]["title"])
    else:
        video_ids.append("/watch?v="+video_id)

```

Removing all the url where there is no description and storing the data frame into FinalYoutube.csv



```python
list1 = [item for item in video_ids if item not in dflink["Videourl"]]
for y in list1:
    indexs=dflink[(dflink["Videourl"]==y)]
    dflink=dflink.drop(indexs.index[0])

#storing title into DataFrame
dflink['Title']=Title
#storing description into DataFrame
dflink['Description']=Description
dflink.drop_duplicates(subset='Videourl', inplace=True)
#storing data into csv
dflink.to_csv("finalYoutube.csv",index=False)

```