Project name:

YouTube Video Downloader

Submitted by:

Irfan Lateef

Submitted to:

1: Sir Farrukh Arslan

2: Sir Khalid Waheed

Team Members:

1: Irfan Lateef (Leader)

2: Muhammad Dawood (Member)

3: Muhammad Abdullah (Member)

4: Mohsin Abbas (Member)

Team:

Bolean Autocrats

YouTube Video Downloader

This is a complete guide of this project. In this manual we'll provide a complete documentation of our project named "YouTube Video Downloader"

Completed codes are available on Github. Please check our Github account ArfsnSandhu

Or you can follow along with this manual to create your own.

This project is actually based on two parts in first part we'll create the GUI (Graphical User Interface) and in Second part we'll create functions (As many as we need) for the execution of our required purposes.

Note:

1. In this documentation all the codes will be highlighted with yellow color.
2. All the references are at the end of the document.
3. Code with the black background is from the editor for better understanding.

Installing Pytube

First of all we have to install the pytube library on our system. For downloading the pytube in our system we'll simply execute the following command in our terminal.

```
Pip install pytube
```

After that the terminal will show you the successful message if that happens than you can continue.

Now we are ready to go.

Creating GUI

So, first we'll create GUI. For creating GUI we need to import the tkinter module.

So, we'll write these three lines of codes to get the required modules from the library.

```
from tkinter import *
```

```
From tkinter import ttk
```

```
from tkinter import filedialog
```

```
from tkinter import *
from tkinter import ttk
from tkinter import filedialog
```

First of all we have to create a window with the following command.

```
root = Tk()
```

```
root = Tk()
```

Then we need to set a title for the window by following command.

root.title(“YouTube Video Downloader”)

```
root.title("YouTube Video Downloader")
```

Then we need to set the window size and we can easily set that by following command.

root.geometry(“700x500”) # You can set width and height of your choice

```
root.geometry("700x500")  # We are setting windows size.
```

And the following code is optional. The following code means we are adjusting the text into center and it’s totally your choice that you want text in center or not.

root.columnconfigure(0, weight=1)

```
root.columnconfigure(0, weight=1)  # Setting all contents in center
```

Alright we have done the basics now our work is easy and strait forward. Actually I mean that you just have to repeat the same line of codes to get your desire look.

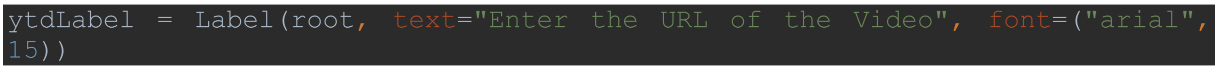
So, First of all lets learn that how to create a Label. Label is a built in command in tkinter to create a text label. All you need is just to give some parameters like your text, font size, color and etc.

So, you just need to write a code like below.

name (This could be any name as it is a variable name) = Label(root, text=”Your text you want to display”, font=(“font name”, font size), fg=”your color name” bg=”your color name”)

In our case our first heading code will look like this.

ytdlabel = Label(root, text=”Enter the URL of the Video”, font=(“Arial”, 15))



```
ytdLabel = Label(root, text="Enter the URL of the Video", font=("arial",
15))
```

But this text won’t be displayed in your window as you just have store that values into a variable. For displaying that text in your window you need to run the following command.

ytdlabel.grid() # It mean that we are displaying that variable into grid.

```
ytdLabel.grid()
```

So, with this simple method you can create as many labels as you want.

Alright, Now we have to create a text field to get the url from the user. Let’s learn how to create an entry field with tkinter and believe me this is as easy as writing single line of code.

So, you just need to write a code like below.

name (This could be any name as it is a variable name) = Entry(root, width=desire, textvariable=your variable name for storing the value entered by user.)

In our case our first Entry code will look like this.

ytdentry = Entry(root, width=50, textvariable=ytdenrtyvalue)

```
ytdentry = Entry(root, width=50, textvariable=ytdentervalue)
```

Again this won’t show up in window until you didn’t execute that variable into grid by following command.

Ytdentry.grid() # It mean that we are displaying that variable into grid.

In our project we need some buttons to do some actions. So, let's learn how to create a button with tkinter. Actually that is similar with creating label you just have to write Button instead of Lable

So, you just need to write a code like below.

name (This could be any name as it is a variable name) = Button(root, width=desire with in numbers, text="Your text you want to display", font=("font name", font size), fg="your color name", bg="your color name")

In our case our first button (named "Choose path") code will look like this.

saveentry = Button(root, width=10, bg="red", fg="white", text="Choose path")

```
saveentry = Button(root, width=10, bg="red", fg="white", text="Choose path",
command=openlocation)
```

for now leave that command parameter, believe me we'll cover that later.

And I want to say that there is always something to learn. For that I will post the links for the references at the end of the documents so that you can learn more.

Alright we are just one step away from creating the GUI. Lastly we just need to create an option menu so that our user could select whether he want to download Audio or Video or resolution of video.

In tkinter an option menu is called **Combobox.** So, you just need to write a code like below.

name (This could be any name as it is a variable name) = ttk.Combobox(root, values=choices)

In our case our Combobox code will look like this.

ytdchoices = ttk.Combobox(root, values=choices)

```
choices = ["Lowest available quality", "Highest available quality", "Only
Audio"]
ytdchoices = ttk.Combobox(root, values=choices)
ytdchoices.grid()
```

Again this won't show up in window until you didn't execute that variable into grid by following command.

ytdchoices.grid() # It mean that we are displaying that variable into grid.

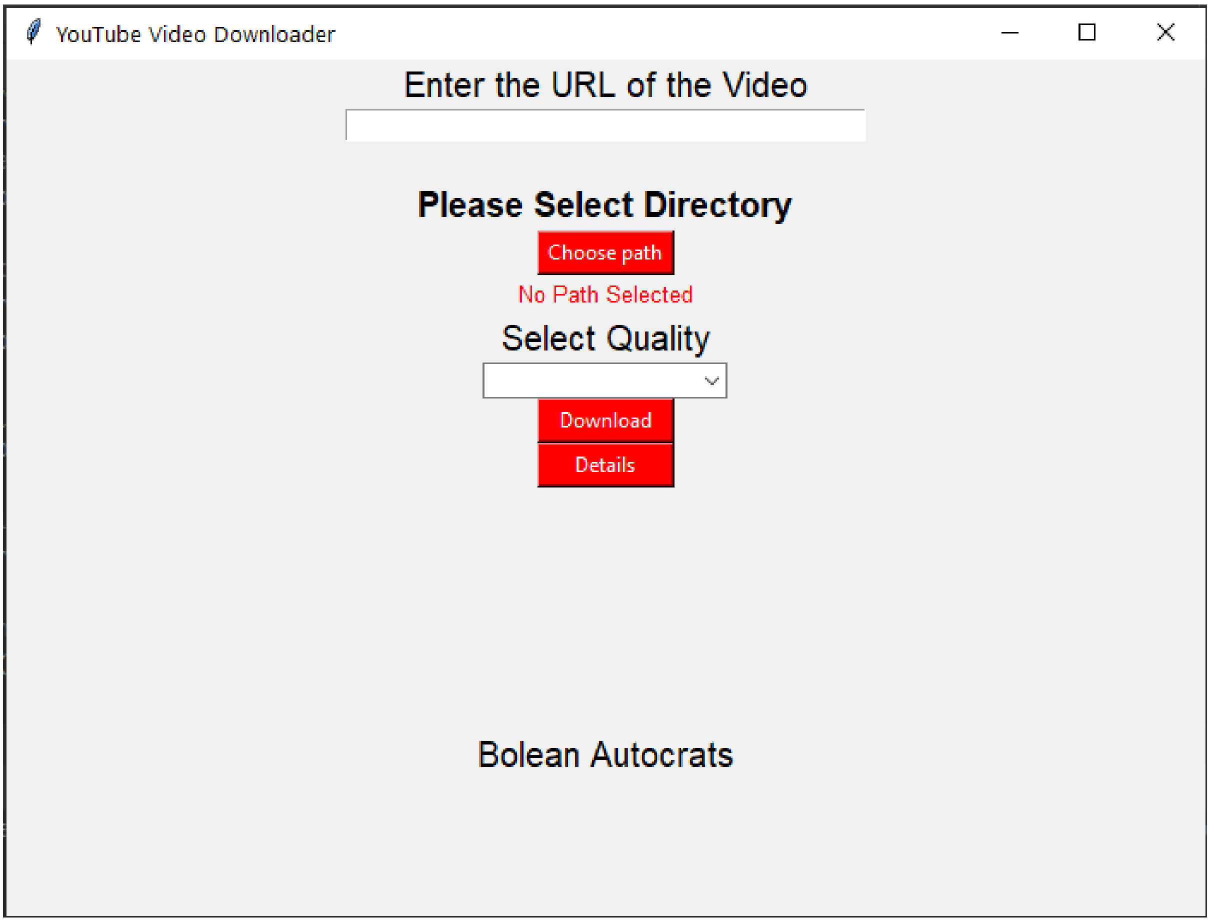
There is a picture after creating the GUI with following code.

Code for GUI of this project:

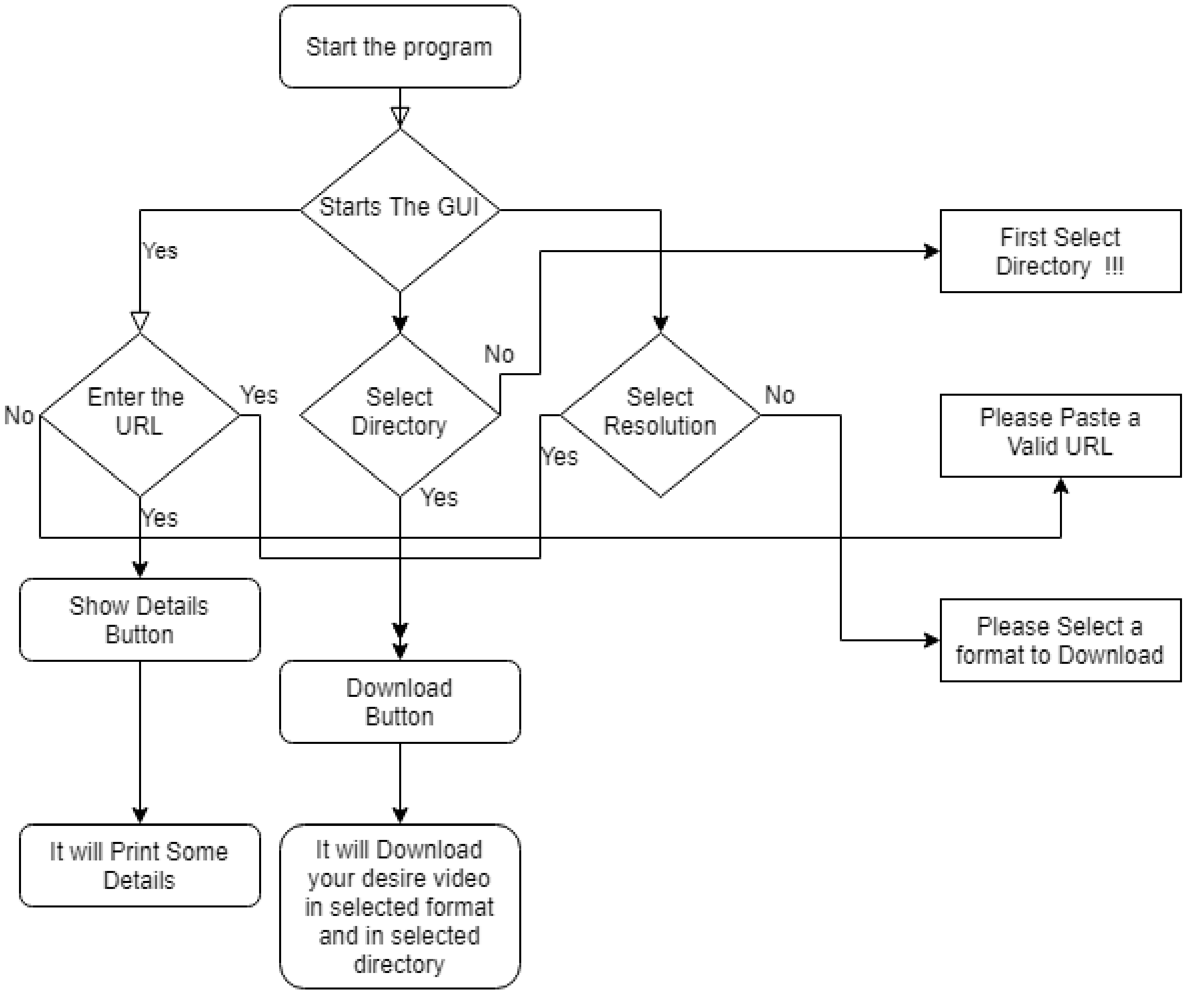
```
# From here the GUI (Graphical User Interface) starts.

root = Tk()
root.title("YouTube Video Downloader")
root.geometry("700x500")  # We are setting windows size.
root.columnconfigure(0, weight=1)  # Setting all contents in center
# Setting Link Label
ytdLabel = Label(root, text="Enter the URL of the Video", font=("arial",
15))
ytdLabel.grid()
# Setting the input value box
ytdentervalue = StringVar()
ytdentry = Entry(root, width=50, textvariable=ytdentervalue)
ytdentry.grid()
#Setting the error message
ytderror = Label(root, text="", fg="red", font=("arial", 10))
ytderror.grid()
# Asking for file location Label
savelabel = Label(root, text="Please Select Directory", font=("arial", 15,
"bold"))
savelabel.grid()
# Creating a button for user to select directory
saveentry = Button(root, width=10, bg="red", fg="white", text="Choose
path", command=openlocation)
saveentry.grid()
#Setting the error message for location
ytderrorlocation = Label(root, text="No Path Selected", fg="red",
font=("arial", 10))
ytderrorlocation.grid()
# Setting label for asking download quality
ytdquality = Label(root, text="Select Quality", font=("arial", 15))
ytdquality.grid()
# Giving user an option to select resolution
choices = ["Lowest available quality", "Highest available quality", "Only
Audio"]
ytdchoices = ttk.Combobox(root, values=choices)
ytdchoices.grid()
# Creating a button for user to download video
downloadbtn = Button(root, width=10, bg="red", fg="white", text="Download",
command=downloadvideo)
downloadbtn.grid()

# Creating a button for user to See video Details
Detailbtn = Button(root, width=10, bg="red", fg="white", text="Details",
command=details)
Detailbtn.grid()
# Displaying the Video Details
# Title heading
titleheading = Label(root, text="", font=("arial", 12, "bold"))
titleheading.grid()
# Displaying Title
titlelabel = Label(root, text="", font=("arial", 10))
titlelabel.grid()
# Duration Heading
durationheading = Label(root, text="", font=("arial", 12, "bold"))
durationheading.grid()
# Displaying Duration
```

```
durationlabel = Label(root, text="", font=("arial", 10))
durationlabel.grid()
# author Heading
authorheading = Label(root, text="", font=("arial", 12, "bold"))
authorheading.grid()
# Displaying author
authorlabel = Label(root, text="", font=("arial", 10))
authorlabel.grid()


# Developer Label
developerlabel = Label(root, text="Bolean Autocrats", font=("roboto", 15))
developerlabel.grid()


root.mainloop()
```

Result of above code:



Flow Chart of this project.


Start the program
Starts The GUI
Yes
Enter the URL
No
Yes
Yes
Select Directory
No
Yes
Select Resolution
No
Yes
First Select Directory !!!
Please Paste a Valid URL
Please Select a format to Download
Show Details Button
Download Button
It will Print Some Details
It will Download your desire video in selected format and in selected directory

Writing Functions

In second part we’ll create some functions to download video (and for other actions). But actually we’ll write these function first. There is only one purpose for writing functions first is beauty of the code. As a good programmer is always tries to make a program readable for other one. Enough talk let’s come to the work.

Alright first of all I want to tell you that in this project we’ll write 3 function one for asking file directory from user, the second one is for downloading video and the third one is for displaying some important information about that video.

Function for getting and handling file location.

First we’ll declare a global variable for storing location path later.

Folder_name = “”

```
Folder_name = ""
```

Now we’ll define a function and write the code like below.

def openlocation():

    global Folder_name

    Folder_name = filedialog.askdirectory() # This will show the user a dialog box so that user can select the directory visualy.

Now we’ll write the if statement to configure the text path error to the exact text path. So, that user will have no doubt that either he had select the directory or not.

    if(len(Folder_name) > 1):

        your_variable_name.config(text=Folder_name, fg=”your color name’)

    else:

        your_variable_name.config(text=”your error message”, fg=”your color name’)

And in our case the code will look like this.

```
def openlocation():
    global Folder_name
    Folder_name = filedialog.askdirectory()
    if(len(Folder_name) > 1):
        ytderrorlocation.config(text=Folder_name, fg="green")
    else:
        ytderrorlocation.config(text="First Select Directory !!!",
fg="red")
```

Function to download video

Alright first of all we’ll define a function

def downloadvideo():

```
def downloadvideo():
```

Do you remember that above we have stored a list of choices in a variable called **ytdchoices.** so, now we’ll get that values and store them into a new variable called **choice**.

choice = ytdchoices.get()

```
choice = ytdchoices.get()
```

And now we’ll get the value from the entry field and store that value into a variable called **url.**

url = ytdentry.get()

```
url = ytdentry.get()
```

Now we have to write two if statements (Nested if statements). In our first if statement we’ll confirm that either our user has entered the url or not, if yes then continue to further procedure if not then show an error message.

```
If(len(url) > 1):
        ytd = YouTube(url)
```

Our nested if statements will go here that we have explained below.

```
else:
        ytderror.config(text=”Please paste valid link”, fg=”red”)
```

```
if(len(url) > 1):
    ytderror.config(text="")
    yt=YouTube(url)
    # Other if statements will go here thst we have explained below.
else:
    ytderror.config(text="Please paste valid  lnk", fg="red")
```

Nested if statement

So, as you know that we have created a list of three choices. So, we have to write an if statement inside our first if statement with two elif statements.

In this scenario first I will show you the code so that you will get an idia of what we are going to do with these three choices.

And I will only share the code from the editor as this is very straight forward and easy.

```
if(choice == choices[0]):
    select = yt.streams.filter(progressive=True,
file_extension='mp4').first()
    select.download(Folder_name)
    ytderror.config(text="Download Completed")
elif(choice == choices[1]):
    select = yt.streams.filter(progressive=True,
file_extension='mp4').last()
    select.download(Folder_name)
    ytderror.config(text="Download Completed")
elif(choice == choices[2]):
    select = yt.streams.filter(only_audio=True).first()
    select.download(Folder_name)
    ytderror.config(text="Download Completed")
else:
    ytderror.config(text="Please paste link again or restart the app",
fg="red")
```

Okay our only big question is what does the below code means.

```
select = yt.streams.filter(progressive=True, file_extension='mp4').first()
```

As you know that we have declared a variable called **yt** at the very first of writing this function. And we have stored the url from the user into a variable called **url** and then we have use the pytube library for getting the information about that video for further use. So, that information has saved in a variable called **yt**.

Now we just have to write codes like below that what we want to fetch about that video, For example.

If we want to fetch the stream we'll write yt.streams

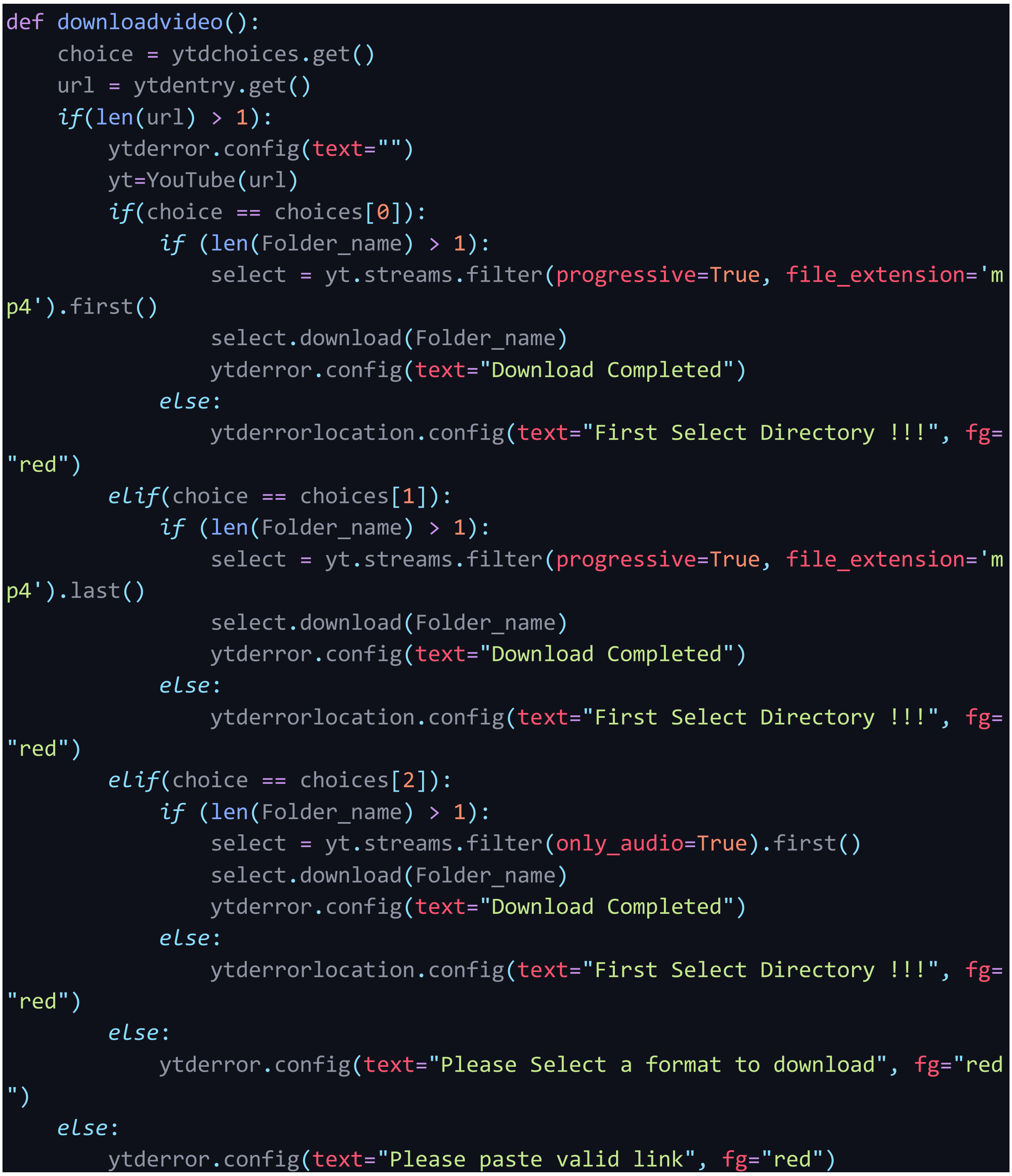
If we want to fetch the title name we'll write yt.title

If we want to fetch the description we'll write yt.description

And so on.

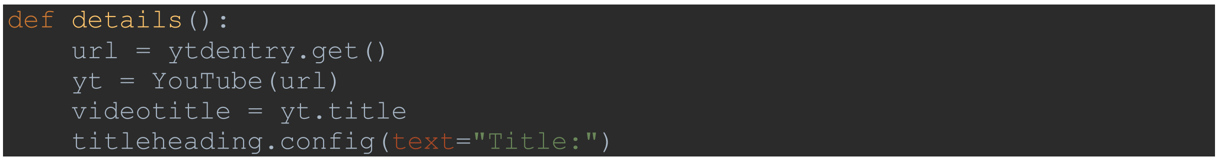
Again if you want to learn more then please check the references at the end of this document.

So, there is the final code of this function



```
def downloadvideo():
    choice = ytdchoices.get()
    url = ytdentry.get()
    if(len(url) > 1):
        ytderror.config(text="")
        yt=YouTube(url)
        if(choice == choices[0]):
            if (len(Folder_name) > 1):
                select = yt.streams.filter(progressive=True, file_extension='mp4').first()
                select.download(Folder_name)
                ytderror.config(text="Download Completed")
            else:
                ytderrorlocation.config(text="First Select Directory !!!", fg="red")
        elif(choice == choices[1]):
            if (len(Folder_name) > 1):
                select = yt.streams.filter(progressive=True, file_extension='mp4').last()
                select.download(Folder_name)
                ytderror.config(text="Download Completed")
            else:
                ytderrorlocation.config(text="First Select Directory !!!", fg="red")
        elif(choice == choices[2]):
            if (len(Folder_name) > 1):
                select = yt.streams.filter(only_audio=True).first()
                select.download(Folder_name)
                ytderror.config(text="Download Completed")
            else:
                ytderrorlocation.config(text="First Select Directory !!!", fg="red")
        else:
            ytderror.config(text="Please Select a format to download", fg="red")
    else:
        ytderror.config(text="Please paste valid link", fg="red")
```

Okay this is our final function and I am not going to explain it as I have explained every thing about this short code in the above 8 pages. And our function is following.



```
def details():
    url = ytdentry.get()
    yt = YouTube(url)
    videotitle = yt.title
    titleheading.config(text="Title:")
    titlelabel.config(text=videotitle)
    if(yt.length <= 60):
        videolength = yt.length
        finalvideolength = str(videolength) + " Sec"
    else:
        videolength = round(yt.length/60, 0)
        finalvideolength = str(videolength) + " Min"
    durationheading.config(text="Duration:")
    durationlabel.config(text=finalvideolength)
    videoauthor = yt.author
    authorheading.config(text="Author:")
    authorlabel.config(text=videoauthor)
```

And our project has been completed.

Here I will share some references about these topics so that you can learn more.

Oooh wait, wait, wait, and almost forgot to tell you something.

Do you remember this line of code?

```
 saveentry = Button(root, width=10, bg="red", fg="white", text="Choose path", command=openlocation)
```

So, as you know that we have defined our functions but we all know that functions won't run until we call them. And in our case we want to call our functions on a button click. So, in that specific button we'll give the parameter of command and then we will write the name of that function we want to execute (Without parentheses).

Command = name of that function we want to execute (Without parentheses).

Now we can say that we have done anything.

Regards.

References

An open source for learning python:

https://www.w3schools.com/python/default.asp

Learn tkinter complete:

https://docs.python.org/3/library/tk.html

Learn basics of tkinter:

https://www.geeksforgeeks.org/python-gui-tkinter/

Link for our GitHub account for project file:

https://github.com/ArfsnSandhu/youtube-video-downloader