

AIoT：樹莓派應用

Chapter 7：影像串流

賴秉樑 debugger

學院創辦人

課程網址 <https://max543.com/debugger>

串流技術

- 串流 (Streaming)：從伺服器端傳送壓縮媒體資料至客戶端來及時觀看影像，在 Web 客戶端的使用者不需要花費時間等待下載整個檔案，就能即時播放視訊內容。

實驗 7-1：架設 RTSP 串流伺服器

目的：學會設定影像串流 (Streaming)

RTSP Server

- Pi 已內建 RTSP，輸入：

自行視情況加入旋轉參數

```
$ raspivid -o - -t 0 -hf --rotation 180 -w 320 -h 240 -  
fps 15 | cvlc -vvv stream:///dev/stdin --sout  
'#rtp{sdp=rtsp://:8554}' :demux=h264
```

Port Number

- 若有無法執行，檢查有無 port 占用

找出佔用的 Port Number 的 PID

```
$ sudo netstat -ltn |grep 8554
```

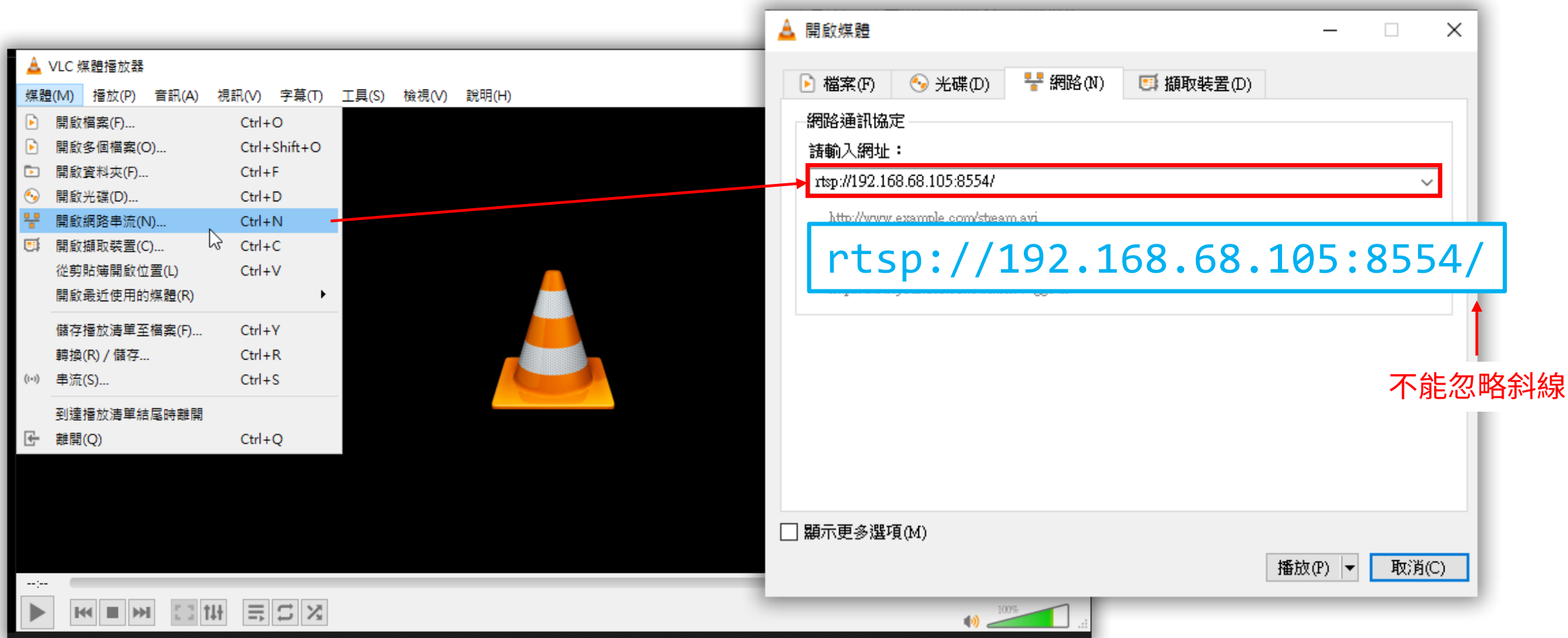
刪除佔用的 PID

```
$ sudo kill -9 PID
```

RTSP Client

- 在 Windows 使用 VLC

✓ <https://get.videolan.org/vlc/3.0.14/win64/vlc-3.0.14-win64.exe>



如何降低延遲？

- 安裝輕量級的 RTSP Server

```
$ cd ~
```

```
$ git clone https://github.com/mpromonet/h264_v4l2_rtspserver.git
```

```
$ sudo apt-get install liblivemedia-dev libv4l-dev cmake
```

```
$ cd h264_v4l2_rtspserver
```

```
$ cmake .
```

```
$ make -j4
```

```
$ sudo ./v4l2rtspserver -F 15 -W 800 -H 600 -P 8554 /dev/video0
```

沒有支援旋轉鏡頭參數的指令，也可以用以下的指令：

```
$ v4l2-ctl --set-ctrl vertical_flip=1
```

RTSP Client

VLC 媒體播放器

媒體(M) 播放(P) 音訊(A) 視訊(V) 字幕(T) 工具(S) 檢視(V) 說明(H)

- 開啟檔案(F)... Ctrl+O
- 開啟多個檔案(O)... Ctrl+Shift+O
- 開啟資料夾(F)... Ctrl+F
- 開啟光碟(D)... Ctrl+D
- 開啟網路串流(N)... Ctrl+N
- 開啟擷取裝置(C)... Ctrl+C
- 從剪貼簿開啟位置(L) Ctrl+V
- 開啟最近使用的媒體(R)
- 儲存播放清單至檔案(F)... Ctrl+Y
- 轉換(R) / 儲存... Ctrl+R
- 串流(S)... Ctrl+S
- 到達播放清單結尾時離開
- 離開(Q) Ctrl+Q

開啟媒體

檔案(F) 光碟(D) 網路(N) 擷取裝置(D)

網路通訊協定

請輸入網址：

rtsp://192.168.68.105:8554/

http://www.example.com/stream.avi

顯示更多選項(M)

播放(P) 取消(C)

加上 unicast

rtsp://192.168.68.105:8554/unicast

RTSP Client for Android / iOS



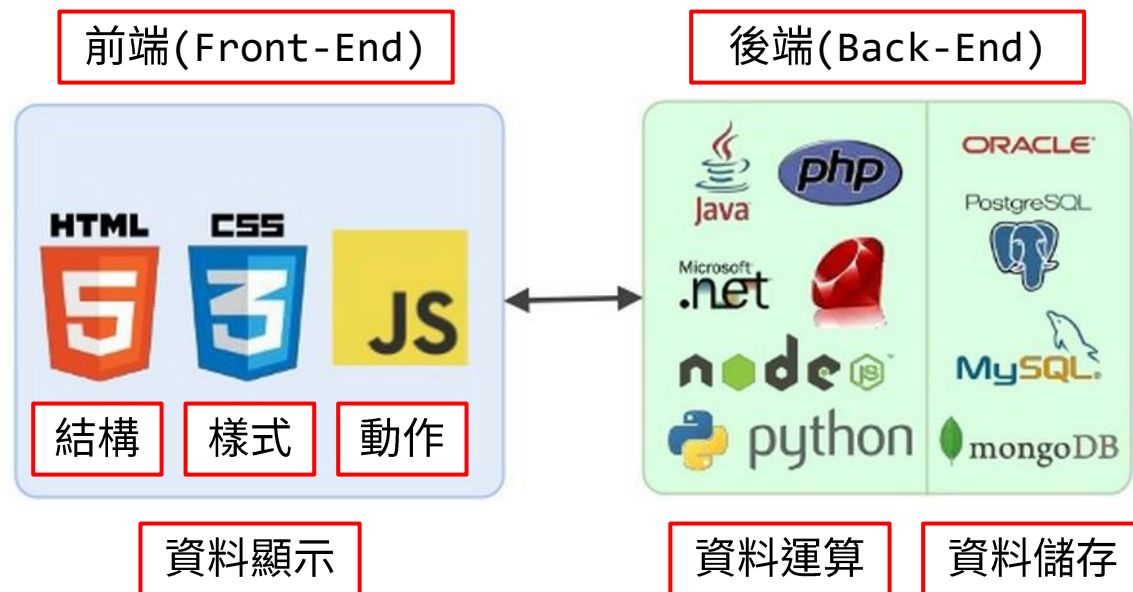
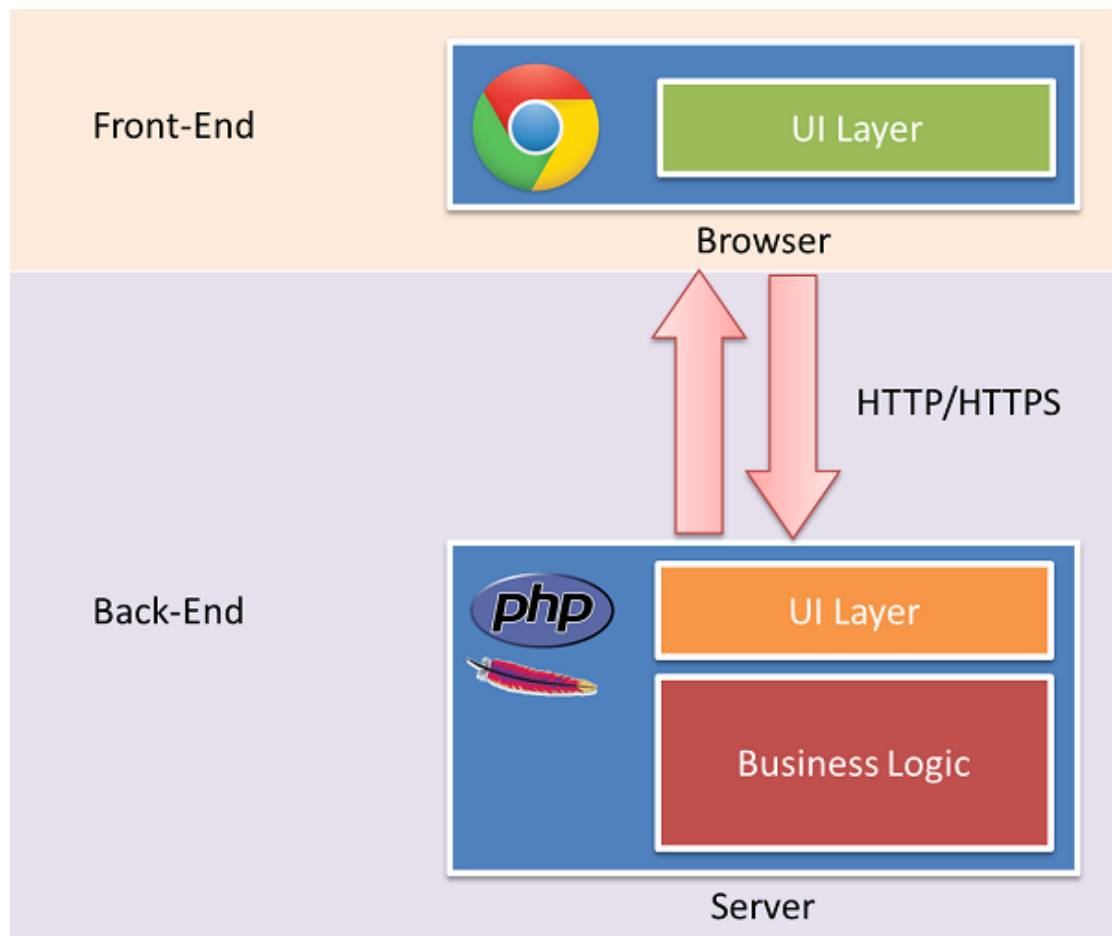
實驗 7-2：用 HTTP 做 Streaming

目的：使用 HTTP + MJPG

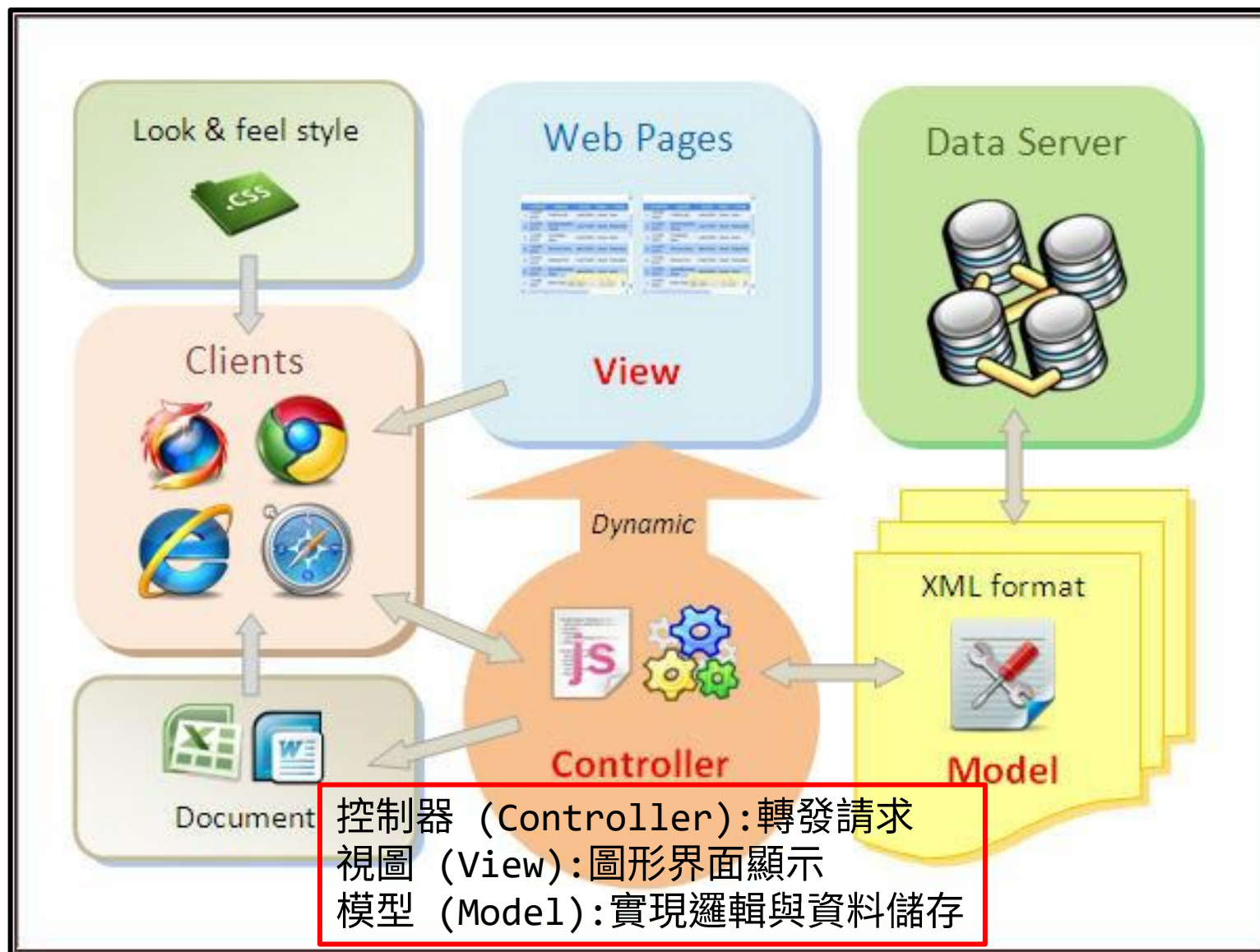
使用 HTTP 做 Video Streaming

- 原理
 - ✓ 向 Web Server 請求一個很大的檔案。
 - ✓ 該檔案是一個即時的資料。

網頁的運作與實作原理

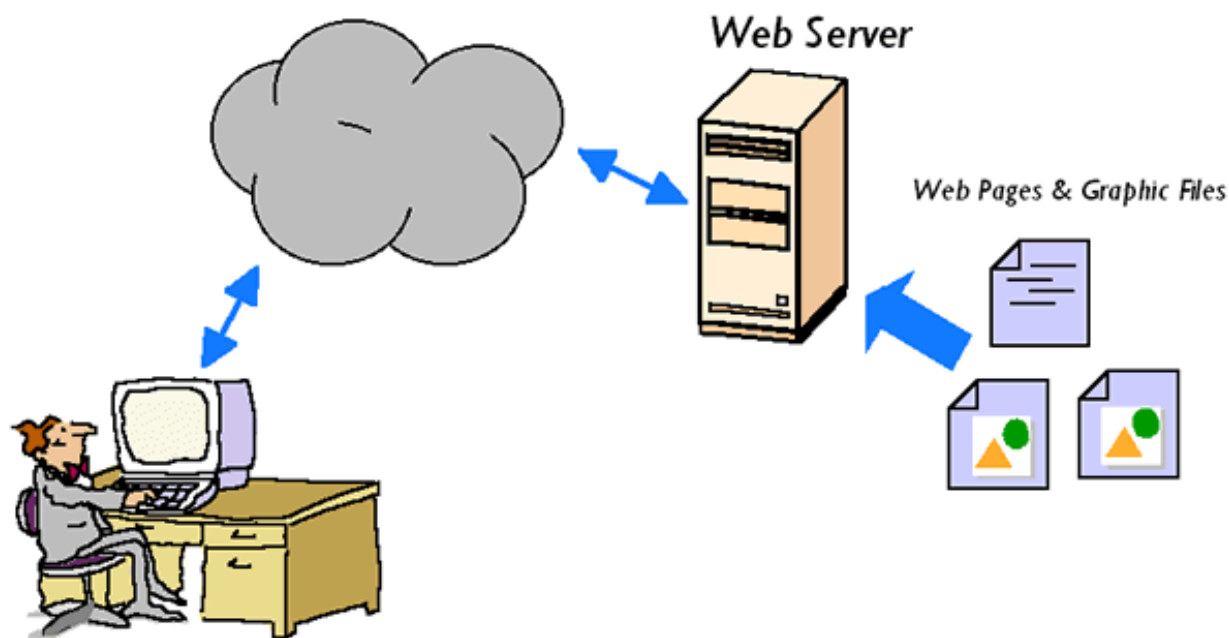


Model-View-Controller 設計模式



網頁伺服器 (Web Server)

- 是一個軟體
 - ✓ 回應從 80/8080 port 進來的 HTTP 要求
 - ✓ 可透過 CGI 或 module 方式擴充
 - ✓ 如 Apache, Nginx, Boa





A Python Microframework

- 目前網路程式架構很多，例如：Django、Flask、Pyramid、Bottle、Tornado 等等。其中 Flask 是發佈於 2010 年的年輕網路架構，吸收其他框架的優點而建立。
- 網站微框架（microframework），卻兼具核心程式碼簡潔以及強大的擴充能力。除此之外，Flask 也不會替開發者做任何技術決定，不會替您決定該使用哪種資料庫，但提供各種擴充套件協助整合各種不同資料庫。

一個測試訊息

- 在沒有文件上，測試直接運行的訊息。

7-1

測試直接運行的訊息。(7-1-app-hello.py)

```
from flask import Flask

app = Flask(__name__) # flask 才會知道你的 root 在何處

@app.route("/") ← Controller # 定義路由路徑，/ 指的是 web root (URL)
def index(): # 一個接著要執行的 function
    return "Hello Flask" ← View

if __name__ == "__main__": # 是直接運行的，才會執行 app.run() 這個方法
    app.run(host = '0.0.0.0', port = 80, debug = True)
```

Demo

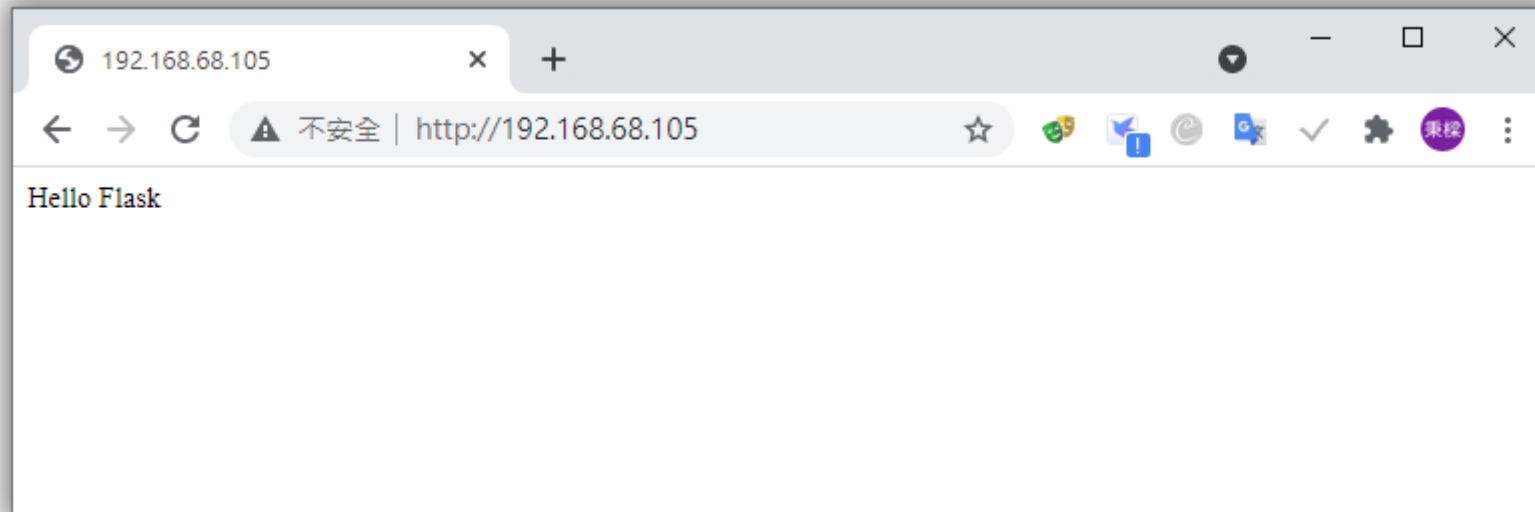
7-1-app-hello.py

```
$ cd ~/Lab/ch07
```

```
$ sudo python3 7-1-app-hello.py
```


Demo 7-1-app-hello.py

```
pi@raspberrypi: ~/code/exper6-2
pi@raspberrypi:~/code/exper6-2 $ sudo python3 Lab6-1-app-hello.py
* Serving Flask app "Lab6-1-app-hello" (lazy loading)
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://0.0.0.0:80/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 221-014-888
```



若發生 Address 被占用，無法執行時

ex7-5

```
pi@raspberrypi: ~/code/exper6-2
* Environment: production
  WARNING: Do not use the development server in a production environment.
  Use a production WSGI server instead.
* Debug mode: on
Traceback (most recent call last):
  File "Lab6-1-app-hello.py", line 23, in <module>
    app.run(host='0.0.0.0', port=80, debug=True)
  File "/usr/lib/python3/dist-packages/flask/app.py", line 943, in run
    run_simple(host, port, self, **options)
  File "/usr/lib/python3/dist-packages/werkzeug/serving.py", line 795, in run_simple
    s.bind(get_sockaddr(hostname, port, address_family))
OSError: [Errno 98] Address already in use
pi@raspberrypi:~/code/exper6-2 $
```

- 查找哪一個 python 程序正在執行中

```
$ ps -fA | grep python
```

- 刪除 root 占用的程序（需自行找出哪一個 PID）

```
$ sudo kill -9 PID （執行中的 python 是 root 的程序，所以加 sudo）
```

樣板引擎 (Template Engine)

動態網頁 (1/3)

Yahoo! 股市

登入 信箱

搜尋台股股代號/名稱 台股代號查詢 距離台股收盤還有02小時56分鐘 更新

新戶滿額送購物金 1000元

長榮 2603 航運 ☆ 加入自選股

132.0 ▲ 8.0 (6.45%)

開盤 | 2021/06/15 10:32 更新

368,285 成交量 10.63 (47.49) 本益比(同業平均) 跌→漲 (▲ 9.25%) 連漲連跌

走勢圖 技術分析 成交彙整 籌碼 股利 財務 健診 基本 新聞 意見回饋

長榮 110/06/15 10:32

開盤	127.0	漲幅	6.45%
買價	131.5	振幅	6.45%
賣價	132.0	最高	133.5
成交	132.0	最低	125.5
單量	2	漲跌	8.0
總量	368288	均價	130.0
昨量	556662	金額	478.9

委買價/量	委賣價/量
131.5 / 1059	132.0 / 1646
131.0 / 1883	132.5 / 5181
130.5 / 1579	133.0 / 11.7K
130.0 / 3546	133.5 / 5878
129.5 / 2710	134.0 / 5496

委買賣差	-19111
委買賣比	0.36

©Yahoo奇摩股市

相關權證

- 長榮統一12購03
- 長榮統一0C購14
- 長榮群益0C購08
- 長榮元大0B購20
- 長榮凱基0B售07
- 長榮統一11購07
- 長榮凱基11購05
- 長榮元大0B售09
- 長榮元大0B售05
- 長榮元富0B售06
- 長榮凱基14購06
- 長榮永豐0A購11

我的自選股

立即登入瀏覽你的投資組合

登入

網友也在看

股名/股號	股價	漲跌(%)	成交量(張)
陽明 2609.TW	123.00	▲ 7.00 6.03%	268,956
萬海 2615.TW	191.50	▲ 17.00 9.74%	34,872
長榮航	20.15	▼ 0.05	59,342

動態網頁 (2/3)

群創(3481.TW) 走勢圖 - Yahoo! 股市

https://tw.stock.yahoo.com/q/bc?s=3481

登入 信箱

搜尋台股股代號/名稱 台股代號查詢 距離台股收盤還有02小時58分鐘 更新

【TOSHIBA】12KG變頻滾筒洗衣機
#沖浪洗淨X極淨淨化 #洗劑速溶渦輪技術 #全級省水標準
TOSHIBA 新機 X MOMO獨家

群創 3481 光電 ☆ 加入自選股

20.65 ▼0.70 (3.28%)
開盤 | 2021/06/15 10:30 更新

227,701 成交量 11.86 (62.56) 本益比 (同業平均) 連3跌→漲 (▲ 0.95%) 連漲連跌

走勢圖 技術分析 成交彙整 籌碼 股利 財務 健診 基本 新聞 意見回饋

群創 110/06/15 10:30

開盤	20.85	漲幅	-3.28%
買價	20.65	振幅	3.28%
賣價	20.70	最高	21.20
成交	20.65	最低	20.50
單量	5	漲跌	-0.70
總量	227618	均價	20.77
昨量	329912	金額	47.29

委買價/量	委賣價/量
20.65 / 1087	20.70 / 1169
20.60 / 2486	20.75 / 2142
20.55 / 5332	20.80 / 2865
20.50 / 16.6K	20.85 / 2068
20.45 / 5723	20.90 / 3716

委買賣差	19225
委買賣比	2.61

相關權證

- 群創統一0C購11
- 群創元六12購04
- 群創統一15購03
- 群創群益12購03
- 群創統一0B購05
- 群創統一14購01
- 群創永豐0A購10
- 群創永豐0B購01
- 群創凱基0A舊08
- 群創泰證0C購01
- 群創統一0B購13
- 群創統一0B購04

齊天大聖 最可愛的陶藝擺飾 【特價再送金箍棒】 680元

我的自選股

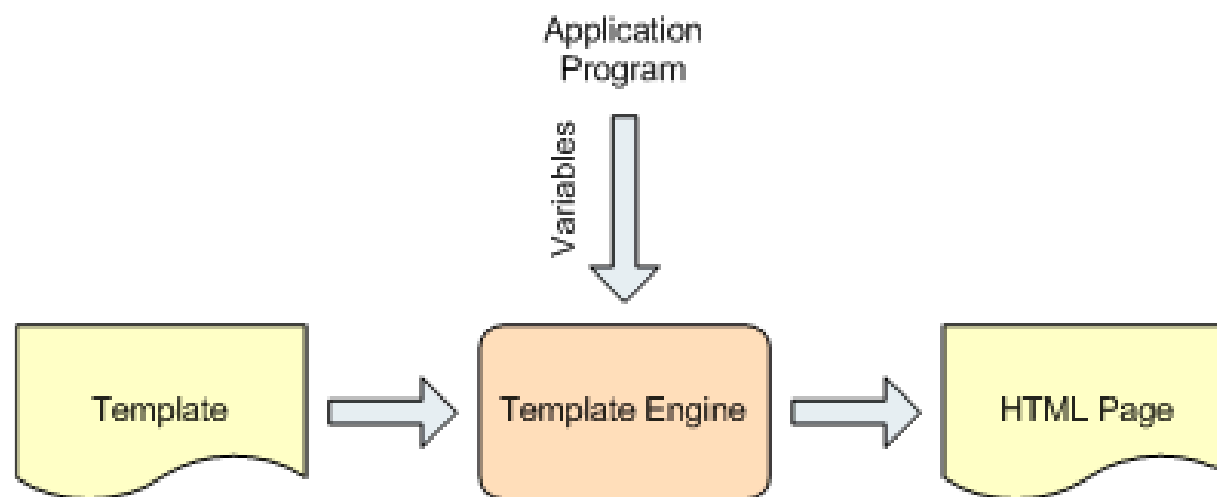
立即登入瀏覽你的投資組合 · 登入

網友也在看

股名/股號	股價	漲跌(%)	成交量(張)
友達 2409.TW	22.40	▼ 0.65 2.82%	147,718
彩晶 6116.TW	22.05	▼ 0.70 3.08%	66,793
聯華	53.30	▲ 1.50	112,767

動態網頁 (3/3)

- 是由靜態 HTML 加上動態文字產生
- 樣板引擎將產生動態的程式碼與使用者介面分離
- Flask 預設使用 Jinja 做為樣板引擎



新增一個 route & template

ex7-6

7-2 route & template ◦ (7-2-app-route.py)

```
from flask import Flask, render_template, Response
```

```
app = Flask(__name__)
```

```
@app.route("/")
```

```
def index():
```

```
    return render_template('link.html') ← View 可由自訂的樣板產生
```

```
@app.route("/foo")
```

```
def foo():
```

```
    extns = ['Flask', 'Jinja2', 'Awesome'] ← 樣板內可塞變數
```

```
    return render_template('bar.html', extns = extns)
```

```
if __name__ == "__main__":
```

```
    app.run(host = '0.0.0.0', port = 80, debug = True)
```

建立 template

ex7-7

```
$ mkdir templates
```

```
$ nano templates/link.html
```

link.html

```
<h1>Hello Template</h1>
<a href="{{ url_for('foo') }}">foo</a>
```

```
$ nano templates/bar.html
```

bar.html

```
<ul>
{% for ext in extns %}
<li>{{ ext }}</li>
{% endfor %}
</ul>
```

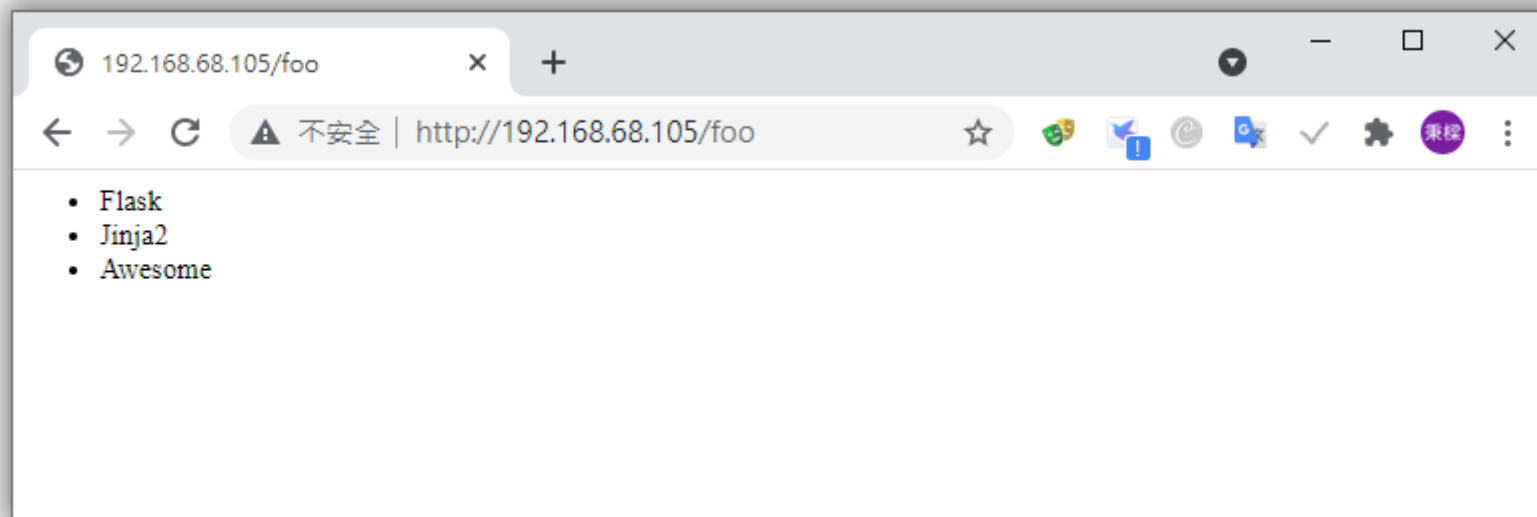
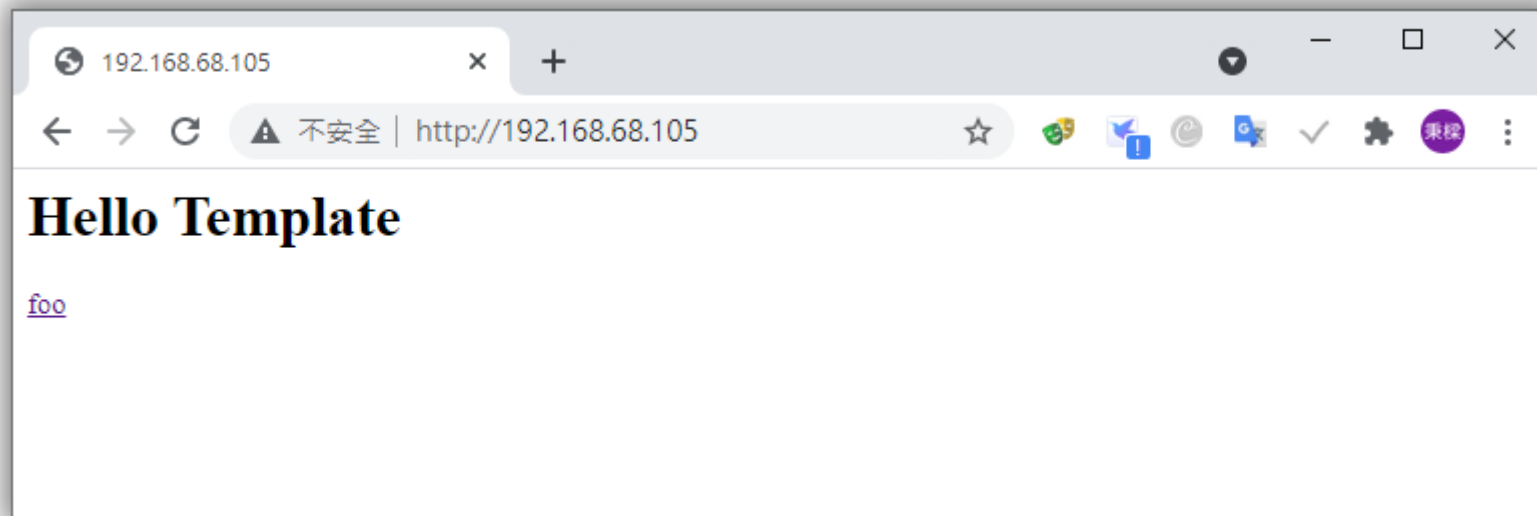

Demo

7-2-app-route.py

```
$ cd ~/Lab/ch07
```

```
$ sudo python3 7-2-app-route.py
```

Demo 7-2-app-route.py



Streaming 圖片

建立 Camera 類別 (模組化)

ex7-8

- 自訂副函式 (function)、類別 (class)、模組 (module)，完成模組化的好處。

自訂類別 自訂一個模組，裡頭只有一個 Camera 類別。(stream_pi.py)

```
from time import time

class Camera(object):
    def __init__(self):
        self.frames = [open("static/" + f + '.jpg', 'rb').read()
                       for f in ['1', '2', '3']]

    def get_frame(self):
        return self.frames[int(time()) % 3]
```

7-3

串流一連串圖片。(7-3-app-stream.py)

```
from flask import Flask, render_template, Response
from stream_pi import Camera ← 匯入自訂模組 stream_pi

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('stream.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(Camera()),
                   mimetype = 'multipart/x-mixed-replace; boundary=frame')

if __name__ == "__main__":
    app.run(host = '0.0.0.0', port = 80, debug = True)
```

Camera 類別

回傳內容

修改 templates，新增一個網頁檔

```
$ nano templates/stream.html
```

```
stream.html
```

```
<h1>Hello Stream</h1>  

```

- 圖檔已建立好：1.jpg, 2.jpg, 3.jpg

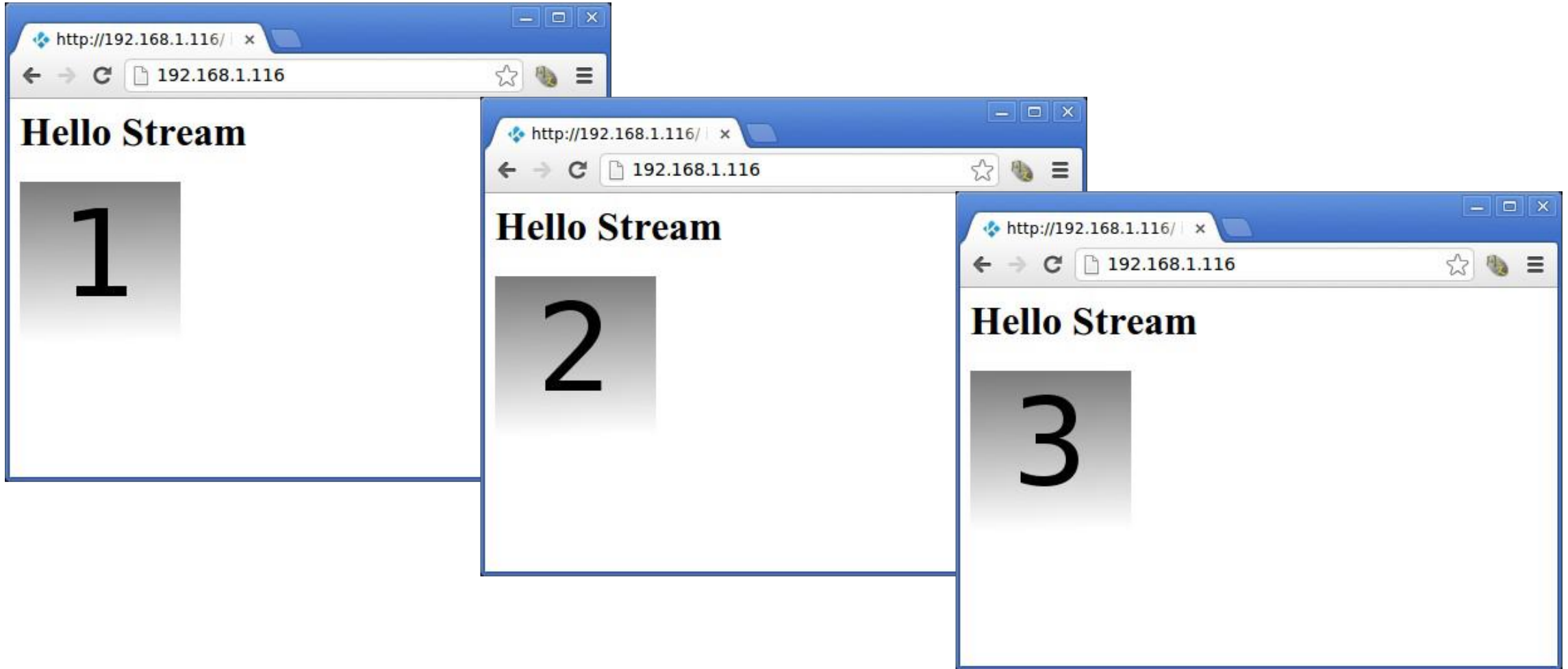
Demo

7-3-app-stream.py

```
$ cd ~/Lab/ch07
```

```
$ sudo python3 7-3-app-stream.py
```

Demo 7-3-app-stream.py



Streaming 即時影像

HTTP + MJPEG

- MJPEG = Motion JPEG
 - ✓ 一種視訊壓縮格式
 - ✓ 每一個 frame 都使用 JPEG 編碼
 - ✓ 對運算能力與記憶體的需求較低

建立 Camera 類別 (從 Camera 讀取影像)

自訂類別 自訂一個模組，Camera 類別用來讀取影像。(camera_pi.py)

```
import cv2

class Camera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        success, image = self.video.read()
        ret, jpeg = cv2.imencode('.jpg', image)
        return jpeg.tostring()
```

開啟 /dev/videoX，預設的 camera 是 0

V4L2 API

7-4

串流即時影像。(7-4-app-camera.py)

```
from flask import Flask, render_template, Response
from camera_pi import Camera

app = Flask(__name__)

@app.route('/')
def index():
    return render_template('stream.html')

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
              b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(Camera()),
                    mimetype = 'multipart/x-mixed-replace; boundary = frame')

if __name__ == "__main__":
    app.run(host = '0.0.0.0', port = 80, debug = True)
```

Demo

7-4-app-camera.py

```
$ cd ~/Lab/ch07
```

```
$ sudo python3 7-4-app-camera.py
```

如何降低延遲？

自訂類別 改進的 camera_pi.py

```
import cv2

class Camera(object):
    def __init__(self):
        self.video = cv2.VideoCapture(0)
        self.video.set(PROV_FRAME_WIDTH, 320)
        self.video.set(PROV_FRAME_HEIGHT, 240)

    def __del__(self):
        self.video.release()

    def get_frame(self):
        success, image = self.video.read()
        ret, jpeg = cv2.imencode('.jpg', image, [1, 50])
        return jpeg.tostring()
```

← 改變視窗的寬跟高

← 調整 JPG 壓縮品質

小結

- 串流服務可使用 RTSP + H.264，或是 HTTP + MJPG 等多種組合。
- 使用自己寫的 HTTP 串流，可以在 JPG 丟出去前再行處理。

整合專案：加入 OpenCV 人臉辨識

Demo

app-opencv.py

```
$ cd ~/Lab/ch07/project_opencv
```

```
$ sudo python3 app-opencv.py
```

整合專案：加入 LINE 推播

Demo

app-camera.py

```
$ cd ~/Lab/ch07/project_opencv+line
```

```
$ sudo python3 app-opencv+line.py
```