



# 111-1基礎程式設計(17)

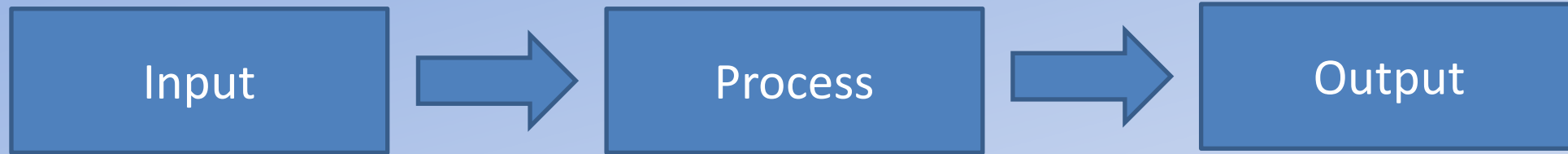
亞大資工系

# 課程大綱

- Week17-網路爬蟲相關基礎套件
  - Topic 1(主題1)- urllib函數庫
  - Topic 2(主題2)- SQLite 數據庫
  - Topic 3(主題3)-收集學校新聞的大數據
  - Topic 4(主題4)-非同步式(asynchronous)
  - Topic 5(主題5)-並行 Concurrency
  - Topic 6(主題6)-內建函數和函數庫的複習



# IPO Model (W17)



從網路讀取資源  
從資料庫查詢資料

非同步式(asynchronous)  
並行(concurrency)

寫入資料庫



# Topic 1- urllib函數庫

`urllib.request` 是一個用來從URLs (Uniform Resource Locators)取得資料的Python模組。它提供了一個非常簡單的介面能接受多種不同的協議，`urlopen` 函數。也提供了較複雜的介面用於處理一些常見的狀況，例如:基本的authentication、cookies、proxies等等，這些都可以由handler或opener物件操作。

```
import urllib.request
with urllib.request.urlopen('http://www.asia.edu.tw/') as response:
    html = response.read()
```



# Topic 2-SQLite 數據庫 DB-API 2.0 接口

- 創建一個 Connection 物件
- 創建一個 Cursor 游標物件

```
import sqlite3
con = sqlite3.connect('example.db')

cur = con.cursor()

# Create table
cur.execute('''CREATE TABLE stocks(date text, trans text, symbol text, qty real, price real)''')

# Insert a row of data
cur.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")

# Save (commit) the changes
con.commit()

# We can also close the connection if we are done with it.
# Just be sure any changes have been committed or they will be lost.
con.close()
```



# SQL (Structured Query Language)

- DDL: data definition language

```
CREATE TABLE Books  
(Id INT PRIMARY KEY IDENTITY(1,1),  
Name VARCHAR (50) NOT NULL,  
Price INT)
```

- DML:

```
INSERT into students values (1,'ashish','java');  
INSERT into students values (2,'rahul','C++');  
SELECT * from students;
```



# Topic 3-讀取學校的新聞標題

- 讀取學校的新聞標題
- 建立新聞標題的資料庫
- 查詢學校新聞標題有人工智慧



# Topic 4- asyncio (Since 3.4)

## 非同步式(asynchronous)

同步的網頁要求

```
import requests
import time
```

```
url = 'https://www.google.com.tw/'
start_time = time.time()
```

```
def send_req(url):
```

```
    res = requests.get(url)    非同步的網頁要求
```

```
for i in range(10):
```

```
    send_req(url)
```

```
url = 'https://www.asia.edu.tw/'
```

```
async def send_req(url):
```

```
    res = await loop.run_in_executor(None, requests.get, url)
```

```
tasks = []
```

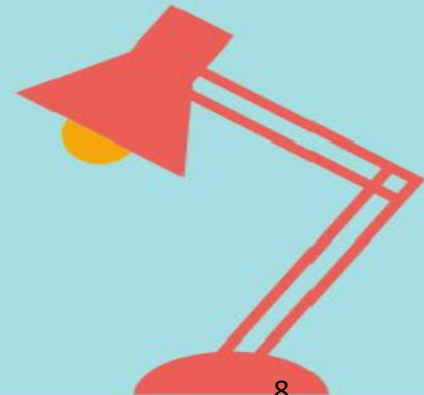
```
loop = asyncio.get_event_loop()
```

```
for i in range(10):
```

```
    task = loop.create_task(send_req(url))
```

```
    tasks.append(task)
```

```
loop.run_until_complete(asyncio.wait(tasks))
```





# Topic 5-並行 Concurrency

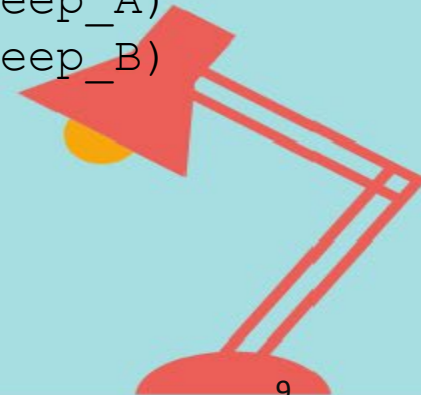
兩個函數在同一個process(thread)依序執行

```
import time
def sleep_A():
    for i in range(2):
        print(i, end="_")
        time.sleep(1)
    return
def sleep_B():
    for i in range(3):
        print(i, end="=")
        time.sleep(1)
    return
```

```
sleep_A()
sleep_B()
```

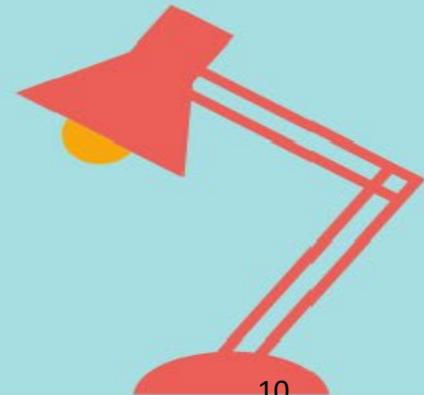
兩個函數在不同的thread同時執行

```
def sleep_A():
    for i in range(2):
        print(i, end="_")
        time.sleep(1)
    return
def sleep_B():
    for i in range(3):
        print(i, end="=")
        time.sleep(1)
    return
thread_1 = threading.Thread(target=sleep_A)
thread_2 = threading.Thread(target=sleep_B)
thread_1.start() # 啟動這個執行緒
thread_2.start() # 啟動這個執行緒
thread_1.join()
thread_2.join()
```



# Topic 6-內建函數和函數庫的複習

- 文字排序
- 每個月的第一天星期幾和有幾天





Thanks!

Q&A

