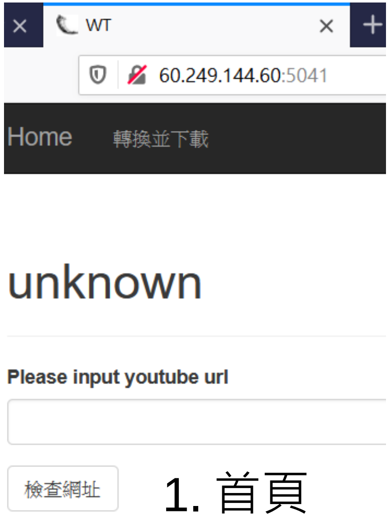
Web Application By Python & Flask

在 Flask 框架上實現 Youtube 轉 mp3

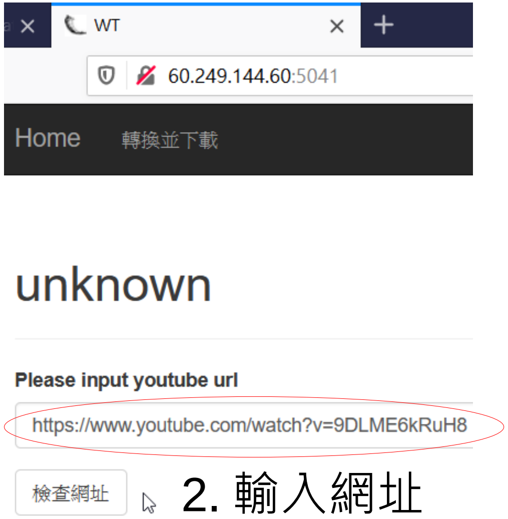


60.249.144.60:5041

unknown

Please input youtube url

1. 首頁



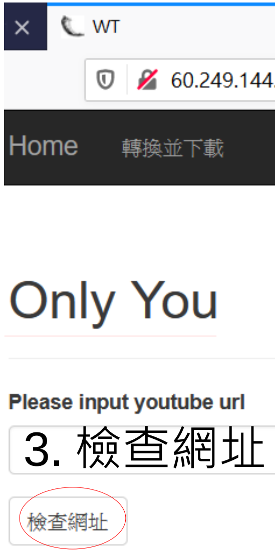
60.249.144.60:5041

unknown

Please input youtube url

https://www.youtube.com/watch?v=9DLME6kRuH8

2. 輸入網址



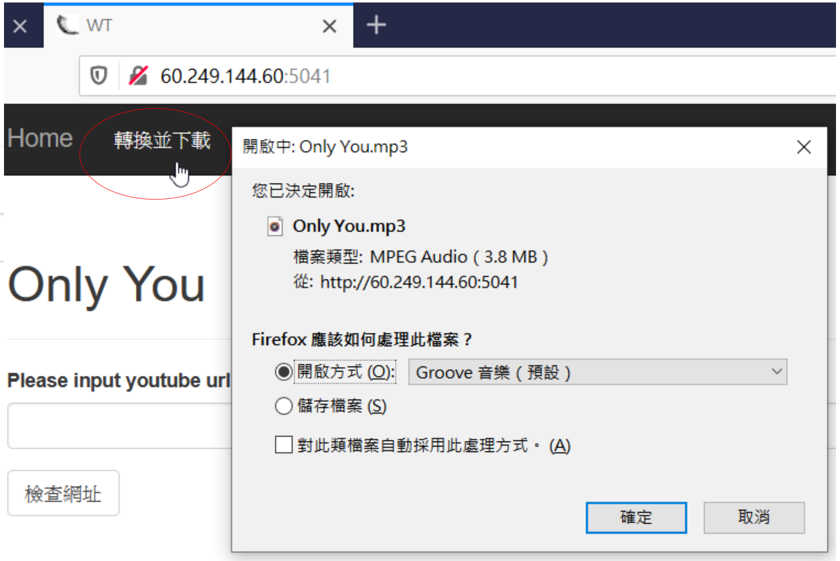
60.249.144.60:5041

Only You

Please input youtube url

3. 檢查網址

4. 轉換並下載



60.249.144.60:5041

Only You

Please input youtube url

開啟中: Only You.mp3

您已決定開啟:

Only You.mp3

檔案類型: MPEG Audio (3.8 MB)

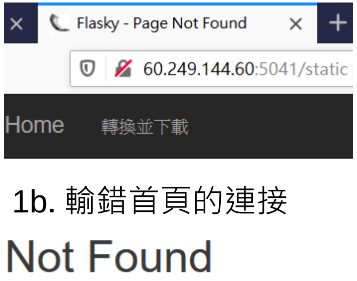
從: http://60.249.144.60:5041

Firefox 應該如何處理此檔案？

● 開啟方式 (O): Groove 音樂 (預設)

○ 儲存檔案 (S)

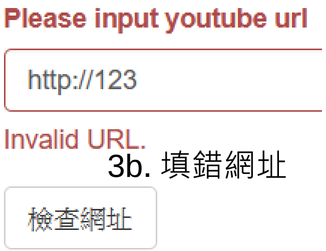
□ 對此類檔案自動採用此處理方式。(A)



60.249.144.60:5041/static

1b. 輸錯首頁的連接

Not Found

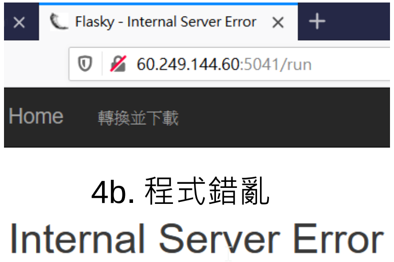


Please input youtube url

http://123

Invalid URL.

3b. 填錯網址



60.249.144.60:5041/run

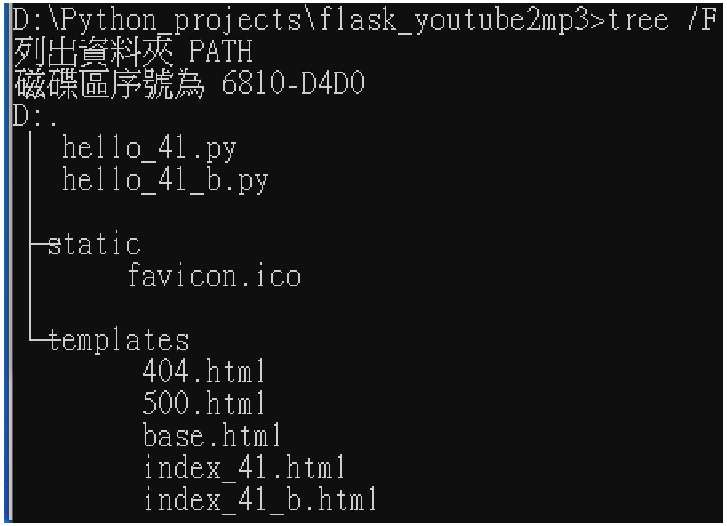
4b. 程式錯亂

Internal Server Error

安裝 Flask 和其相關子 Packages

- Flask
  - 虛擬環境（非必要）
    - python3 -m venv venv_flask
  - 主要的 flask 框架
    - pip3 install flask
- 子 packages
  - pip3 install
    - flask-bootstrap, flask-moment, flask-wtf

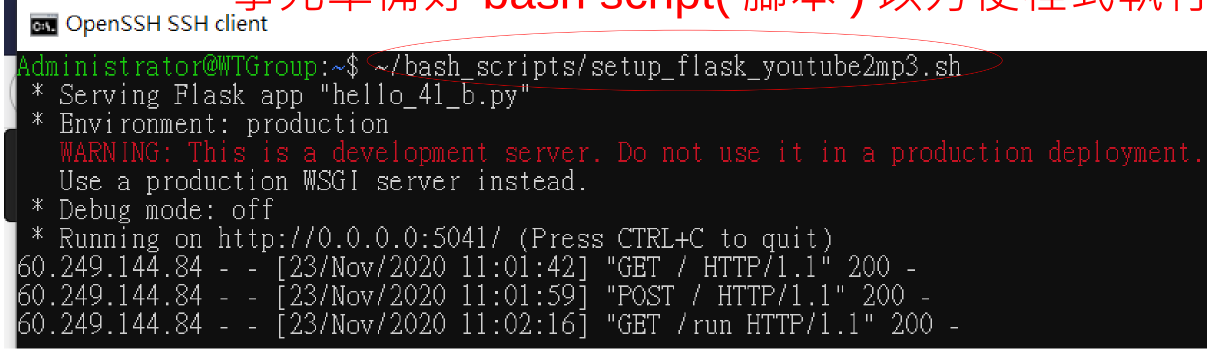
程式架構和執行



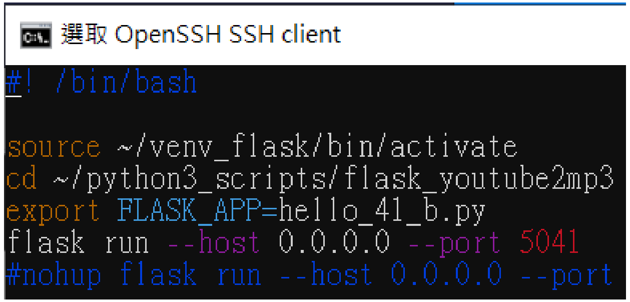
```
D:\Python_projects\flask_youtube2mp3>tree /F
列出資料夾 PATH
磁碟區序號為 6810-D4D0
D:.
│  hello_41.py
│  hello_41_b.py
│
├─static
│      favicon.ico
│
└─templates
        404.html
        500.html
        base.html
        index_41.html
        index_41_b.html
```

程式檔案的架構，以上檔案皆為書本的範例，我只是修改內容

事先準備好 bash script(腳本) 以方便程式執行



```
Administrator@WTGroup:~$ ~/bash_scripts/setup_flask_youtube2mp3.sh
 * Serving Flask app "hello_41_b.py"
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://0.0.0.0:5041/ (Press CTRL+C to quit)
60.249.144.84 - - [23/Nov/2020 11:01:42] "GET / HTTP/1.1" 200 -
60.249.144.84 - - [23/Nov/2020 11:01:59] "POST / HTTP/1.1" 200 -
60.249.144.84 - - [23/Nov/2020 11:02:16] "GET /run HTTP/1.1" 200 -
```

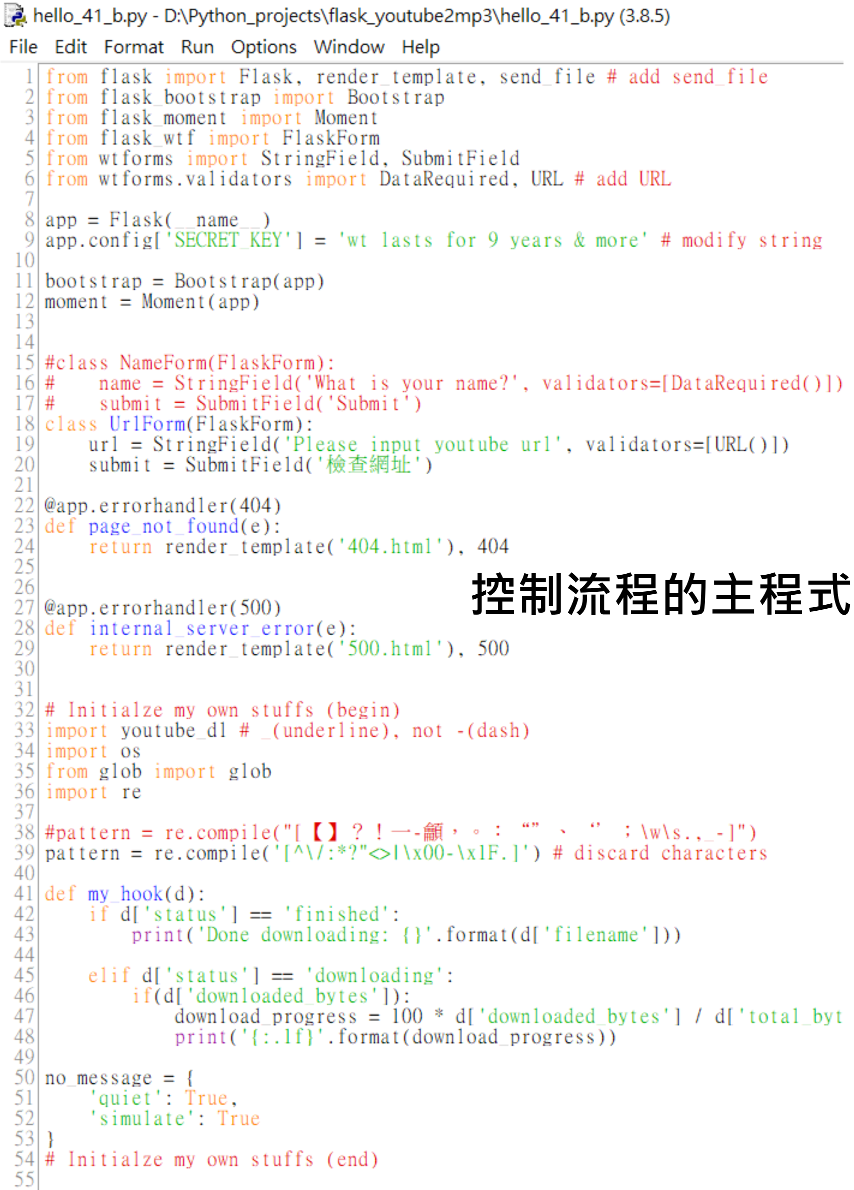


```
#! /bin/bash

source ~/venv_flask/bin/activate
cd ~/python3_scripts/flask_youtube2mp3
export FLASK_APP=hello_41_b.py
flask run --host 0.0.0.0 --port 5041
#nohup flask run --host 0.0.0.0 --port
```

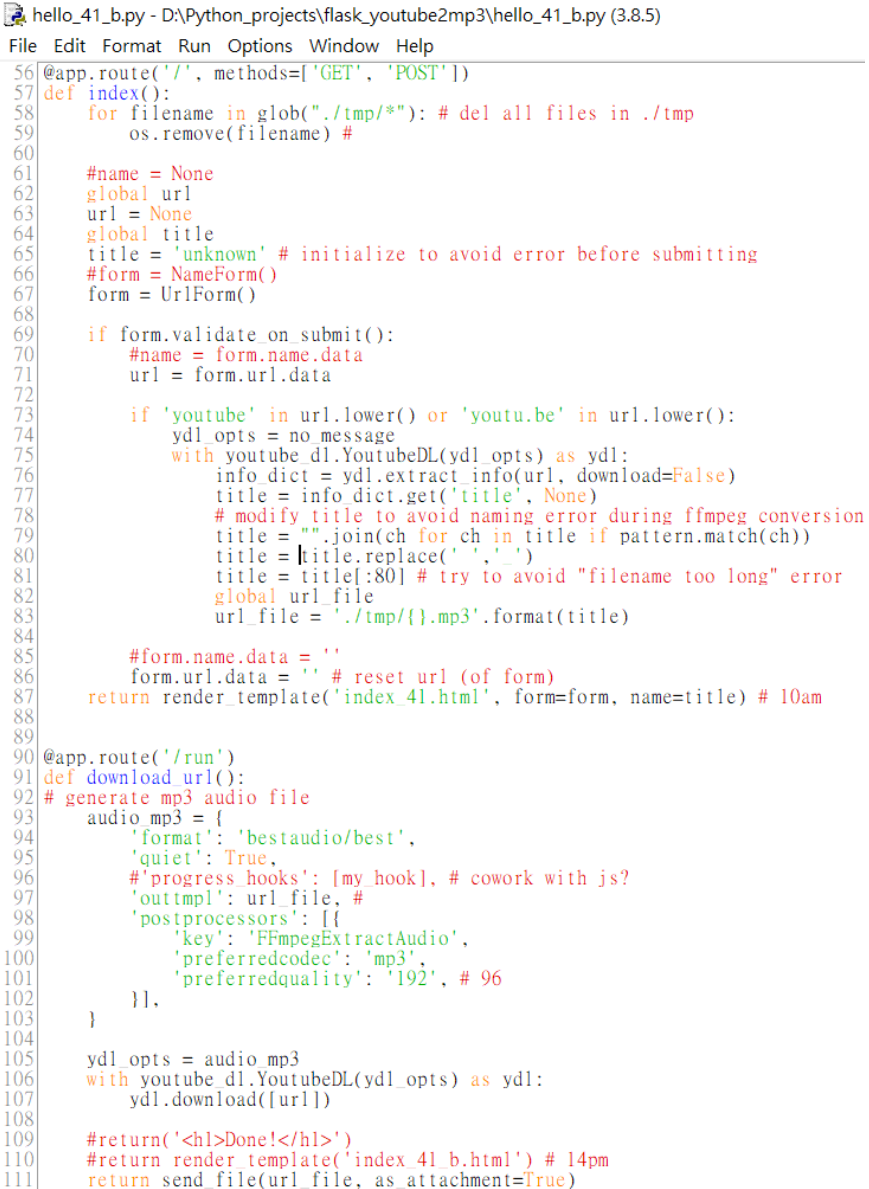
bash script 的內容

在 server 端執行 vi ~/bash_scripts/setup_flask_youtube2mp3.sh 後產生上圖

控制流程的主程式 (Python)

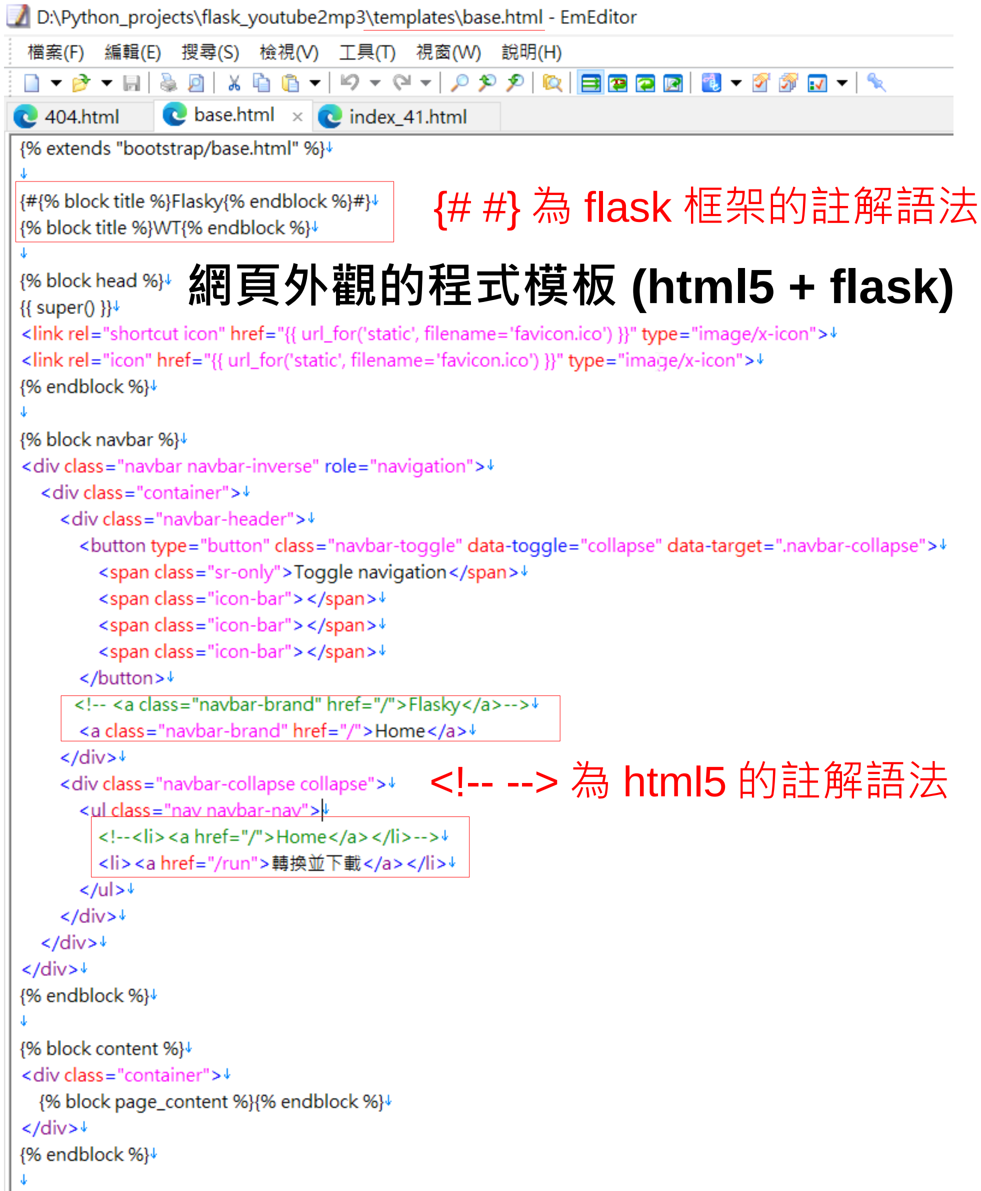

hello_41_b.py - D:\Python_projects\flask_youtube2mp3\hello_41_b.py (3.8.5)
File Edit Format Run Options Window Help

```python
from flask import Flask, render_template, send_file # add send_file
from flask_bootstrap import Bootstrap
from flask_moment import Moment
from flask_wtf import FlaskForm
from wtforms import StringField, SubmitField
from wtforms.validators import DataRequired, URL # add URL

app = Flask(__name__)
app.config['SECRET_KEY'] = 'wt lasts for 9 years & more' # modify string

bootstrap = Bootstrap(app)
moment = Moment(app)


#class NameForm(FlaskForm):
#    name = StringField('What is your name?', validators=[DataRequired()])
#    submit = SubmitField('Submit')
class UrlForm(FlaskForm):
    url = StringField('Please input youtube url', validators=[URL()])
    submit = SubmitField('檢查網址')

@app.errorhandler(404)
def page_not_found(e):
    return render_template('404.html'), 404


@app.errorhandler(500)
def internal_server_error(e):
    return render_template('500.html'), 500


# Initialze my own stuffs (begin)
import youtube_dl # _(underline), not -(dash)
import os
from glob import glob
import re

#pattern = re.compile("[【】？！—-顫，。：“”、‘’；\w\s.,_-]")
pattern = re.compile('[^\/:*?"<>|\x00-\x1F.]') # discard characters

def my_hook(d):
    if d['status'] == 'finished':
        print('Done downloading: {}'.format(d['filename']))

    elif d['status'] == 'downloading':
        if(d['downloaded_bytes']):
            download_progress = 100 * d['downloaded_bytes'] / d['total_byt
            print('{:.1f}'.format(download_progress))

no_message = {
    'quiet': True,
    'simulate': True
}
# Initialze my own stuffs (end)

```


hello_41_b.py - D:\Python_projects\flask_youtube2mp3\hello_41_b.py (3.8.5)
File Edit Format Run Options Window Help

```python
@app.route('/', methods=['GET', 'POST'])
def index():
    for filename in glob("./tmp/*"): # del all files in ./tmp
        os.remove(filename) #

    #name = None
    global url
    url = None
    global title
    title = 'unknown' # initialize to avoid error before submitting
    #form = NameForm()
    form = UrlForm()

    if form.validate_on_submit():
        #name = form.name.data
        url = form.url.data

        if 'youtube' in url.lower() or 'youtu.be' in url.lower():
            ydl_opts = no_message
            with youtube_dl.YoutubeDL(ydl_opts) as ydl:
                info_dict = ydl.extract_info(url, download=False)
                title = info_dict.get('title', None)
                # modify title to avoid naming error during ffmpeg conversion
                title = "".join(ch for ch in title if pattern.match(ch))
                title = title.replace(' ','_')
                title = title[:80] # try to avoid "filename too long" error
                global url_file
                url_file = './tmp/{}.mp3'.format(title)

        #form.name.data = ''
        form.url.data = '' # reset url (of form)
    return render_template('index_41.html', form=form, name=title) # 10am


@app.route('/run')
def download_url():
# generate mp3 audio file
    audio_mp3 = {
        'format': 'bestaudio/best',
        'quiet': True,
        #'progress_hooks': [my_hook], # cowork with js?
        'outtmpl': url_file, #
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192', # 96
        }],
    }

    ydl_opts = audio_mp3
    with youtube_dl.YoutubeDL(ydl_opts) as ydl:
        ydl.download([url])

    #return('<h1>Done!</h1>')
    #return render_template('index_41_b.html') # 14pm
    return send_file(url_file, as_attachment=True)
```

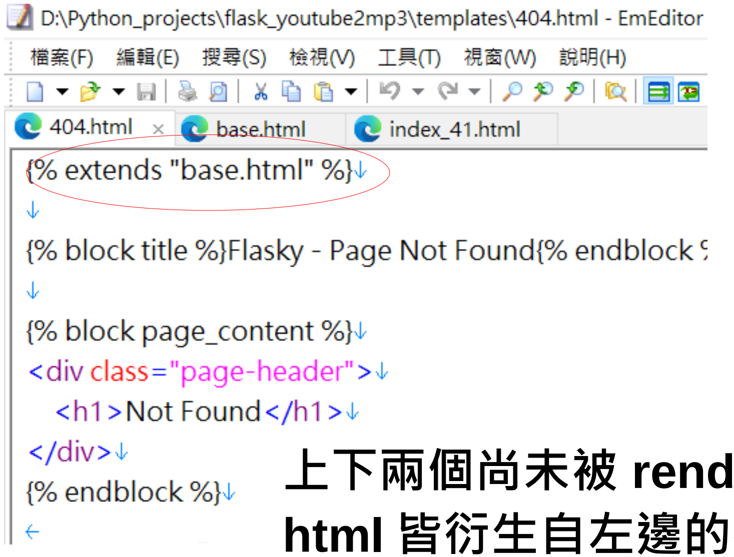
網頁外觀的程式模板 (html5 + flask)


D:\Python_projects\flask_youtube2mp3\templates\base.html - EmEditor
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 工具(T) 視窗(W) 說明(H)
404.html
base.html
index_41.html

```
{% extends "bootstrap/base.html" %}

{#{% block title %}Flasky{% endblock %}#}
{% block title %}WT{% endblock %}

{% block head %}
{{ super() }}
<link rel="shortcut icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
<link rel="icon" href="{{ url_for('static', filename='favicon.ico') }}" type="image/x-icon">
{% endblock %}

{% block navbar %}
<div class="navbar navbar-inverse" role="navigation">
    <div class="container">
        <div class="navbar-header">
            <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
                <span class="sr-only">Toggle navigation</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
           <!-- <a class="navbar-brand" href="/">Flasky</a>-->
            <a class="navbar-brand" href="/">Home</a>
        </div>
        <div class="navbar-collapse collapse">
            <ul class="nav navbar-nav">
                <!--<li><a href="/">Home</a></li>-->
                <li><a href="/run">轉換並下載</a></li>
            </ul>
        </div>
    </div>
</div>
{% endblock %}

{% block content %}
<div class="container">
    {% block page_content %}{% endblock %}
</div>
{% endblock %}

```

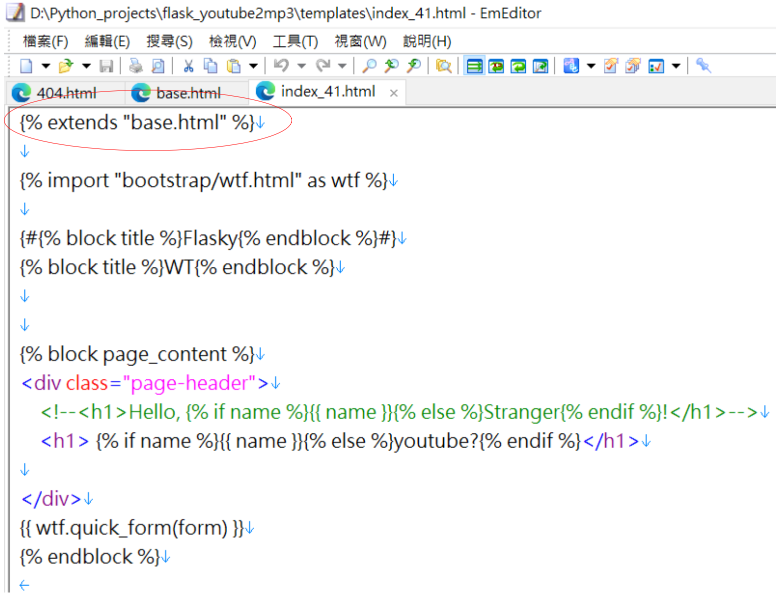
{# #} 為 flask 框架的註解語法

<!-- --> 為 html5 的註解語法


D:\Python_projects\flask_youtube2mp3\templates\404.html - EmEditor
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 工具(T) 視窗(W) 說明(H)
404.html
base.html
index_41.html

```
{% extends "base.html" %}

{% block title %}Flasky - Page Not Found{% endblock

{% block page_content %}
<div class="page-header">
    <h1>Not Found</h1>
</div>
{% endblock %}
```

上下兩個尚未被 render 的 html 皆衍生自左邊的模板

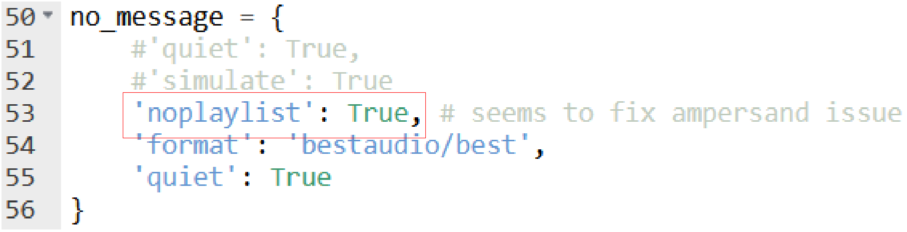

D:\Python_projects\flask_youtube2mp3\templates\index_41.html - EmEditor
檔案(F) 編輯(E) 搜尋(S) 檢視(V) 工具(T) 視窗(W) 說明(H)
404.html
base.html
index_41.html

```
{% extends "base.html" %}

{% import "bootstrap/wtf.html" as wtf %}

{#{% block title %}Flasky{% endblock %}#}
{% block title %}WT{% endblock %}


{% block page_content %}
<div class="page-header">
    <!--<h1>Hello, {% if name %}{{ name }}{% else %}Stranger{% endif %}!</h1>-->
    <h1> {% if name %}{{ name }}{% else %}youtube?{% endif %}</h1>

</div>
{{ wtf.quick_form(form) }}
{% endblock %}

```

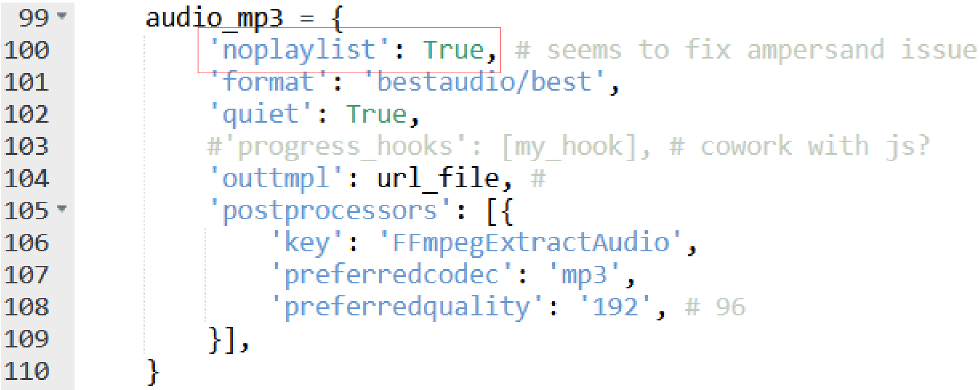
改善穩定度 (youtube-dl)

https://www.youtube.com/watch?v=ZAlSw_dVq1Q&list=FLE68DZvmQ0NxAkasEBbagAw&index=35 (可能會出錯)

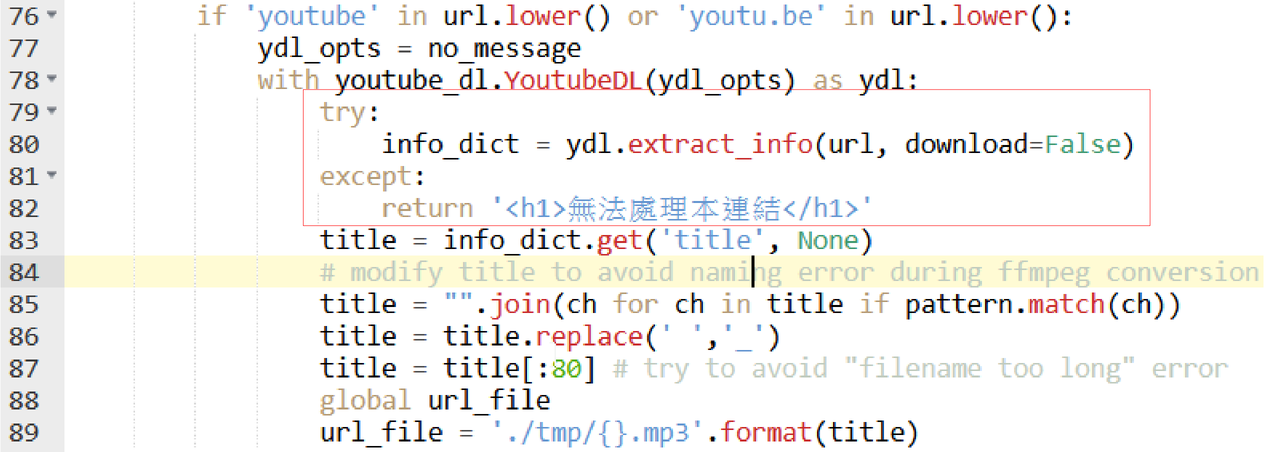
https://www.youtube.com/watch?v=ZAlSw_dVq1Q (正常)



```
no_message = {
    #'quiet': True,
    #'simulate': True
    'noplaylist': True, # seems to fix ampersand issue
    'format': 'bestaudio/best',
    'quiet': True
}
```



```
    audio_mp3 = {
        'noplaylist': True, # seems to fix ampersand issue
        'format': 'bestaudio/best',
        'quiet': True,
        #'progress_hooks': [my_hook], # cowork with js?
        'outtmpl': url_file, #
        'postprocessors': [{
            'key': 'FFmpegExtractAudio',
            'preferredcodec': 'mp3',
            'preferredquality': '192', # 96
        }],
    }
```



```
        if 'youtube' in url.lower() or 'youtu.be' in url.lower():
            ydl_opts = no_message
            with youtube_dl.YoutubeDL(ydl_opts) as ydl:
                try:
                    info_dict = ydl.extract_info(url, download=False)
                except:
                    return '<h1>無法處理本連結</h1>'
                title = info_dict.get('title', None)
                # modify title to avoid naming error during ffmpeg conversion
                title = "".join(ch for ch in title if pattern.match(ch))
                title = title.replace(' ','_')
                title = title[:80] # try to avoid "filename too long" error
                global url_file
                url_file = './tmp/{}.mp3'.format(title)
```