Python 程式教學-chapter 30

- 執行緒的使用方法
- 多執行緒下載檔案

陳岳洋(Jerry老師)

- NBA爬蟲多執行緒執行效率
- 爬取資料:2019/10/1~2019/10/31

```
36, 29], ['2019-10-27', '鵜鶘', 24, 37, 39, 23], ['2019-10-27', '馬刺',
30, 30, 32, 32], ['2019-10-27', '巫師', 26, 34, 30, 32], ['2019-10-27',
'爵士', 35, 31, 27, 20], ['2019-10-27', '國王', 25, 16, 17, 23],
['2019-10-27', '太陽', 26, 29, 36, 39], ['2019-10-27', '快艇', 22, 33,
30, 37], ['2019-10-28', '雷霆', 35, 35, 35, 15], ['2019-10-28', '勇士',
20, 17, 31, 24], ['2019-10-28', '灰熊', 35, 24, 33, 28], ['2019-10-28',
'籃網', 29, 28, 40, 23], ['2019-10-28', '獨行俠', 40, 31, 24, 24],
['2019-10-28', '拓荒者', 25, 34, 36, 26], ['2019-10-28', '灰狼', 36,
19, 22, 39], ['2019-10-28', '熱火', 23, 34, 27, 25], ['2019-10-28', '湖
人', 30, 33, 24, 33], ['2019-10-28', '黃蜂', 28, 34, 18, 21],
['2019-10-29', '活塞', 33, 26, 19, 18], ['2019-10-29', '溜馬', 25, 28,
18, 23], ['2019-10-29', '尼克', 15, 27, 30, 33], ['2019-10-29', '公牛',
33, 19, 28, 18], ['2019-10-29', '老鷹', 40, 25, 18, 20], ['2019-10-29',
'76人', 31, 32, 19, 23], ['2019-10-29', '暴龍', 31, 20, 27, 26],
['2019-10-29', '魔術', 22, 24, 21, 28], ['2019-10-29', '火箭', 22, 30,
39, 25], ['2019-10-29', '雷霆', 35, 27, 18, 32], ['2019-10-29', '公鹿',
30, 29, 32, 38], ['2019-10-29', '騎士', 31, 21, 28, 32], ['2019-10-29',
'鵜鶘', 23, 32, 24, 44], ['2019-10-29', '勇士', 27, 45, 31, 31],
['2019-10-29', '馬刺', 19, 26, 37, 31], ['2019-10-29', '拓荒者', 33,
18, 20, 39], ['2019-10-29', '太陽', 21, 18, 31, 25], ['2019-10-29', '爵
士', 28, 18, 24, 26], ['2019-10-29', '國王', 28, 25, 17, 24],
['2019-10-29', '金塊', 21, 26, 30, 24], ['2019-10-29', '快艇', 29, 28,
29, 25], ['2019-10-29', '黃蜂', 28, 26, 20, 22], ['2019-10-30', '熱火',
29, 30, 29, 24], ['2019-10-30', '老鷹', 26, 23, 21, 27], ['2019-10-30',
'金塊', 31, 30, 27, 18], ['2019-10-30', '獨行俠', 27, 33, 23, 26],
['2019-10-30', '湖人', 27, 22, 39, 32], ['2019-10-30', '灰熊', 32, 15,
20, 24]]
358.9296329021454
3040
```

單執行緒版本耗時
359秒

nba_single_thread.py

- 多執行緒版本爬取速度快四倍！

```
['2019-10-27', '魔術', 26, 24, 25, 24], ['2019-10-27', '尼克', 24, 27,
25, 19], ['2019-10-27', '塞爾蒂克', 22, 24, 36, 36], ['2019-10-27', '騎
士', 26, 39, 25, 20], ['2019-10-27', '溜馬', 28, 20, 26, 25],
['2019-10-27', '公牛', 23, 17, 22, 22], ['2019-10-27', '暴龍', 24, 24,
36, 24], ['2019-10-27', '火箭', 29, 32, 36, 29], ['2019-10-27', '鵜鶘',
24, 37, 39, 23], ['2019-10-27', '馬刺', 30, 30, 32, 32], ['2019-10-27',
'巫師', 26, 34, 30, 32], ['2019-10-27', '爵士', 35, 31, 27, 20],
['2019-10-27', '國王', 25, 16, 17, 23], ['2019-10-27', '太陽', 26, 29,
36, 39], ['2019-10-27', '快艇', 22, 33, 30, 37], ['2019-10-29', '活塞',
33, 26, 19, 18], ['2019-10-29', '溜馬', 25, 28, 18, 23], ['2019-10-29',
'尼克', 15, 27, 30, 33], ['2019-10-29', '公牛', 33, 19, 28, 18],
['2019-10-29', '老鷹', 40, 25, 18, 20], ['2019-10-29', '76人', 31, 32,
19, 23], ['2019-10-29', '暴龍', 31, 20, 27, 26], ['2019-10-29', '魔術',
22, 24, 21, 28], ['2019-10-29', '火箭', 22, 30, 39, 25], ['2019-10-29',
'雷霆', 35, 27, 18, 32], ['2019-10-29', '公鹿', 30, 29, 32, 38],
['2019-10-29', '騎士', 31, 21, 28, 32], ['2019-10-29', '鵜鶘', 23, 32,
24, 44], ['2019-10-29', '勇士', 27, 45, 31, 31], ['2019-10-29', '馬刺',
19, 26, 37, 31], ['2019-10-29', '拓荒者', 33, 18, 20, 39],
['2019-10-29', '太陽', 21, 18, 31, 25], ['2019-10-29', '爵士', 28, 18,
24, 26], ['2019-10-29', '國王', 28, 25, 17, 24], ['2019-10-29', '金塊',
21, 26, 30, 24], ['2019-10-29', '快艇', 29, 28, 29, 25], ['2019-10-29',
'黃蜂', 28, 26, 20, 22], ['2019-10-30', '熱火', 29, 30, 29, 24],
['2019-10-30', '老鷹', 26, 23, 21, 27], ['2019-10-30', '金塊', 31, 30,
27, 18], ['2019-10-30', '獨行俠', 27, 33, 23, 26], ['2019-10-30', '湖
人', 27, 22, 39, 32], ['2019-10-30', '灰熊', 32, 15, 20, 24]]
87.61120748519897
3040
```

多執行緒版本耗時
88秒

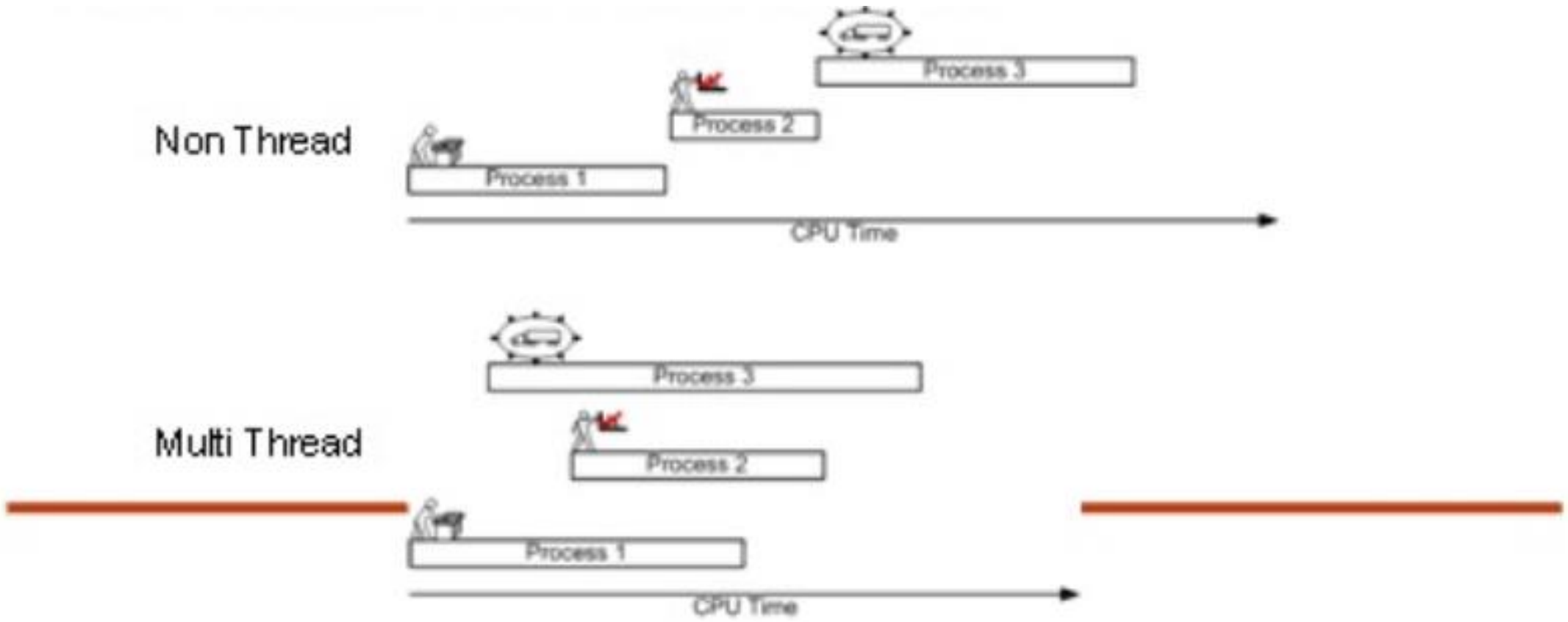
Nba_mulit_thread.py

多執行緒

- 優點；提高程式的執行效率、提高CPU和記憶體的利用率、增加執行效率。

- 缺點：每開啟一條執行緒就會佔用一定記憶體空間，降低程式效能；執行緒越多，CPU排程（多個執行緒之間切換）開銷越大，時間開銷、空間開銷。

- 多執行緒 vs 單執行緒


Non Thread
Process 1
Process 2
Process 3
CPU Time
Multi Thread
Process 1
Process 2
Process 3
CPU Time

範例程式

```python
import time

def f1():
    print('f1 start!')
    for i in range(10):
        time.sleep(0.1)
    print('f1 finish!')


def f2():
    print('f2 start!')
    for i in range(20):
        time.sleep(0.1)
    print('f2 finish!')

def f3():
    print('f3 start!')
    for i in range(30):
        time.sleep(0.1)
    print('f3 finish!')

f1()
f2()
f3()
print('all done!')
```

單執行緒依序執行!

```
f1 start!
f1 finish!
f2 start!
f2 finish!
f3 start!
f3 finish!
all done!
```

ch30_1.py

- 使用多執行緒方式
  - import threading

```
t=threading.Thread(target=函式,args=(,))
```

函式需要的參數

將要執行的動作
宣告成函示

```
t.start()
```

啟動

- 讓t1, t2, t3同時間運行減省時間

```
threads=[]

t1=threading.Thread(target=f1)
t1.start()
threads.append(t1)
t2=threading.Thread(target=f2)
t2.start()
threads.append(t2)
t3=threading.Thread(target=f3)
t3.start()
threads.append(t3)

print('all down!')
```

改由執行緒執行

```
f1 start!
f2 start!
f3 start!all down!

f1 finish!
f2 finish!
f3 finish!
```

ch30_2.py

- 將多執行緒併入主執行緒

```
for thread in threads:
    thread.join()
```

讓主程式等待執行緒都完成後再繼續下面程式碼

```
print('all down!')
```

```
f1 start!f2 start!

f3 start!
f1 finish!
f2 finish!
f3 finish!
all down!
```

```python
import threading
import time

def f1():
    print('f1 start!')
    for i in range(10):
        time.sleep(0.1)
    print('f1 finish!')


def f2():
    print('f2 start!')
    for i in range(20):
        time.sleep(0.1)
    print('f2 finish!')

def f3():
    print('f3 start!')
    for i in range(30):
        time.sleep(0.1)
    print('f3 finish!')

threads=[]

t1=threading.Thread(target=f1)
t1.start()
threads.append(t1)
t2=threading.Thread(target=f2)
t2.start()
threads.append(t2)
t3=threading.Thread(target=f3)
t3.start()
threads.append(t3)


for thread in threads:
    thread.join()

print('all down!')
```

```python
t=threading.Thread(target=f1)
threads.append(t)
t=threading.Thread(target=f2)
threads.append(t)
t=threading.Thread(target=f3)
threads.append(t)

for thread in threads:
    thread.start()

for thread in threads:
    thread.join()
```

```
f1 start!f2 start!

f3 start!
f1 finish!
f2 finish!
f3 finish!
all down!
```

ch30_3.py

- Thread運用在有返回值的函式

```
def job(data):
    for i in range(len(data)):
        data[i]=data[i]**2

    return data
```

有返回值無法直接使用
threading.Thread(target=job,args=(,))

ch30_5.py

■ Queue模組

Python中，佇列（First In First Out）是執行緒間最常用的交換數據的形式。Queue **模組** 是提供佇列操作的**模組**。

```
from queue import Queue

q = Queue()

q.put(10)
q.put(11)
q.put(12)
q.put(13)

print(q.qsize())

for  i in range(q.qsize()):
    print(q.get())
```

將結果暫存至Queue(佇列)

```
4
10
11
12
13
```

ch30_6.py

- 改寫ch30_5.py(使用Queue模組)

from queue import Queue

```
from queue import Queue

#傳入串列將每個值平方
def job(data,q):
    for i in range(len(data)):
        data[i]=data[i]**2

    return q.put(data)
```

傳入Queue物件(q)

將結果先置入q

主程式結合queue

```
datas=[[1,2,3,4],[5,5,5],list(range(10))]
threads=[]
#產生空佇列
q=Queue()

for data in datas:
    t=threading.Thread(target=job,args=(data,q))
    t.start()
    threads.append(t)

#合併在主線程
for thread in threads:
    thread.join()

#取得回傳值
result=[]
for thread in threads:
    result.append(q.get())

print(result)
print('done!')
```

將data跟q當參數傳入

將queue中的資料進行取得!

```python
import threading

from queue import Queue

#傳入串列將每個值平方
def job(data,q):
    for i in range(len(data)):
        data[i]=data[i]**2

    return q.put(data)

datas=[[1,2,3,4],[5,5,5],list(range(10))]
threads=[]
#產生空佇列
q=Queue()


for data in datas:
    t=threading.Thread(target=job,args=(data,q))
    t.start()
    threads.append(t)

#合併在主線程
for thread in threads:
    thread.join()

#取得回傳值
result=[]
for thread in threads:
    result.append(q.get())

print(result)
print('done!')
```

```
[[1, 4, 9, 16], [25, 25, 25], [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]]
done!
```

ch30_7.py

速度?

```python
#傳入串列將每個值平方
def job1(data):
    for i in range(len(data)):
        data[i]=data[i]**2

    return data

datas=[[1,2,3,4],[5,5,5],list(range(10))]

print('單執行緒')
st=time.time()
result=[]
for data in datas:
    result.append(job1(data))

print(result)
print(time.time()-st)
print('done!')
```

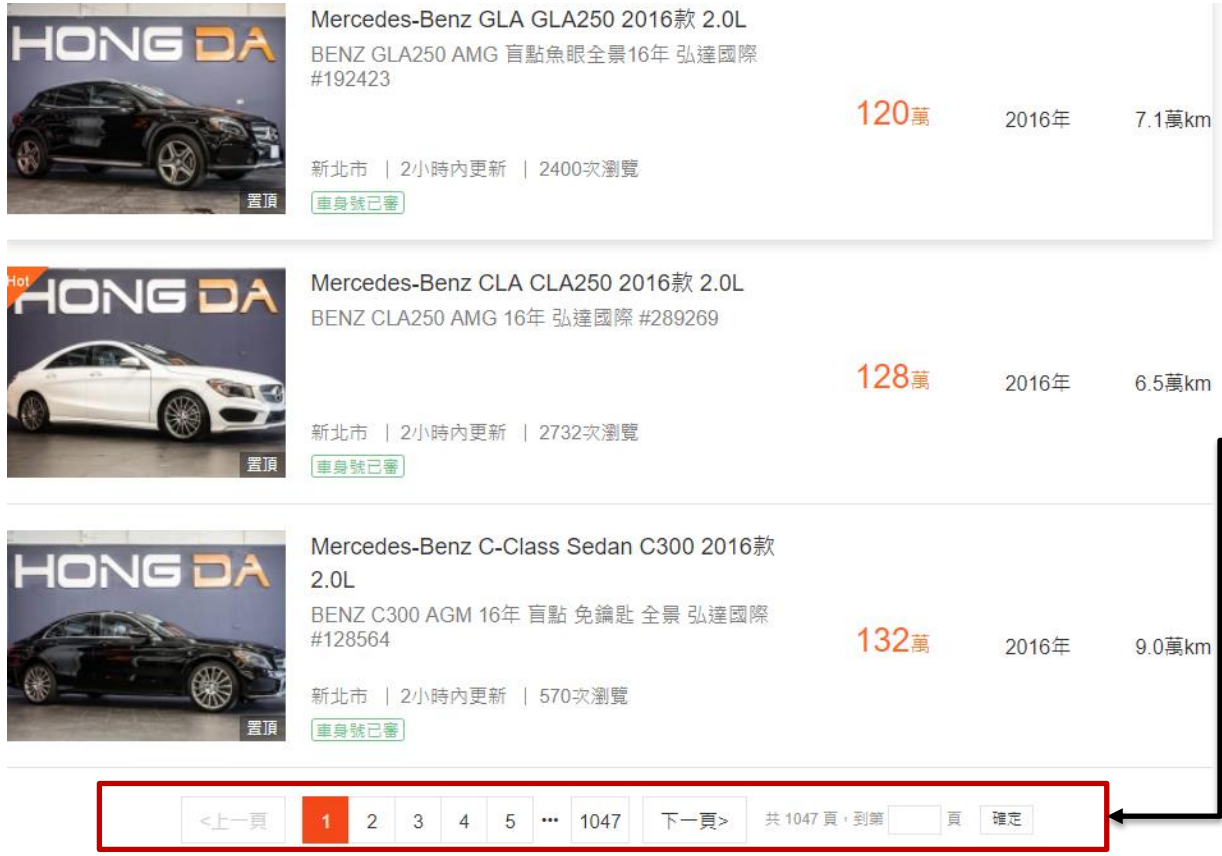
一般或少量運算使用多執行緒有可能反而比較慢!!

```
單執行緒
[[1, 4, 9, 16], [25, 25, 25], [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]]
0.0
done!
多執行緒
[[1, 4, 9, 16], [25, 25, 25], [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]]
0.0009899139404296875
done!
```

ch30_8.py

運用在網頁資料分析上

- 先取得主頁頁數連結(使用requests, BS4套件)
- 再使用Threading進行分頁資料的多執行緒抓取



PAGE.1

PAGE.2

PAGE.3

PAGE.4

PAGE.5

PAGE.6

PAGE.7

PAGE.8

運用在網頁檔案連結下載上

- 先取得所有連結(使用requests, BS4套件)
- 再使用Threading進行連結資料的下載

抓取(json文檔, 圖片，影片，檔案等等)

取得所有URL
開始多執行緒下載

- 應用在網頁爬蟲搜尋

```
def getData(url):
    print('抓取:'+url)
    resp=requests.get(url)
    if resp.status_code==200:
        print(resp)
    #time.sleep(2)

urls=['https://www.google.com','https://www.yahoo.com.tw']
```

- 多執行緒比較快完成！

```python
st=time.time()
for url in urls:
    getData(url)

print('done!')
time.time()-st
```

```
抓取:https://www.google.com
<Response [200]>
抓取:https://www.yahoo.com.tw
<Response [200]>
done!

1.5685021877288818
```

```python
import threading

threads=[]
st=time.time()
for url in urls:
    t = threading.Thread(target=getData,args=(url,))
    t.start()
    threads.append(t)


for t in threads:
    t.join()

print('done!')
time.time()-st
```

```
抓取:https://www.google.com
抓取:https://www.yahoo.com.tw
<Response [200]>
<Response [200]>
done!

0.4020719528198242
```

ch30_9.py

- 如何修改?
- 請將nba_single_thread.py 進行多執行緒修改
- 測試2019/10/1~2019/10/31

```
'鵜鶘', 24, 37, 39, 23], ['2019-10-27', '馬刺', 30, 30, 32, 32],
['2019-10-27', '巫師', 26, 34, 30, 32], ['2019-10-27', '爵士', 35, 31,
27, 20], ['2019-10-27', '國王', 25, 16, 17, 23], ['2019-10-27', '太陽',
26, 29, 36, 39], ['2019-10-27', '快艇', 22, 33, 30, 37], ['2019-10-28',
'雷霆', 35, 35, 35, 15], ['2019-10-28', '勇士', 20, 17, 31, 24],
['2019-10-28', '灰熊', 35, 24, 33, 28], ['2019-10-28', '籃網', 29, 28,
40, 23], ['2019-10-28', '獨行俠', 40, 31, 24, 24], ['2019-10-28', '拓荒
者', 25, 34, 36, 26], ['2019-10-28', '灰狼', 36, 19, 22, 39],
['2019-10-28', '熱火', 23, 34, 27, 25], ['2019-10-28', '湖人', 30, 33,
24, 33], ['2019-10-28', '黃蜂', 28, 34, 18, 21], ['2019-10-29', '活塞',
33, 26, 19, 18], ['2019-10-29', '溜馬', 25, 28, 18, 23], ['2019-10-29',
'尼克', 15, 27, 30, 33], ['2019-10-29', '公牛', 33, 19, 28, 18],
['2019-10-29', '老鷹', 40, 25, 18, 20], ['2019-10-29', '76人', 31, 32,
19, 23], ['2019-10-29', '暴龍', 31, 20, 27, 26], ['2019-10-29', '魔術',
22, 24, 21, 28], ['2019-10-29', '火箭', 22, 30, 39, 25], ['2019-10-29',
'雷霆', 35, 27, 18, 32], ['2019-10-29', '公鹿', 30, 29, 32, 38],
['2019-10-29', '騎士', 31, 21, 28, 32], ['2019-10-29', '鵜鶘', 23, 32,
24, 44], ['2019-10-29', '勇士', 27, 45, 31, 31], ['2019-10-29', '馬刺',
19, 26, 37, 31], ['2019-10-29', '拓荒者', 33, 18, 20, 39],
['2019-10-29', '太陽', 21, 18, 31, 25], ['2019-10-29', '爵士', 28, 18,
24, 26], ['2019-10-29', '國王', 28, 25, 17, 24], ['2019-10-29', '金塊',
21, 26, 30, 24], ['2019-10-29', '快艇', 29, 28, 29, 25], ['2019-10-29',
'黃蜂', 28, 26, 20, 22], ['2019-10-30', '熱火', 29, 30, 29, 24],
['2019-10-30', '老鷹', 26, 23, 21, 27], ['2019-10-30', '金塊', 31, 30,
27, 18], ['2019-10-30', '獨行俠', 27, 33, 23, 26], ['2019-10-30', '湖
人', 27, 22, 39, 32], ['2019-10-30', '灰熊', 32, 15, 20, 24]]
19.68757462501526
266
```

單執行緒版本(19秒多)

nba_single_thread.py

修改一版

```
threads=[]
q = Queue()
st=time.time()


while start_day<end_day:
    print('擷取日期:%s'%(start_day.strftime("%Y-%m-%d")))
    t1=threading.Thread(target=getNBAData,args=(api_url,start_day.strftime("%Y-%m-%d"),q))
    t1.start()
    threads.append(t1)
    start_day+=datetime.timedelta(days=1)

    #print(q.qsize())
    #time.sleep(1)

for thread in threads:
    thread.join()
print(q.qsize())

for i in range(q.qsize()):
    list_data+=q.get()

print(list_data)
print(time.time()-st)
print(len(list_data))
```

```
21, 24, 17, 33], ['2019-10-26', '國王', 27, 30, 27, 28], ['2019-10-26',
'拓荒者', 25, 32, 37, 28], ['2019-10-26', '湖人', 24, 19, 31, 21],
['2019-10-26', '爵士', 17, 20, 18, 31], ['2019-10-24', '黃蜂', 37, 26,
33, 30], ['2019-10-24', '公牛', 28, 27, 40, 30], ['2019-10-24', '溜馬',
24, 31, 31, 24], ['2019-10-24', '活塞', 27, 27, 29, 36], ['2019-10-24',
'魔術', 28, 27, 16, 23], ['2019-10-24', '騎士', 24, 17, 24, 20],
['2019-10-24', '籃網', 22, 34, 37, 22], ['2019-10-24', '灰狼', 33, 35,
20, 27], ['2019-10-24', '熱火', 24, 30, 29, 37], ['2019-10-24', '灰熊',
32, 28, 24, 17], ['2019-10-24', '76人', 20, 29, 28, 30], ['2019-10-24',
'塞爾蒂克', 25, 23, 20, 25], ['2019-10-24', '獨行俠', 24, 38, 29, 17],
['2019-10-24', '巫師', 25, 23, 20, 32], ['2019-10-24', '馬刺', 22, 37,
24, 37], ['2019-10-24', '尼克', 15, 36, 33, 27], ['2019-10-24', '爵士',
23, 26, 19, 32], ['2019-10-24', '雷霆', 12, 34, 28, 21], ['2019-10-24',
'太陽', 25, 29, 32, 38], ['2019-10-24', '國王', 29, 30, 17, 19],
['2019-10-24', '拓荒者', 27, 23, 24, 26], ['2019-10-24', '金塊', 24,
30, 19, 35], ['2019-10-09', '76人', 36, 46, 35, 27], ['2019-10-09',
'Long-Lions', 22, 19, 20, 25], ['2019-10-09', '熱火', 22, 31, 28, 26],
['2019-10-09', '馬刺', 23, 23, 18, 25], ['2019-10-09', '灰熊', 27, 28,
22, 31], ['2019-10-09', 'Breakers', 14, 22, 29, 29], ['2019-10-09', '雷
霆', 33, 33, 31, 22], ['2019-10-09', '獨行俠', 28, 34, 27, 15],
['2019-10-09', '太陽', 37, 19, 34, 21], ['2019-10-09', '灰狼', 26, 28,
28, 24], ['2019-10-09', '拓荒者', 22, 28, 18, 26], ['2019-10-09', '金
塊', 27, 25, 26, 27]]
2.741436004638672
266
```

縮減到2秒多!!
但進行長周期抓取，
會發生問題!!!

修改二版(設定同時最大執行執行緒)

```python
threads=[]
q = Queue()
st=time.time()

count=0
while start_day<end_day:
    print('擷取日期:%s'%(start_day.strftime("%Y-%m-%d")))
    t1=threading.Thread(target=getNBAData,args=(api_url,start_day.strftime("%Y-%m-%d"),q))
    t1.start()
    threads.append(t1)
    start_day+=datetime.timedelta(days=1)

    count+=1

    if count%5==0:
        for thread in threads:
            thread.join()
        for i in range(q.qsize()):
            list_data+=q.get()
        threads=[]
        count=0


for thread in threads:
    thread.join()
print(q.qsize())

for i in range(q.qsize()):
    list_data+=q.get()

print(list_data)
print(time.time()-st)
print(len(list_data))
```

限制最多同時10個執行緒

```
['2019-10-27', '老鷹', 23, 29, 25, 26], ['2019-10-27', '魔術', 26, 24, 25, 24], ['2019-10-27', '尼克', 24, 27, 25, 19], ['2019-10-27', '塞爾蒂克', 22, 24, 36, 36], ['2019-10-27', '騎士', 26, 39, 25, 20], ['2019-10-27', '溜馬', 28, 20, 26, 25], ['2019-10-27', '公牛', 23, 17, 22, 22], ['2019-10-27', '暴龍', 24, 24, 36, 24], ['2019-10-27', '火箭', 29, 32, 36, 29], ['2019-10-27', '鵜鶘', 24, 37, 39, 23], ['2019-10-27', '馬刺', 30, 30, 32, 32], ['2019-10-27', '巫師', 26, 34, 30, 32], ['2019-10-27', '爵士', 35, 31, 27, 20], ['2019-10-27', '國王', 25, 16, 17, 23], ['2019-10-27', '太陽', 26, 29, 36, 39], ['2019-10-27', '快艇', 22, 33, 30, 37], ['2019-10-24', '黃蜂', 37, 26, 33, 30], ['2019-10-24', '公牛', 28, 27, 40, 30], ['2019-10-24', '溜馬', 24, 31, 31, 24], ['2019-10-24', '活塞', 27, 27, 29, 36], ['2019-10-24', '魔術', 28, 27, 16, 23], ['2019-10-24', '騎士', 24, 17, 24, 20], ['2019-10-24', '籃網', 22, 34, 37, 22], ['2019-10-24', '灰狼', 33, 35, 20, 27], ['2019-10-24', '熱火', 24, 30, 29, 37], ['2019-10-24', '灰熊', 32, 28, 24, 17], ['2019-10-24', '76人', 20, 29, 28, 30], ['2019-10-24', '塞爾蒂克', 25, 23, 20, 25], ['2019-10-24', '獨行俠', 24, 38, 29, 17], ['2019-10-24', '巫師', 25, 23, 20, 32], ['2019-10-24', '馬刺', 22, 37, 24, 37], ['2019-10-24', '尼克', 15, 36, 33, 27], ['2019-10-24', '爵士', 23, 26, 19, 32], ['2019-10-24', '雷霆', 12, 34, 28, 21], ['2019-10-24', '太陽', 25, 29, 32, 38], ['2019-10-24', '國王', 29, 30, 17, 19], ['2019-10-24', '拓荒者', 27, 23, 24, 26], ['2019-10-24', '金塊', 24, 30, 19, 35], ['2019-10-30', '熱火', 29, 30, 29, 24], ['2019-10-30', '老鷹', 26, 23, 21, 27], ['2019-10-30', '金塊', 31, 30, 27, 18], ['2019-10-30', '獨行俠', 27, 33, 23, 26], ['2019-10-30', '湖人', 27, 22, 39, 32], ['2019-10-30', '灰熊', 32, 15, 20, 24]]
5.461116075515747
266
```

_nda_multi_thread.py

- 程式練習
- 請讀取comic_test.csv進行圖片下載
- 使用downPic函式進行下載
- 主連結為

http://img.17dm.com/wangzhe/manhua/613/1.jpg

- 儲存目錄為 comic_pic/
- 檔名依序為

6131.jpg

6132.jpg

.

.

	話	主連結	頁數
1			
2	613	http://img.17dm.com/wangzhe/manhua/613/	18
3			

最大頁數

- 修改成多執行緒版本
- 同時最多五個執行緒

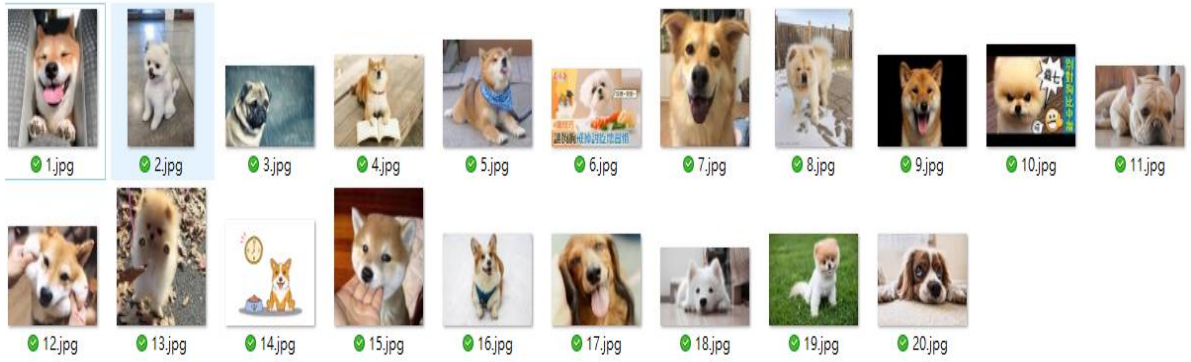
comic_downloader.py

補充

google 搜尋圖片跟下載

- 使用webdriver開啟google搜尋網頁資料

```
from selenium import webdriver

from bs4 import BeautifulSoup
import os
import urllib.request
import time

#儲存圖片
def savePic(url,file_name):
    urllib.request.urlretrieve(url, file_name)

def getWebDriver(url,path,option=None,wait=10):
    if option=='hide':
        option=webdriver.ChromeOptions()
        option.add_argument('--headless')
    try:
        chrome=webdriver.Chrome(path,options=option)
        chrome.implicitly_wait(wait)
        chrome.get(url)
    except:
        return None
    return chrome


url = 'https://www.google.com/search?q=%E7%8B%97%E7%8B%97&rlz=1C1MSIM_zh-TWTW840
#使用webdriver
path=r'C:\webdriver\chromedriver'
chrome=getWebDriver(url,path,option='hide')

soup = BeautifulSoup(chrome.page_source, 'lxml')
images = soup.find_all('div',class_='rg_bx rg_di rg_el ivg-i')
```



開啟並下載

```
image_lists=[]
for image in images:
    try:
        image_lists.append(image.find('img')['src'])
    except:
        pass

print('共%d張圖'%len(image_lists))

folder_path ='dog_photo/'

#判斷資料夾是否存在
if (not os.path.exists(folder_path)):
    os.makedirs(folder_path)


st=time.time()
for i,image in enumerate(image_lists):
    savePic(image,folder_path+str(i)+'.png')

print('done!')
time.time()-st
```

```
dog_photo/1.jpg 儲存完畢!
dog_photo/2.jpg 儲存完畢!
dog_photo/3.jpg 儲存完畢!
dog_photo/4.jpg 儲存完畢!
dog_photo/5.jpg 儲存完畢!
dog_photo/6.jpg 儲存完畢!
dog_photo/7.jpg 儲存完畢!
dog_photo/8.jpg 儲存完畢!
dog_photo/9.jpg 儲存完畢!
dog_photo/10.jpg 儲存完畢!
dog_photo/11.jpg 儲存完畢!
dog_photo/12.jpg 儲存完畢!
dog_photo/13.jpg 儲存完畢!
dog_photo/14.jpg 儲存完畢!
dog_photo/15.jpg 儲存完畢!
dog_photo/16.jpg 儲存完畢!
dog_photo/17.jpg 儲存完畢!
dog_photo/18.jpg 儲存完畢!
dog_photo/19.jpg 儲存完畢!
dog_photo/20.jpg 儲存完畢!
```

ch30_10.py

- 程式練習
- 將ch30_10.py 下載檔案部分改成多執行緒版本
- 計算時間差異

```
import time

st=time.time()
for i,image in enumerate(image_lists):
    savePic(image,folder_path+str(i)+'.png')

print('done!')
time.time()-st
```

改成多執行緒版本

ch30_11.py

多執行緒版本

```python
import threading
import time

st=time.time()
threads=[]
for i,image in enumerate(image_lists):
    t=threading.Thread(target=savePic,args=(image,folder_path+str(i)+'.png'))
    t.start()
    threads.append(t)


for thread in threads:
    thread.join()

print('done!')
time.time()-st
```

```
dog_photo1/0.png 儲存完畢!dog_photo1/6.png 儲存完畢!
dog_photo1/2.png 儲存完畢!

dog_photo1/7.png 儲存完畢!dog_photo1/1.png 儲存完畢!
dog_photo1/8.png 儲存完畢!
dog_photo1/9.png 儲存完畢!dog_photo1/4.png 儲存完畢!


dog_photo1/5.png 儲存完畢!

dog_photo1/13.png 儲存完畢!
dog_photo1/3.png 儲存完畢!dog_photo1/11.png 儲存完畢!

dog_photo1/10.png 儲存完畢!
dog_photo1/15.png 儲存完畢!
dog_photo1/12.png 儲存完畢!
dog_photo1/17.png 儲存完畢!
dog_photo1/14.png 儲存完畢!
dog_photo1/18.png 儲存完畢!
dog_photo1/19.png 儲存完畢!dog_photo1/16.png 儲存完畢!

done!
0.21093225479125977
```

ch30_11.py

- 補充

```
import threading

#目前線程數
print(threading.active_count())
#目前所有執行緒
print(threading.enumerate())
#當前執行緒
print(threading.current_thread())
```

useragent補充資料

- headers = {'user-agent': 'Mozilla/5.0 (Macintosh Intel Mac OS X 10_13_4)\AppleWebKit/537.36 (KHTML, like Gecko) Chrome/66.0.3359.181 Safari/537.36'}

- pip3 install fake-useragent

https://zhuanlan.zhihu.com/p/27436023