**Download-YouTube-videos-using-Python**

Hi All,

This doc is made with the reference of JournalDev, Stack Overflow and Python.org.

**Step 1:** Install pytube or pytube3 library using pip command.

pip install pytube

Some system may get error while installing pytube lib:

>>> from pytube import YouTube

Traceback (most recent call last):

  File "<stdin>", line 1, in <module>

  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/pytube/\_\_init\_\_.py", line 16, in <module>

    from pytube.streams import Stream

  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/pytube/streams.py", line 17, in <module>

    from pytube import extract

  File "/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/pytube/extract.py", line 7, in <module>

    from pytube.compat import quote

ImportError: cannot import name 'quote' from 'pytube.compat' (/Library/Frameworks/Python.framework/Versions/3.7/lib/python3.7/site-packages/pytube/compat.py)

>>>

So, I would recommend you to use pytube3 library. I am using the same in this tutorial.

**Step 2:** The first step is to import the YouTube class from the pytube module.

|  |
| --- |
|  |
| from pytube import YouTube  **Step 3:** Next step is to create the YouTube object by passing the YouTube video URL.  youtube\_video\_url = 'https://www.youtube.com/watch?v=\_kUrW9SEaJc' |

yt\_obj = YouTube(youtube\_video\_url)

(P.S: Don’t try to change the variable name)

**Step 4:** The YouTube object opens different streams from the YouTube video URL. We can get all the stream information using the following code.

for stream in yt\_obj.streams:

print(stream)

It will produce the following output:

<Stream: itag="18" mime\_type="video/mp4" res="360p" fps="30fps" vcodec="avc1.42001E" acodec="mp4a.40.2" progressive="True" type="video">

<Stream: itag="22" mime\_type="video/mp4" res="720p" fps="30fps" vcodec="avc1.64001F" acodec="mp4a.40.2" progressive="True" type="video">

<Stream: itag="137" mime\_type="video/mp4" res="1080p" fps="30fps" vcodec="avc1.64001e" progressive="False" type="video">

<Stream: itag="248" mime\_type="video/webm" res="1080p" fps="30fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="299" mime\_type="video/mp4" res="1080p" fps="60fps" vcodec="avc1.64002a" progressive="False" type="video">

<Stream: itag="303" mime\_type="video/webm" res="1080p" fps="60fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="136" mime\_type="video/mp4" res="720p" fps="30fps" vcodec="avc1.4d4016" progressive="False" type="video">

<Stream: itag="247" mime\_type="video/webm" res="720p" fps="30fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="298" mime\_type="video/mp4" res="720p" fps="60fps" vcodec="avc1.4d4016" progressive="False" type="video">

<Stream: itag="302" mime\_type="video/webm" res="720p" fps="60fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="135" mime\_type="video/mp4" res="480p" fps="30fps" vcodec="avc1.4d4014" progressive="False" type="video">

<Stream: itag="244" mime\_type="video/webm" res="480p" fps="30fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="134" mime\_type="video/mp4" res="360p" fps="30fps" vcodec="avc1.4d401e" progressive="False" type="video">

<Stream: itag="243" mime\_type="video/webm" res="360p" fps="30fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="133" mime\_type="video/mp4" res="240p" fps="30fps" vcodec="avc1.4d400c" progressive="False" type="video">

<Stream: itag="242" mime\_type="video/webm" res="240p" fps="30fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="160" mime\_type="video/mp4" res="144p" fps="30fps" vcodec="avc1.4d400b" progressive="False" type="video">

<Stream: itag="278" mime\_type="video/webm" res="144p" fps="30fps" vcodec="vp9" progressive="False" type="video">

<Stream: itag="140" mime\_type="audio/mp4" abr="128kbps" acodec="mp4a.40.2" progressive="False" type="audio">

<Stream: itag="251" mime\_type="audio/webm" abr="160kbps" acodec="opus" progressive="False" type="audio">

**Step 5:** The produced output is different types of codecs. In simple language, these are types of download. What exactly you want, refer:

* Progressive: Stream contains the file having both audio and video.
* Adaptive: stream contains either audio or video.
* mime\_type, res, fps : Attributes can be used to filter the stream that we want to download.

**Step 6:** We can use the filter() function to extract only specific streams.

filters = yt\_obj.streams.filter(progressive=True, file\_extension='mp4')

for mp4\_filter in filters:

    print(mp4\_filter)

**Step 7:** The final step is to call the download() method on the specific stream to download the YouTube video.

filters.get\_highest\_resolution().download()

***And you’re done!!! You’ll get your video where the python is installed in your system.***

There are few useful functions to get the highest and lowest resolution videos:

filters.get\_highest\_resolution()

filters.get\_lowest\_resolution()

You can also try this:

**Complete Code to Download the Highest Resolution YouTube video**

from pytube import YouTube

youtube\_video\_url = 'https://www.youtube.com/watch?v=DkU9WFj8sYo'

try:

    yt\_obj = YouTube(youtube\_video\_url)

    filters = yt\_obj.streams.filter(progressive=True, file\_extension='mp4')

    # download the highest quality video

    filters.get\_highest\_resolution().download()

    print('Video Downloaded Successfully')

except Exception as e:

    print(e)

-----------------------------------------------------------------------------

Author : Vallabh Walse

Github : <https://github.com/Vallabh18/>