



mongoDB® 資料操作

朱克剛

安裝 PYTHON LIBRARY

```
$ pip3 install pymongo
```

```
$ pip3 install "pymongo[srv]"
```

使用 MongoDB 雲端服務
Atlas 時要裝

連線

```
import pymongo
```

```
client = pymongo.MongoClient(host='localhost', port=27017)
```



預設值

簡單範例 – 新增資料

◆從政府資料開放平台儲存空氣品質指標資料

```
import urllib.request as urllib
import json
import pymongo
```

```
url = '空氣品質指標網址'
response = urllib.urlopen(url)
text = response.read().decode('utf-8')
text = text.replace('PM2.5', 'PM2_5')
jsonObj = json.loads(text)
```

MongyDB的Key不可以包含「.」

```
client = pymongo.MongoClient()
db = client.opendata
db.AQI.insert_many(jsonObj['records'])
```

注意這裡先只存 records 下的陣列資料，所以函數要使用 insert_many()

查詢所有資料

◆查詢 AQI (Collection) 中的所有資料 (Documents)

```
import pymongo
from pprint import pprint

client = pymongo.MongoClient()

db = client.opendata
cursor = db.AQI.find()

pprint(list(cursor))
```

也可以這樣寫：

```
db = client['opendata']
cursor = db['AQI'].find()
```

只列出部分欄位

◆只顯示 County 與 AQI

可省略

```
cursor = db.AQI.find({}, projection={'County': 1, 'AQI': 1})
```

◆除 County 與 AQI 外，其他都要

```
cursor = db.AQI.find({}, {'County': 0, 'AQI': 0})
```

_ID欄位

每一筆資料 (Document) 都有 `_id` 欄位 (key)，值唯一並預設為索引

新增資料時，MongoDB 自動填入，也可手動填值。資料型態不限純量型別，也可為陣列或字典。

- 例如：

```
cursor = db.test.insert_one({  
    '_id': {'firstname': '小毛', 'lastname': '王'},  
    'age': 20  
})
```

查詢條件

- ◆只列出臺中市西屯的資料 (如確定只有一筆，用 `find_one`)

```
doc = db.AQI.find_one({'County': '臺中市', 'SiteName': '西屯'})
```

- ◆列出臺中市與新北市的資料 - `$or`, `$and` (可能有很多筆時，用 `find`)

```
cursor = db.AQI.find({  
    '$or': [  
        {'County': '臺中市'},  
        {'County': '新北市'}  
    ]  
})
```

- ◆練習把「臺中市西屯」改為 `$and` 語法

使用 IN 與 NOT IN

◆列出臺中市與新北市的資料

```
cursor = db.AQI.find({  
    'County': {'$in': ['臺中市', '新北市']}  
})
```

比較運算子

```
cursor = db.AQI.find({'value': {'$gte': 50}})
```

運算子	說明
\$gt	大於運算子 (greater than) 。查詢數字超過某個值的資料，相當於數學符號「 $>$ 」。
\$gte	大於等於運算子 (greater than or equal) 。查詢數字大於或等於某個值的資料，相當於數學符號「 \geq 」。
\$lt	小於運算子 (less than) 。查詢數字不到某個值的資料，相當於數學符號「 $<$ 」。
\$lte	小於等於運算子 (less than or equal) 。查詢數字小於或等於某個值的資料，相當於數學符號「 \leq 」。
\$ne	不等於運算子 (not equal) 。查詢所有不等於該值的資料，相當於數學符號「 \neq 」。

\$WHERE 語句

- ◆ 列出 AQI 超過 160 的資料
- ◆ 由於AQI欄位的型態為字串，在MongoDB中不能比較大小，但 JavaScript會自動把字串轉成數字，因此也可以使用 JavaScript 處理

```
cursor = db.AQI.find().where('parseInt(this.AQI) > 100')
```

使用正規表示法查詢

◆列出 County 有「中」字的資料

```
cursor = db.AQI.find({  
    'County': {'$regex': '中'}  
})
```

^中：字串開頭為「中」
中\$：字串結尾為「中」
中|南：有「中」或者「南」

查詢AQI ≥ 150 的資料

```
cursor = db.AQI.find({  
    'AQI': {  
        '$regex': '1[5-9] | [2-9]..'   
    }  
})
```

排序

中文按照 Unicode 排序
(跟亂排差不多)

◆按照 County 正向排序

```
cursor = db.AQI.find().sort('County')
```

◆按照 County 反向排序

```
cursor = db.AQI.find().sort('County', -1)
```

◆按照中文筆畫數排

```
cursor = db.AQI.find().collation({'locale': 'zh_Hant'}).sort('County')
```

◆按照注音符號排



```
.collation({'locale': 'zh@collation=zhuyin'})
```

插入一筆資料

```
db.AQI.insert_one({  
    'County': '臺中市',  
    'SiteName': '我的家',  
    'AQI': '30',  
    'PublishTime': '2021/3/20 09:30:00',  
    'Custom': True  
})
```

插入多筆資料

```
db.AQI.insert_many([
    {
        'County': '臺北市',
        'SiteName': '內湖',
        'AQI': '120',
        'PublishTime': '2021/3/20 10:28:00',
        'Custom': True
    },
    {
        'County': '南投縣',
        'SiteName': '日月潭',
        'AQI': '80',
        'PublishTime': '2021/3/20 10:42:00',
        'Custom': True
    }
])
```


查詢有特定欄位的資料

```
cursor = db.AQI.find({  
    'Custom': {  
        '$exists': True  
    }  
})
```

修改資料

◆修改南投縣日月潭的AQI

```
db.AQI.update_one(  
    { 'County': '南投縣', 'SiteName': '日月潭' },  
    { '$set': { 'AQI': '200' } }  
)
```

第一個參數是條件

◆沒有 key 就新增 (增加經緯度座標)

```
db.AQI.update_one(  
    { 'County': '南投縣', 'SiteName': '日月潭' },  
    { '$set': { 'Lat': 23.858149, 'Lon': 120.916006 } }  
)
```

刪除欄位

\$unset

刪除資料

◆刪除日月潭資料

```
db.AQI.delete_one({'SiteName': '日月潭'})
```

◆刪除所有使用者自行增加的資料

```
db.AQI.delete_many({'Custom': True})
```

取代資料

◆將 SiteName 為沙鹿的資料取代成別的資料

```
db.AQI.replace_one({'SiteName': '沙鹿'}, {'SiteName': '沙鹿', 'AQI': '100'})
```



第一個參數是條件

◆upsert=True：找不到就新增

```
db.AQI.replace_one(
    {'SiteName': '南屯'},
    {'County': '臺中市', 'SiteName': '南屯', 'AQI': '80'},
    upsert = True
)
```