pytube使用方法

# pytube下载地址及介绍说明

<https://pypi.python.org/pypi/pytube/#downloads>

GitHub 上的：<https://github.com/nficano/pytube>

安装方式1：利用pip命令Download using pip via pypi.（推荐）

pip install pytube

安装方式2：下载whl包

pip install pytube\*\*\*.whl

# pytube介绍

pytube is a lightweight, dependency-free Python library (and command-line utility) **for downloading YouTube Videos**.

YouTube is the most popular video-sharing platform in the world and as a hacker you may encounter a situation where you want to script something to download videos. For this I present to you pytube.

**pytube is a lightweight library written in Python**. **It has no third party dependencies and aims to be highly reliable.**

pytube also makes pipelining easy, allowing you to specify callback functions for different download events, such as **on progress or on complete**.

Finally pytube also **includes a command-line utility**, allowing you to quickly download videos right from terminal.

# pytube的特征Features

* Support for Both Progressive & DASH Streams
* Easily Register **on\_download\_progress & on\_download\_complete** callbacks
* Command-line Interfaced Included
* Caption Track Support
* Outputs Caption Tracks to .srt format (SubRip Subtitle)
* Ability to Capture Thumbnail URL.
* Extensively Documented Source Code
* No Third-Party Dependencies

# 示例：

下载一个视频：Shakira - Waka Waka (This Time for Africa) (The Official 2010 FIFA World Cup™ Song)

Let’s begin with showing **how easy it is to download a video with pytube**:（一句就可以搞定一个视频）

>>>from pytube import YouTube

**>>>YouTube('https://www.youtube.com/watch?v=pRpeEdMmmQ0').streams.first().download()**

This example will download **the highest quality progressive** download stream available.**下载分辨率最高的视频。**

# Getting Strarted 详细介绍

## 导入pytube模块中的YouTube

from pytube import YouTube

## 根据视频链接创建一个YouTube对象

yt = **YouTube**('https://www.youtube.com/watch?v=pRpeEdMmmQ0')

## 查看所有的分辨率格式视频、音频

**Next, let’s explore how we would view what video streams are available:**

**>>> YouTube**('https://www.youtube.com/watch?v=pRpeEdMmmQ0').streams.**all()**

OR

>>>yt = **YouTube**('https://www.youtube.com/watch?v=pRpeEdMmmQ0')

>>>yt.streams.all()



You may notice that some streams listed have **both a video codec and audio codec**, while others have **just video or just audio**, this is a result of YouTube supporting a streaming technique called **Dynamic Adaptive Streaming over HTTP (DASH**).

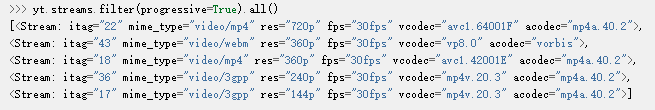
In the context of pytube, the implications are **for the highest quality streams**; you now need to download both the audio and video tracks and then post-process them with software like FFmpeg to merge them.

The legacy streams that contain the audio and video in a single file (referred to as “progressive download”) are still available, but only for resolutions 720p and below.

## 过滤器filter()

### progressive=True

>>> yt.streams.filter(progressive=True).all()



### adaptive=True

Conversely, if you only want to see the DASH streams (also referred to as “adaptive”) you can do:

>>> yt.streams.filter(adaptive=True).all()

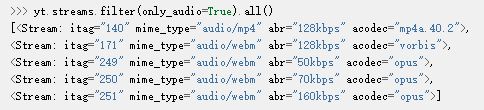


Pytube allows you to filter on every property available (see the documentation for the complete list), let’s take a look at some of the most useful ones.

### only\_audio=True 只下载音频

To list the audio only streams:

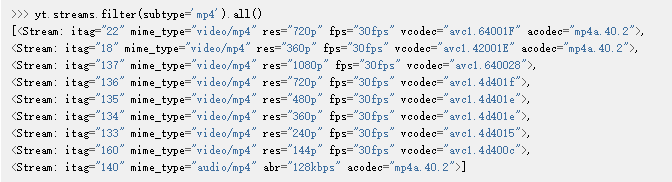
>>> yt.streams.filter(only\_audio=True).all()



### subtype='mp4' 只下载mp4视频

To list only mp4 streams:

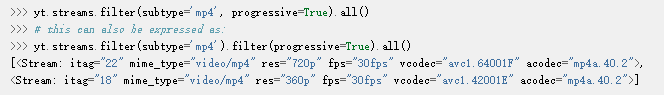
>>> yt.streams.filter(subtype='mp4').all()



### 可以一次多次过滤

Multiple filters can also be specified:

>>> yt.streams.filter(**subtype='mp4', progressive=True**).all()



## 通过标签tag筛选

You also have an interface to select streams by their itag, without needing to filter:

>>> yt.streams.**get\_by\_itag**(22)



## 选取特定特征的视频：如高分辨率

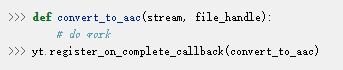
If you need to **optimize for a specific feature, such as the “highest resolution” or “lowest average bitrate”**:

>>> yt.streams.**filter(progressive=True).order\_by('resolution').desc()**.all()

注意：Note that**order\_by** cannot be used if your attribute is undefined in any of the Stream instances, so be sure to apply a filter to remove those before calling it.先过滤再排序。

## “on download complete” callback function：下载完成返回函数

If your application requires **post-processing logic**, pytube allows you to specify an “on download complete” callback function:



## 下载进度条

Similarly, if your application requires **on-download progress logic**, pytube exposes a callback for this as well:



# 命令行工具（直接在命令行下）

pytube also ships with a tiny cli interface for downloading and probing videos.

Let’s start with downloading:

>>> pytube http://youtube.com/watch?v=9bZkp7q19f0 --itag=22

To view available streams:

>>> pytube <http://youtube.com/watch?v=9bZkp7q19f0> --**list**





Finally, if you’re filing a bug report, the cli contains a switch called --build-playback-report, which bundles up the state, allowing others to easily replay your issue.