

## ● Project Name：查找歌曲小工具

## ● Proposal Plan

## ● Project motivation

youtube 上歌曲 mv 內容完整且齊全，但是資訊欄的部分內容較為雜亂，格式沒有統一，有些沒有附上歌詞，雖然有標示專輯和歌手，但分類的模式比 spotify 等線上串流音樂平台還不完善及直觀（例如點入歌手的 youtube 頻道，裡面有時會包含歌曲以外的影片，沒辦法單純只瀏覽歌曲清單），因此想要結合串流音樂平台的分類模式和 youtube 的 mv，建立一個平台能夠同時簡潔明瞭的顯示歌曲 mv、歌詞、專輯和歌手，並能夠快速的去取得同專輯或歌手的其他作品。

## ● Project plan

- 一、歌曲資訊（包含歌曲名稱、歌手、專輯）：使用 spotipy 取得該歌曲的所有資訊後進行過濾，留下想要的資訊並印出。
- 二、專輯及歌手的歌曲列表：使用 spotipy 取得，存成陣列後印出。
- 三、歌詞：使用 beautifulsoup 爬蟲「魔鏡歌詞網」（因為該網頁的歌詞齊全），若有外文歌詞，可讓使用者選擇是否顯示中文翻譯。
- 四、youtube mv：先獲得歌曲的 youtube MV 網址（想辦法確保取得的是 MV 而不是歌詞版或演唱會版的影片），再進一步透過 youtube api 取得影片。
- 五、主程式架構：各步驟的溝通，像是從歌曲回到上一層（歌手或專輯）的寫法，整個輸出版面的配置。
- 六、使用者輸入的格式不符合預期的處理：利用 try&except 請使用者再次輸入。
- 七、使用者輸入的歌名有多種結果的處理：目前計畫讓使用者再進一步選擇歌手或者透過瀏覽影片的縮圖讓使用者選擇。

## ● How a user would interact with your finished project

- 一、輸入：使用者輸入歌曲就直接印出該歌曲資訊。
- 二、印出歌曲資訊：印出以下內容
  - a. 歌曲名稱、專輯跟歌手
  - b. 歌詞
  - c. MV
- 三、回到上一層：使用者可選擇回到該歌曲的專輯或者歌手，再次選擇想查詢的歌曲。

## ● Timeline of the project

12/14~12/20 認識和取得 youtube 和 spotify 的 api，確定要使用的 function。

12/21~12/27 程式基本架構的完成。

12/28~1/3 檢查使用者體驗是否良好，若輸出結果不符合預期的處理，檢查一些特殊或極端的例子是否也可以適用。

1/4~1/7 最後的優化和 demo。

## ● Update 1

### ● What I have done

1. 認識 youtube 及 spotify 的 api：先註冊和授權取得 api key，透過官方網站的介紹選擇自己要使用的函式庫。
2. 透過各種管道取得歌曲資訊，分別為：
  - a. youtube api 取得歌曲影片
  - b. spotify api 取得歌手和專輯名稱
  - c. 魔鏡歌詞網取得歌詞
3. 中文轉碼，否則有包含中文的網址在執行時會出現 error
4. 架構出部分主程式：和使用者互動（輸入歌曲名）、歌曲資訊的排版
5. 相同歌名的處理：因為無法確保該影片縮圖能讓使用者辨識出歌曲，故後來決定讓使用者透過選擇歌手找到自己真正想要的歌
6. 多處使用 try&except 迴避掉錯誤和非預期的輸入和操作

### ● Any changes on the final project plan

1. 目前不確定 youtube api 能否直接引入 mv 影片放置在 jupyter 頁面，（仍然在爬文及測試中），若無法做到的話將以影片縮圖及影片網址代替
2. 較難確保取得影片是 MV 而不是歌詞版或演唱會版，而且有些歌曲也沒有 MV，目前想出較佳的作法是抓取搜尋結果中觀看數最高的，（利用 search function 中 order="viewCount" 去指定），經過觀察發現觀看數最高的影片往往是歌曲 MV 或最其具代表性的影片
3. spotify 即使是中文歌手仍會顯示其英文名字（例如周杰倫會是 Jay Chou），我選擇透過 google 進入該歌手的維基百科以取得歌手的中文名
4. 魔鏡歌詞網的外文歌不一定有中文翻譯，故最後統一一律顯示原文。

### ● Timeline for the rest

1/1~1/3：

完成所有主程式架構：各層級之間的溝通（從歌曲回到歌手或專輯等）

研究如何在 jupyter 介面顯示 youtube 影片或尋找替代方案

1/4~1/7：

找不到歌曲資訊或歌詞的處理

確保能夠抓到較冷門歌曲

小細節的修正和優化（確保介面簡潔、使用者輸入方法直觀）

## ● Update 2

1. 因為找不到合適方法將 **youtube** 影片或影片縮圖放入頁面，故以 **youtube** 影片網址代替（若要置入影片必須得先下載該影片，置入圖片則只能多開一個彈出式視窗顯示該圖片，經評估之後覺得有點多此一舉，故捨棄）
2. 後來發現如果搜尋歌曲 **MV** 時用觀看數當排序依據，第一個結果有可能會出現類似歌名但觀看數較高的歌曲（即搜尋結果不夠精準），故最後選擇用關聯性（**relevance**）為選擇歌曲 **MV** 的標準。
3. 適時的讓使用者能夠清空輸出頁面，避免資訊太過混亂
4. 在多處加上返回紐，讓使用者在發現結果不符合預期時能夠返回重新搜尋。
5. 為輸出結果加上顏色和加粗，因為輸出資訊量很大，透過顏色讓使用者能夠更快速找到自己需要的資訊在哪

## ● depiction of my final project

一個方便使用者搜尋歌曲並透過**結合各平台**完整顯示歌曲資訊（歌名、歌手、專輯、歌詞）的小工具，並能夠透過歌手和專輯查找相關歌曲。

## ● 心得

## ● 遇到的困難：

1. 中文歌手在 **spotify** 裡會是英文名：當初花了很多時間處理這個問題，本來想使用 **google** 翻譯，但發現翻譯的結果不盡正確，後來爬蟲 **google** 搜尋「歌手的英文名 維基百科」的網頁，發現搜尋的第一個結果就會是顯示歌手中文名的維基百科。（請見下方例子）



2. 歌詞網站的格式沒有統一，舉例來說，有些歌詞有 **MV** 時間和歌詞的對照表但有些沒有，需要多觀察幾首，並透過 **if** 針對不同格式做不同的操作，確保每次都能只擷取到歌詞的部分。（以下為不同格式的歌詞

形式舉例)

Got a long list of ex-lovers They'll tell you I'm insane But I got a blank space baby And I'll write your name	怎麼一不小心太瘋狂 別後悔 就算錯過 在以後 你少不免想起我 還算不錯 當我不在 你會不會難過 你夠不夠我這樣灑脫 說不上愛別說謊 就一點喜歡 說不上恨別糾纏 別裝作感嘆 將一切都體諒 將一切都原諒 我嘗試找答案 而答案很簡單 簡單得很遺憾 因為成長 我們還不得已要習慣 因為成長 我們忽爾間說散就散
時間和歌詞的對照表(要去除)	
[00:00.00]Blank Space [00:00.88]Taylor Swift [00:01.64] [00:02.76] [00:05.62]Nice to meet you, where you've been? [00:08.04]I can show you incredible things [00:10.57]Magic, madness, heaven, sins [00:13.08]Saw you there and I thought	All comments ( 0 ) "Currently no comment."

3. 無法爬蟲具有中文的網址：解決方法是透過 `urllib.parse.quote` 這個函式把中文先轉成電腦可讀的編碼。

## ● 覺得好玩的地方：

當初在優化程式的時候請一位很愛聽獨立樂團的朋友幫我做測試，發現很多較冷門的歌曲會搜尋不到，故我一次次優化搜尋的方法，確保都能找到歌曲的資訊，也讓自己的程式從本來一字不漏才能搜尋到歌曲到可以容許一點點的出入。透過這次的經驗讓我很佩服 google 等搜尋引擎，就算打的關鍵字有些偏差也往往能搜尋到想要的結果。

另一方面，一開始設計的程式對使用者較不友善，給身邊朋友測試的時候被抱怨操作不夠直觀，因此我透過朋友的建議修改細節，讓即使是初次使用的人也能在不被指導的情況下自己操作程式，這件事讓我體會到除了功能要實用外，也必須要注重使用者的體驗，看到朋友很流暢的使用自己做的小工具真的很有成就感。

## ● 學到的東西：

1. 多次使用 `try` 和 `except` 去迴避掉使用者不符預期的輸入：在設計這個部分時必須要設身處地設想使用者所有可能會發生的錯誤輸入，並提示使用者再次輸入。
2. 自學的能力：以清空輸出結果的寫法為例，在網路上可找到數種方法，但得自己一個個親自測試後才能找出最適合當下情況的函式。此外，也透過這次的 project 才發現 python 的輸出結果可以改顏色和換粗體，這次的經驗讓我發現 python 真的有很多實用的工具和變化性等著我們去發掘。

## ● Run

### ● How to run the project

#### 1. 使用者輸入歌名（中英文皆可）：

提醒使用者若是中文歌但歌名為全英文或數字，在須加上(c)標註（若是中文歌需要將歌手名轉成中文，但是否為中文歌的判斷標準為歌名

是否為中文，故若非中文歌名但仍需轉換就要標註)

**2. 使用者選擇歌手：**

因為同樣或類似的歌名可能是不同首歌，讓使用者透過歌手去選擇自己要的歌（若搜尋結果只有一個，則不用讓使用者選擇）

**3. 印出結果，讓使用者選擇下一步：**

有以下幾個選擇：

a.結束程式運行

b.清除所有結果，使版面整潔（但會留下上首歌的歌手和專輯，讓使用者可以選擇下一步）

c.列出該歌曲歌手的熱門歌曲給使用者選擇要查看哪首

d.列出該歌曲專輯的其他歌曲給使用者選擇要查看哪首

e.不沿用上首歌的資訊，重新開始新的搜尋

**4. 重複上述動作；**

使用者依自己的想法決定要回到哪一層或是重新開始

● **How to install any needed libraries**

（我使用 anaconda 的 jupyter 製作 final project）

請從 anaconda prompt 輸入以下指令下載必要函式庫

1. pip install spotipy

2. pip install google-api-python-client

3. pip install easydl 或 pip install easydl==2.0.7