

## 정리

```
pip install pymongo    # 설치

### DB 연동
import pymongo         # 불러오기
client = pymongo.MongoClient("localhost", 27017)    # 연결하기

### DB이름 확인
client.database_names()    # DB 확인

### DB지정 및 collection 생성
db = client.test          # DB 지정
db_col = db.items         # collection 생성

### 데이터 삽입(data insert)
data = {
    "name": "cola",
    "pty": 5,
    "price": 500
}
db_col.insert(data)    # items 에 데이터 넣기

### 데이터 확인
cursor = db_col.find()
for doc in cursor:
    print(doc)
```

## 01 시작 및 준비

In [5]:

```
### DB 연동
import pymongo    # 불러오기
client = pymongo.MongoClient("localhost", 27017)    # 연결하기
```

## 작업 절차

- (1) DB지정 및 collection 생성
- (2) 데이터 삽입
- (3) 데이터 확인

In [6]:

```

### DB 지정 및 collection 생성
db = client.testA      # DB 지정
db_col = db.items      # collection 생성

### 데이터 삽입(data insert)
for i in range(5):
    data = {
        "name": "cola"+str(i),
        "pty": 5,
        "price": 500*i
    }
    db_col.insert(data)  # items 에 데이터 넣기

### 데이터 확인
cursor = db_col.find()
for doc in cursor:
    print(doc)

```

```

{'_id': ObjectId('5b0ec6145378e607b0e07824'), 'name': 'cola0', 'pty': 5, 'price': 0}
{'_id': ObjectId('5b0ec6155378e607b0e07825'), 'name': 'cola1', 'pty': 5, 'price': 50
0}
{'_id': ObjectId('5b0ec6155378e607b0e07826'), 'name': 'cola2', 'pty': 5, 'price': 10
00}
{'_id': ObjectId('5b0ec6155378e607b0e07827'), 'name': 'cola3', 'pty': 5, 'price': 15
00}
{'_id': ObjectId('5b0ec6155378e607b0e07828'), 'name': 'cola4', 'pty': 5, 'price': 20
00}
{'_id': ObjectId('5b0ec6665378e607b0e07829'), 'name': 'cola0', 'pty': 5, 'price': 0}
{'_id': ObjectId('5b0ec6665378e607b0e0782a'), 'name': 'cola1', 'pty': 5, 'price': 50
0}
{'_id': ObjectId('5b0ec6665378e607b0e0782b'), 'name': 'cola2', 'pty': 5, 'price': 10
00}
{'_id': ObjectId('5b0ec6665378e607b0e0782c'), 'name': 'cola3', 'pty': 5, 'price': 15
00}
{'_id': ObjectId('5b0ec6665378e607b0e0782d'), 'name': 'cola4', 'pty': 5, 'price': 20
00}
{'_id': ObjectId('5b0ec67b5378e607b0e0782f'), 'name': 'cola0', 'pty': 5, 'price': 0}
{'_id': ObjectId('5b0ec67b5378e607b0e07830'), 'name': 'cola1', 'pty': 5, 'price': 50
0}
{'_id': ObjectId('5b0ec67b5378e607b0e07831'), 'name': 'cola2', 'pty': 5, 'price': 10
00}
{'_id': ObjectId('5b0ec67b5378e607b0e07832'), 'name': 'cola3', 'pty': 5, 'price': 15
00}
{'_id': ObjectId('5b0ec67b5378e607b0e07833'), 'name': 'cola4', 'pty': 5, 'price': 20
00}

```

```

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel_launcher.py:12: DeprecationWar
ning: insert is deprecated. Use insert_one or insert_many instead.
  if sys.path[0] == '':

```

## 2-1 Documnet 건수 확인

In [7]:

```
db_col.count()
```

Out[7]:

15

## 2-2 조건 검색 - cola3인 친구만 뽑기

In [8]:

```
for post in db_col.find({"name": "cola3"}):
    print(post)
```

```
{'_id': ObjectId('5b0ec6155378e607b0e07827'), 'name': 'cola3', 'pty': 5, 'price': 1500}
{'_id': ObjectId('5b0ec6665378e607b0e0782c'), 'name': 'cola3', 'pty': 5, 'price': 1500}
{'_id': ObjectId('5b0ec67b5378e607b0e07832'), 'name': 'cola3', 'pty': 5, 'price': 1500}
```

In [12]:

```
### 가격이 1000원 이상인 친구들의 건수
print(db_col.find({"price": 1000}).count() )
print(db_col.find({"price": 1500}).count() )
print(db_col.find({"price": 2000}).count() )
```

```
3
3
3
```

## 이를 해결해 보자.

- (1) 어떻게 하면 위의 내용을 한줄로 표시할 수 있을까? 900보다 큰 데이터 개수 뽑기
- (2) 그렇다면 1100보다 작은 친구들의 개수는 어떻게 한줄로 뽑을 수 있을까?

In [23]:

```
print(db_col.find({"price": { "$gt": 900 } }).count() )
print(db_col.find({"price": { "$lt": 1000 } }).count() )
print(db_col.find({"price": { "$gt": 900, "$lt": 2100 } }).count() )
```

```
9
6
9
```

## 2-3 정렬 - sort

In [13]:

```
db_col.find({"name": "cola3"})
```

Out[13]:

<pymongo.cursor.Cursor at 0x17241f646a0>

In [16]:

```
for post in db_col.find().sort('price'):  
    print(post)
```

```
{'_id': ObjectId('5b0ec6145378e607b0e07824'), 'name': 'cola0', 'pty': 5, 'price': 0}  
{'_id': ObjectId('5b0ec6665378e607b0e07829'), 'name': 'cola0', 'pty': 5, 'price': 0}  
{'_id': ObjectId('5b0ec67b5378e607b0e0782f'), 'name': 'cola0', 'pty': 5, 'price': 0}  
{'_id': ObjectId('5b0ec6155378e607b0e07825'), 'name': 'cola1', 'pty': 5, 'price': 50  
0}  
{'_id': ObjectId('5b0ec6665378e607b0e0782a'), 'name': 'cola1', 'pty': 5, 'price': 50  
0}  
{'_id': ObjectId('5b0ec67b5378e607b0e07830'), 'name': 'cola1', 'pty': 5, 'price': 50  
0}  
{'_id': ObjectId('5b0ec6155378e607b0e07826'), 'name': 'cola2', 'pty': 5, 'price': 10  
00}  
{'_id': ObjectId('5b0ec6665378e607b0e0782b'), 'name': 'cola2', 'pty': 5, 'price': 10  
00}  
{'_id': ObjectId('5b0ec67b5378e607b0e07831'), 'name': 'cola2', 'pty': 5, 'price': 10  
00}  
{'_id': ObjectId('5b0ec6155378e607b0e07827'), 'name': 'cola3', 'pty': 5, 'price': 15  
00}  
{'_id': ObjectId('5b0ec6665378e607b0e0782c'), 'name': 'cola3', 'pty': 5, 'price': 15  
00}  
{'_id': ObjectId('5b0ec67b5378e607b0e07832'), 'name': 'cola3', 'pty': 5, 'price': 15  
00}  
{'_id': ObjectId('5b0ec6155378e607b0e07828'), 'name': 'cola4', 'pty': 5, 'price': 20  
00}  
{'_id': ObjectId('5b0ec6665378e607b0e0782d'), 'name': 'cola4', 'pty': 5, 'price': 20  
00}  
{'_id': ObjectId('5b0ec67b5378e607b0e07833'), 'name': 'cola4', 'pty': 5, 'price': 20  
00}
```

## 2-4 몇건의 데이터만 보기(5건)

In [26]:

```
for post in db_col.find().limit(5):  
    print(post)
```

```
{'_id': ObjectId('5b0ec6145378e607b0e07824'), 'name': 'cola0', 'pty': 5, 'price': 0}  
{'_id': ObjectId('5b0ec6155378e607b0e07825'), 'name': 'cola1', 'pty': 5, 'price': 50  
0}  
{'_id': ObjectId('5b0ec6155378e607b0e07826'), 'name': 'cola2', 'pty': 5, 'price': 10  
00}  
{'_id': ObjectId('5b0ec6155378e607b0e07827'), 'name': 'cola3', 'pty': 5, 'price': 15  
00}  
{'_id': ObjectId('5b0ec6155378e607b0e07828'), 'name': 'cola4', 'pty': 5, 'price': 20  
00}
```

## 2-5 \_id를 이용하여 검색을 해 보기

In [37]:

```
### DB지정 및 collection 생성
db = client.testA      # DB 지정
db_col = db.itemPra    # collection 생성

### 데이터 삽입(data insert)
db_col.insert({ "_id" : 1, "score" : [ -1, 3 ] }) # items 에 데이터 넣기
db_col.insert({ "_id" : 2, "score" : [ 1, 5 ] }) # items 에 데이터 넣기
db_col.insert({ "_id" : 3, "score" : [ 5, 5 ] }) # items 에 데이터 넣기
db_col.insert({ "_id" : 4, "score" : [ -1, 3 ] }) # items 에 데이터 넣기
db_col.insert({ "_id" : 5, "score" : [ 1, 5 ] }) # items 에 데이터 넣기
db_col.insert({ "_id" : 6, "score" : [ 5, 5 ] }) # items 에 데이터 넣기
```

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:6: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:7: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

import sys

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:8: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

Out[37]:

6

In [41]:

```
### 데이터 확인
cursor = db_col.find()
for doc in cursor:
    print(doc)
```

```
{ '_id': 1, 'score': [-1, 3] }
{ '_id': 2, 'score': [1, 5] }
{ '_id': 3, 'score': [5, 5] }
{ '_id': 4, 'score': [-1, 3] }
{ '_id': 5, 'score': [1, 5] }
{ '_id': 6, 'score': [5, 5] }
```

In [48]:

```
# 0보다 작고, -2보다 큰 조건을 둘다 만족하는 데이터 셋 불러오기
for doc in db_col.find( { 'score': { '$gt': -2, '$lt': 0 } } ):
    print(doc)
```

```
{ '_id': 1, 'score': [-1, 3] }
{ '_id': 4, 'score': [-1, 3] }
```

In [51]:

```

### DB지정 및 collection 생성
db = client.testA      # DB 지정
db_col = db.itemPra1   # collection 생성

### 데이터 삽입(data insert)
db_col.insert({ "_id" : 1, "score" : 1 }) # itemPra1 에 데이터 넣기
db_col.insert({ "_id" : 2, "score" : 2 }) # itemPra1 에 데이터 넣기
db_col.insert({ "_id" : 3, "score" : 3 }) # itemPra1 에 데이터 넣기
db_col.insert({ "_id" : 4, "score" : 4 }) # itemPra1 에 데이터 넣기
db_col.insert({ "_id" : 5, "score" : 5 }) # itemPra1 에 데이터 넣기

### 데이터 확인
cursor = db_col.find()
for doc in cursor:
    print(doc)

```

```

{'_id': 1, 'score': 1}
{'_id': 2, 'score': 2}
{'_id': 3, 'score': 3}
{'_id': 4, 'score': 4}
{'_id': 5, 'score': 5}

```

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:6: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:7: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

```
import sys
```

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:8: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:9: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

```
if __name__ == '__main__':
```

C:\Users\WWITHJSW\Anaconda3\lib\site-packages\Wipykernel\_launcher.py:10: DeprecationWarning: insert is deprecated. Use insert\_one or insert\_many instead.

```
# Remove the CWD from sys.path while we load stuff.
```

In [55]:

```

cursor = db_col.find({ 'score': 2 })
for doc in cursor:
    print(doc)

```

```
{'_id': 2, 'score': 2}
```

## Ref

<https://docs.mongodb.com/manual/reference/method/db.collection.find/>

In [ ]:

