***系級：****四電機一乙* ***學號：****B10907128* ***姓名：****王璽鑄*

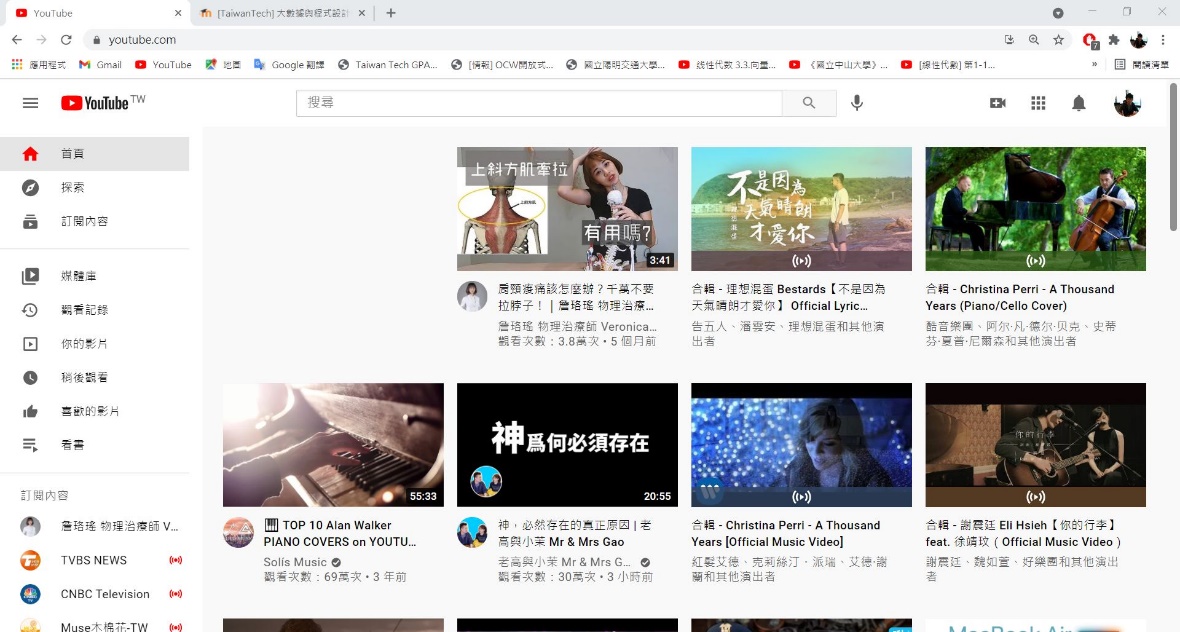
**期末Python程式：YouTube影音擷取爬蟲程式**

### 一、設計此程式的動機

當初設計程式的構想是想要抓取PTT上的文章，後來又去爬了其他的網站，最後選擇了YouTube來進行爬蟲。程式的主軸在於爬取YouTube上的影音檔案來做使用，同時從網址獲取YouTube網頁的重要資訊(影片標題、影片評價、影片長度、影片觀賞次數)，並回傳進度條提醒下載進度。

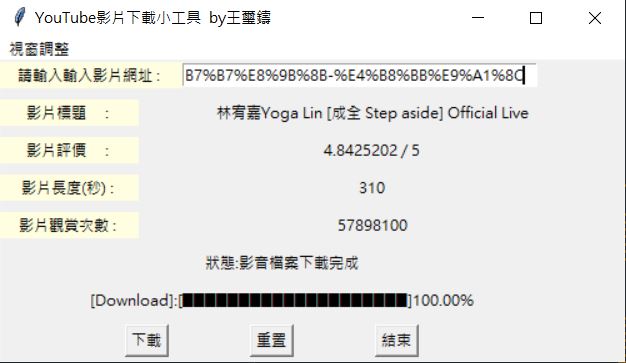
### 二、程式設計目標

撰寫Python程式，自動擷取YouTube網頁上的影音檔案，網址可以任意輸入，只要是YouTube的影音網頁，其網站如圖所示。

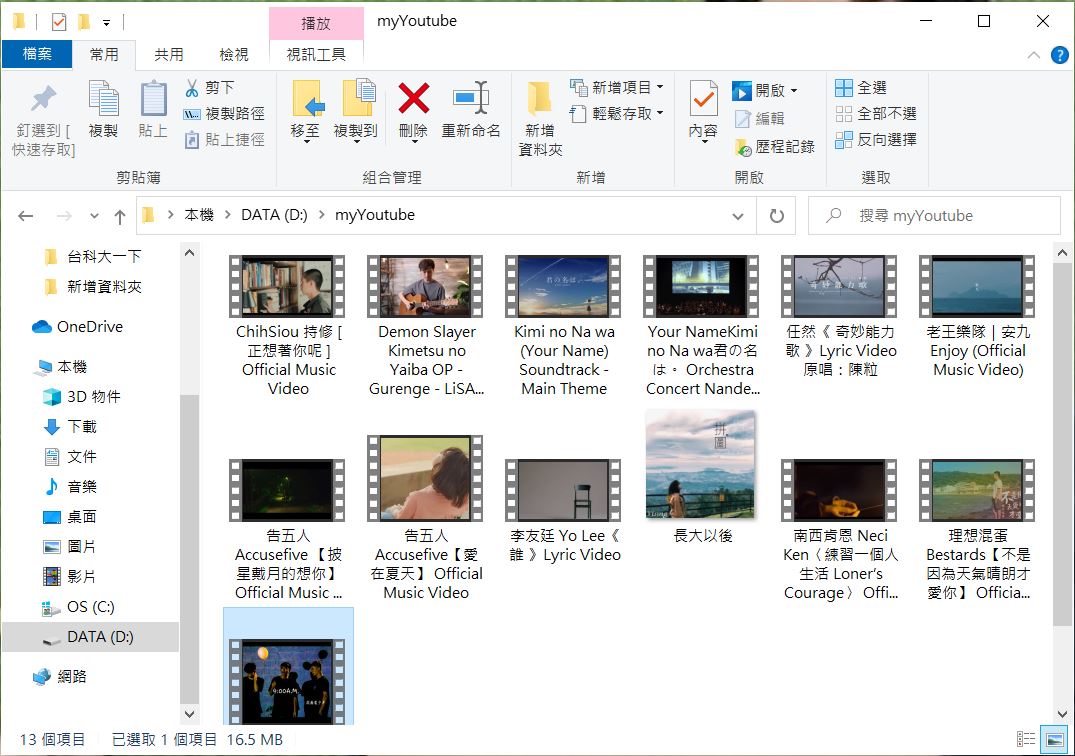


執行程式的時候，會先請求輸入網址，執行軟體後，會抓取網頁原始碼上資料(影片標題、影片評價、影片長度、影片觀賞次數)及進行影音檔下載，結束後並且會印出圖檔下載的狀況，如圖所示。





程式會自動在D槽路徑，建立資料夾「myYoutube」，並將網頁擷取下來的影音檔置於其中，其下載結果如圖所示。



### 三、本程式所使用的套件功能介紹

本程式主要運用4個套件，其功能說明如下：

* **pytube套件**

我們在安裝整合開發環境Anaconda套件時，通常要另外安裝了pytube套件(語法: !pip install pytube)。使用pytube套件的函式，可以輕鬆抓取YouTube上的影片及資訊，其語法如下：

from pytube import YouTube

pytube套件提供的常用函式如下表：

|  |  |
| --- | --- |
| 函式 | 說明 |
| download | 影片下載 |
| title | 影片標題 |
| views | 欣賞次數 |
| length | 影片長度(單位為秒) |
| rating | 影片評價 |

* **os套件**

Python提供os套件來進行檔案的操作，可以取得檔案路徑、檔案大小、建立目錄、刪除目錄、刪除檔案與執行命令…等操作。

os套件提供相關函式，用以處理檔案路徑與名稱資訊，使用os套件必須先匯入，其語法如下：

import os

os套件提供的常用函式如下表：

|  |  |
| --- | --- |
| 函式 | 說明 |
| abspath( ) | 取得檔案的絕對路徑 |
| basename( ) | 取得路徑最後的檔案名稱 |
| dirname( ) | 取得檔案的目錄路徑，如要取得目前Python檔案所在的目錄路徑，其語法為os.path.dirname(\_\_file\_\_) |
| exists( ) | 檢查檔案或路徑是否存在，其回傳值為True或False |
| getsize( ) | 取得檔案的大小，其回傳單位是Bytes |
| isabs( ) | 檢查該路徑是否為完整路徑 |
| isfile( ) | 檢查該路徑是否為檔案 |
| isdir( ) | 檢查該路徑是否為目錄 |
| split( ) | 分割該路徑為目錄路徑與檔案名稱 |
| splitdrive( ) | 分割該路徑為磁碟名稱與檔案路徑 |
| join( ) | 合併檔案路徑和檔案名稱 |

* **tkinter套件**

tkinter 套件除了視窗元件外，還有許多使用者介面元件，包括：標籤、按鈕、 文字方塊…等，提供給開發者運用，相關說明匯入tkinter套件的語法：

from tkinter import \*

標籤（Label） 標籤本身是一個用來顯示資訊的元件，它常用來提示使用者該如何使用這個 介面，標籤與文字方塊都能夠顯示文字訊息，但是使用者無法在標籤內輸入文 字，而文字方塊可以。 建立標籤的語法如下：

tkinter.Label(容器物件[,參數 1=值, 參數 2=值, …])

容器物件：是指標籤置於該物件之上。

參數：是指對於標籤的相關設定，常見的標籤參數如表所示。

|  |  |
| --- | --- |
| 參數 | 說明 |
| text | 標籤文字 |
| width | 標籤寬度 |
| height | 標籤高度 |
| background | 標籤的背景顏色，簡稱 bg |
| foreground | 標籤的文字顏色，簡稱 fg |
| padx | 標籤文字與標籤物件邊緣的水平間距 |
| pady | 標籤文字與標籤物件邊緣的垂直間距 |
| justify | 對齊方式，有靠左（Left）、置中（Center）、靠右（Right） |
| font | 標籤文字字體與大小，例如：font=('新細明體',14) |

按鈕（Button） 按鈕在視窗軟體中，算是必備的項目，在很多應用程式中都會使用按鈕元 件，用來執行各種事件程序。其實在日常生活中，我們也不曾缺少按鈕，像是門鈴按鈕，當我們按下它時，電鈴會發出鈴聲，在電鈴例子中，按鈕所扮演的腳色， 很明顯的是「產生鈴聲」這個動作的啟動器。 在電腦的世界中，也是同樣的狀況，其運作的動作是由開發者用程式碼來定 義，在 tkinter 套件建立按鈕的語法如下：

tkinter.Button(容器物件[,參數 1=值, 參數 2=值, …])

容器物件：是指按鈕置於該物件之上。

參數：是指對於按鈕的相關設定，常見的按鈕參數如表所示。

|  |  |
| --- | --- |
| 參數 | 說明 |
| text | 按鈕文字 |
| width | 按鈕寬度 |
| height | 按鈕高度 |
| background | 按鈕的背景顏色，簡稱 bg |
| foreground | 按鈕的文字顏色，簡稱 fg |
| padx | 按鈕文字與按鈕物件邊緣的水平間距 |
| pady | 按鈕文字與按鈕物件邊緣的垂直間距 |
| justify | 對齊方式，有靠左（Left）、置中（Center）、靠右（Right） |
| font | 按鈕文字字體與大小，例如：font=('新細明體',14) |
| command | 當使用者按下按鈕時，呼叫 command 所指定的函式 |
| textvariable | 按鈕文字之變數，可用作設定或取得按鈕的文字內容 |
| underline | 按鈕文字加上底線，預設值為-1，代表全部不加底線， 0 表示第 1 個字元，1 表示第 2 個字元，2 表示第 3 個 字元，依此類推 |

* **Threading套件**

在 Python 中若要撰寫多執行緒（multithreading）的平行化程式，最基本的方式是使用 [threading](https://docs.python.org/3.6/library/threading.html) 這個模組來建立子執行緒。threading 是 Python 標準函式庫裡面的模組，所以不用特別安裝即可使用，雖然功能不是很多，但是基本多執行緒程式設計常用的功能它都有，匯入threading套件之語法:

import threading

先來介紹基本的程式碼:

|  |  |
| --- | --- |
| **函式** | **說明** |
| threading.Thread(target=function, name="Thread名字", args=參數) | 建立執行緒，將 target 指定為要執行的 function，name 就是設定的執行緒名稱，若要傳入參數，可以設定 args ，設定格式一定要是 Tuple(變數名稱, ) |
| <Thread>.start() | 開始執行指定的執行緒，start前面要放指定的執行緒名稱 |
| <Thread>.join() | 將主執行緒暫停，等待指定的執行緒結束，join 前面要放指定的執行緒名稱 |
| threading.active\_count() | 查看目前有多少個執行緒 |
| threading.enumerate() | 查看目前使用執行緒的資訊 |
| threading.current\_thread() | 查看目前在哪個執行緒中 |

### 四、參考程式碼

|  |  |
| --- | --- |
| 列數 | 程式碼 |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31 32 33  34  35  36 37  38  39  40 41  42  43 44  45  46  47  48 49 50  51 52  53  54  55 56 57 58  59  60 61  62  63  64 65 66 67 68 69 70  71 72  73  74  75 76  77  78  79 80  81  82 83 84 85  86  87  88  89  90  91  92  93  94 95  96  97  98  99  100  101  102  103  104  105  106  107  108  109 110 | from tkinter import \*  from pytube import YouTube  import os  import threading  def loadVideo():  x.set("狀態:檔案下載中(無反應可多按幾次)...")  vlinks = links.get()  yt = YouTube(vlinks)  threading.Thread(target=yt.register\_on\_progress\_callback(progress)).start()  threading.Thread(target=lambda:downoad(yt,path)).start()  videoViews = yt.views  videoSeconds = yt.length  videoRating = yt.rating  videoTitle = yt.title  a.set(str(videoViews ))  b.set(str(videoSeconds))  c.set(str(videoRating)+" / 5")  d.set(str(videoTitle))  x.set("狀態:影音檔案下載中")  y.set("[Download]:")  def downoad(yt,path):  yt.streams.first().download(path)    def progress(streams, chunk, bytes\_remaining):  if int(bytes\_remaining) == 0:  x.set("狀態:影音檔案下載完成")  y.set("[Download]:██████████ 100%"+" in "+ path)  size = streams.filesize  progress = (float(abs(bytes\_remaining-size)/size))\*float(100)  y.set('[Download]:[%s%s]%.2f%%' % ('█' \* int(progress//5), ' '\*(20-int(progress//5)), float(progress)))    def reMove():  links.set(" ")  def width():  window.title("YouTube影片下載小工具 by王璽鑄")  window.geometry("250x240")  def back():  window.title("YouTube影片下載小工具 by王璽鑄")  window.geometry("500x240")    path="d:\\myYoutube" #位置  if not os.path.isdir(path):  os.mkdir(path)  window = Tk()  window.title("YouTube影片下載小工具 by王璽鑄")  window.geometry("500x240")  x = StringVar()  y = StringVar()  a = StringVar()  b = StringVar()  c = StringVar()  d = StringVar()  links = StringVar()  x.set("狀態:等待輸入影片網址")  y.set("[Download]:")  a.set("...")  b.set("...")  c.set("...")  d.set("...")  lab0 = Label(window,text="請輸入輸入影片網址 : ",bg="lightyellow",width=20).grid(row=0)  lab1 = Label(window,text="影片標題 : ",bg="lightyellow",width=15)  lab2 = Label(window,text="影片評價 : ",bg="lightyellow",width=15)  lab3 = Label(window,text="影片長度(秒) : ",bg="lightyellow",width=15)  lab4 = Label(window,text="影片觀賞次數 : ",bg="lightyellow",width=15)  lab5 = Label(window,textvariable=x,width=45)  lab6 = Label(window,textvariable=y,width=45)  lab11 = Label(window,textvariable=d,width=50)  lab22 = Label(window,textvariable=c,width=50)  lab33 = Label(window,textvariable=b,width=50)  lab44 = Label(window,textvariable=a,width=50)  lab1.place(x=0,y=30)  lab2.place(x=0,y=60)  lab3.place(x=0,y=90)  lab4.place(x=0,y=120)  lab5.place(x=65,y=150)  lab6.place(x=80,y=180)  lab11.place(x=120,y=30)  lab22.place(x=120,y=60)  lab33.place(x=120,y=90)  lab44.place(x=120,y=120)  e1 = Entry(window,textvariable=links,width=40)  e1.grid(row=0,column=1)  btn1 = Button(window,text="下載",command=loadVideo)  btn1.place(x=100,y=210)  btn2 = Button(window,text="重置",command=reMove)  btn2.place(x=200,y=210)  btn3 = Button(window,text="結束",command=window.destroy)  btn3.place(x=300,y=210)  menu = Menu(window) #視窗變化  window["menu"] = menu  filemenu = Menu(menu, tearoff=0)  menu.add\_cascade(label="視窗調整", menu=filemenu)  filemenu.add\_command(label="視窗變窄", command=width)  filemenu.add\_separator()  filemenu.add\_command(label="回復", command=back)  window.mainloop() |

### 五、程式碼解說

第1~4行:匯入各項所需的套件，包括：tkinter套件、os套件、pytube套件、threading套件等。

第6~23行:設定函式loadVideo()，為了讓程式的進度條可以順利運作，使用了threading套件進行多線程運作，並使用pytube進行爬蟲抓取(影片標題、影片評價、影片長度、影片觀賞次數)。

第25~26行:為pytube語法的抓取影片函式，而命名為downoad則是因為保留字源的關係，無法使用download。

第28~34行:為下載進行中所顯示的進度條，而progress則與yt.register\_on\_progress\_callback進行下載進度回傳，下載完成則顯示:"狀態:影音檔案下載完成"，下載中則會持續更新進度條，這邊有搭配字元轉型來做顯示。

第36~37行:清空輸入欄的功能

第39~41行:減小視窗欄的功能

第42~44行:回復視窗欄的功能

第46~48行:設定影片下載的位置路徑

第50~52行:使用tkinter創造出一個功能視窗

第54~66行:參數設定

第68~78行:設定視窗內的顯示標籤(字體、顏色、長度)

第80~89行:設定各個標籤之位置

第91~92行:設定輸入框及所在位置

第93~94行:設定下載鈕及所在位置

第95~96行:設定重置鈕及所在位置

第97~98行:設定結束鈕及所在位置

第100~106行:設定視窗調整子欄位，點開提供功能"視窗變窄"及"回復"選擇

第108行:程式重複執行

### 六、心得

當初在做這個專題的時候，我同時也打了一個抓PTT文章的爬蟲，但是後來覺得抓Youtube的這個比較有趣一點，畢竟有影音檔而且實用度也比較高。一開始在做的時候功能比較簡單一點，只有輸入網址跟下載功能，後來覺得太少了，於是就再加上圖形化介面tkinter以及影片資訊欄的輸出，整體也比較不會太單調，後來在上傳檔案前又看到了還可以加入進度條的功能，就讓程式變得更豐富了些。

在弄程式的時候，花了非常大量的時間在處理套件的更新上，十分頭痛，因為pytube這個套件常常會抓不到影片；YouTube 對於這種使用爬蟲技術來下載影片的方法一直都在換花樣阻擋，必須要隨時更新我們使用的套件版本才有辦法使用。

第二個碰到的困難就是下載進度回傳的問題，剛開始一直無法進行回傳的功能，後來詢問了資工系跟電機系的學長，採用了多線程式的方法解決，進度條才順利完成。

程式就是一門學不完的學問，一直弄一直撞牆。身為電機系的學生雖然碰到程式不算陌生，但是對於Python是第一次接觸，算是開啟了另一片新的天地，感覺起來語法十分直觀簡單，有些很麻煩的轉型都不太需要用到，大部分的套件(函式庫)也都已經有人打好了，只需要找到並且套用即可，非常方便。

這次的專題就像一顆種子一樣，期待有一天可以從土裡萌芽成長茁壯；我也準備好暑假的時候要多多練習新的程式跟實作，持續的增進自己的軟體實力，不再一直待在舒適圈裡，加油!

