Pymongo

- MongoDB를 Python에서 사용할수있도록 도와주는 ORM
- w3schools.com/python/python_mongodb_getstarted.asp

In [1]:

```
import requests, pymongo
import pandas as pd
```

Connect Server

In [2]:

```
# client 연결
client = pymongo.MongoClient('mongodb://test:testpw@43.201.104.0:27017')
client
```

Out[2]:

```
MongoClient(host=['43.201.104.0:27017'], document_class=dict, tz_aware=False, co
nnect=True)
```

Create or Connect Database

In [3]:

```
# 데이터 베이스 목록 출력
list(client.list_databases())
```

Out[3]:

```
[{'name': 'admin', 'sizeOnDisk': 114688.0, 'empty': False},
 {'name': 'config', 'sizeOnDisk': 110592.0, 'empty': False},
 {'name': 'local', 'sizeOnDisk': 73728.0, 'empty': False},
 {'name': 'mongo', 'sizeOnDisk': 106496.0, 'empty': False}]
```

In [4]:

```
# 데이터 베이스에 접속 및 생성
db = client.mongo
db
```

Out[4]:

```
Database(MongoClient(host=['43.201.104.0:27017'], document_class=dict, tz_aware=
False, connect=True), 'mongo')
```

Collection

- CRUD : create read update delete

READ Collection

In [5]:

```
# 데이터 베이스의 컬렉션 리스트 확인
db.list_collection_names()
```

Out[5]:

```
['info', 'user']
```

CREATE Collection

In [6]:

```
# 컬렉션 선택 생성
collection = db.info
collection
```

Out[6]:

```
Collection(Database(MongoClient(host=['43.201.104.0:27017'], document_class=dic
t, tz_aware=False, connect=True), 'mongo'), 'info')
```

UPDATE Collection

In [8]:

```
db.user.rename('users')
```

Out[8]:

```
{'ok': 1.0}
```

In [9]:

```
# 데이터 베이스의 컬렉션 리스트 확인
db.list_collection_names()
```

Out[9]:

```
['info', 'users']
```

DELETE Collection

In [10]:

```
# 컬렉션 선택 및 삭제
# collection = db.info1
# collection.drop()
```

In [11]:

```
# 데이터 베이스의 컬렉션 리스트 확인
# db.list_collection_names()
```

Document

- CRUD : create read update delete

READ Document

In [12]:

```
# 한개의 도큐먼트 가져오기
document = collection.find_one({"subject":"python"})
document
```

Out[12]:

```
{'_id': ObjectId('643ff4007304400a91102974'), 'subject': 'python', 'level': 3}
```

In [13]:

```
# 모든 도큐먼트 가져오기
documents = collection.find()
documents
```

Out[13]:

```
<pymongo.cursor.Cursor at 0x7f6c721378b0>
```

In [14]:

```
# 도큐먼트 리스트로 형변환해서 data 변수에 저장
data = list(documents)
print(len(data), data)
```

```
14 [{'_id': ObjectId('643ff4007304400a91102974'), 'subject': 'python', 'level':
3}, {'_id': ObjectId('643ff4007304400a91102975'), 'subject': 'web', 'level': 1},
{'_id': ObjectId('643ff4007304400a91102976'), 'subject': 'sql', 'level': 2}, {'_
id': ObjectId('643ff4007304400a91102977'), 'subject': 'java', 'level': 3}, {'_i
d': ObjectId('643ff4007304400a91102978'), 'subject': 'html', 'level': 1}, {'_i
d': ObjectId('643ff4007304400a91102979'), 'subject': 'css', 'level': 2}, {'_id':
ObjectId('6446ac98a524e8861d56d39a'), 'subject': 'css', 'level': 1}, {'_id': Obj
ectId('6446ac9ba524e8861d56d39b'), 'subject': 'java', 'level': 1, 'comments':
[{'name': 'peter', 'msg': 'easy'}]}, {'_id': ObjectId('6446ac9ba524e8861d56d39
c'), 'subject': 'html', 'level': 2, 'comments': [{'name': 'peter', 'msg': 'mediu
m'}]}, {'_id': ObjectId('6446ac9ba524e8861d56d39d'), 'subject': 'gulp', 'level':
3, 'comments': [{'name': 'peter', 'msg': 'hard'}]}, {'_id': ObjectId('6449462106
10ae8503719174'), 'subject': 'css', 'level': 1}, {'_id': ObjectId('644946440610a
e8503719175'), 'subject': 'java', 'level': 1, 'comments': [{'name': 'peter', 'ms
g': 'easy'}]}, {'_id': ObjectId('644946440610ae8503719176'), 'subject': 'html',
'level': 2, 'comments': [{'name': 'peter', 'msg': 'medium'}]}, {'_id': ObjectId
('644946440610ae8503719177'), 'subject': 'gulp', 'level': 3, 'comments': [{'nam
e': 'peter', 'msg': 'hard'}]}]
```

In [15]:

```
# pandas DataFrame 으로 변환
df = pd.DataFrame(data)
df.tail()
```

Out[15]:

	_id	subject	level	comments
9	6446ac9ba524e8861d56d39d	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
10	644946210610ae8503719174	css	1	NaN
11	644946440610ae8503719175	java	1	[{'name': 'peter', 'msg': 'easy'}]
12	644946440610ae8503719176	html	2	[{'name': 'peter', 'msg': 'medium'}]
13	644946440610ae8503719177	gulp	3	[{'name': 'peter', 'msg': 'hard'}]

In [16]:

```
# 데이터를 한번 읽으면 데이터가 사라짐
list(documents)
```

Out[16]:

```
[]
```

In [17]:

```
# 도큐먼트의 갯수를 가져옴
count = collection.count_documents({})
count
```

Out[17]:

14

In [18]:

```
# 레벨이 2이상인 도큐먼트를 subject 내림차순으로 정렬
documents = collection.find({"level": {"$gte": 2}}).sort("subject", pymongo.DESCENDING)
data = pd.DataFrame(list(documents))
data
```

Out[18]:

	_id	subject	level	comments
0	643ff4007304400a91102976	sql	2	NaN
1	643ff4007304400a91102974	python	3	NaN
2	643ff4007304400a91102977	java	3	NaN
3	6446ac9ba524e8861d56d39c	html	2	[{'name': 'peter', 'msg': 'medium'}]
4	644946440610ae8503719176	html	2	[{'name': 'peter', 'msg': 'medium'}]
5	6446ac9ba524e8861d56d39d	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
6	644946440610ae8503719177	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
7	643ff4007304400a91102979	css	2	NaN

CREATE Documnet

In [19]:

```
# 한개의 대이터 저장 - insert_one
data = {"subject":"css", "level":1}
result = collection.insert_one(data)
print(result.inserted_id)
```

644947207a5088f5dbdfe222

In [20]:

```
documents = collection.find()
pd.DataFrame(list(documents)).tail(3)
```

Out[20]:

	_id	subject	level	comments
12	644946440610ae8503719176	html	2	[{'name': 'peter', 'msg': 'medium'}]
13	644946440610ae8503719177	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
14	644947207a5088f5dbdfe222	css	1	NaN

In [21]:

```
# 여러개의 데이터 저장 - insert_many
data = [
    {"subject":"java", "level":1, "comments":[{"name":"peter", "msg":"easy"}]},
    {"subject":"html", "level":2, "comments":[{"name":"peter", "msg":"medium"}]},
    {"subject":"gulp", "level":3, "comments":[{"name":"peter", "msg":"hard"}]},
]
result = collection.insert_many(data)
print(result.inserted_ids)
```

```
[ObjectId('644947297a5088f5dbdfe223'), ObjectId('644947297a5088f5dbdfe224'), Obj
ectId('644947297a5088f5dbdfe225')]
```

In [81]:

```
# limit 데이터 갯수 제한
documents = collection.find().limit(3)
pd.DataFrame(list(documents))
```

Out[81]:

	_id	subject	level
0	643ff4007304400a91102974	python	3
1	643ff4007304400a91102975	web	1
2	643ff4007304400a91102976	sql	2

직방 원룸 데이터 수집해서 데이터베이스에 저장하기

In [25]:

```
# !pip install geohash2
```

In [26]:

```
import zigbang as zb
```

In [42]:

```
data = zb.oneroom('망원동')
data.tail(2)
```

Out[42]:

	item_id	sales_type	deposit	rent	size_m2	floor	building_floor	title	address1	mana
95	36361661	월세	300	55	19.83	1	3	🌸풀옵션🌸마포구청역5분🌸깔끔한방🌸	서울시 마포구 망원동	
96	36362867	월세	1000	65	42.98	반지하	3	💜6호선 초역세💖 고양이와 함께💖 채광굿 반지하💜	서울시 마포구 망원동	

In [54]:

```
data.to_dict('records')[0]
```

Out[54]:


```
{'item_id': 36363281,
 'sales_type': '전세',
 'deposit': 26000,
 'rent': 0,
 'size_m2': 50.0,
 'floor': '2',
 'building_floor': '4',
 'title': '저렴한 방3개 전세',
 'address1': '서울시 마포구 망원동',
 'manage_cost': '3',
 'reg_date': '2023-04-25T16:25:36+09:00',
 'is_new': True}
```

In [55]:

```
db = client.zigbang # zigbang database
collection = db.oneroom # oneroom collection
result = collection.insert_many(data.to_dict('records'))
len(result.inserted_ids)
```

Out[55]:

97

In [56]:

```
# 저장된 데이터 읽어오기
data = collection.find()
df = pd.DataFrame(list(data))
df.tail(2)
```

Out[56]:

	_id	item_id	sales_type	deposit	rent	size_m2	floor	building_fl
95	644949277a5088f5dbdfe285	36361661	월세	300	55	19.83	1	
96	644949277a5088f5dbdfe286	36362867	월세	1000	65	42.98	반지하	

In [66]:

```
# deposit : 1000 이하, rent : 100 이하, size_m2: 30 이상, floor: 반지하 아님manage_cost : 오름차
query = {'deposit': {'$lte': 1000}, 'rent': {'$lte': 100}, 'size_m2': {'$gte': 30}, 'floor': {'$
documents = collection.find(query).sort("manage_cost", pymongo.DESCENDING)
df1 = pd.DataFrame(list(documents))
df1
```

Out[66]:

	_id	item_id	sales_type	deposit	rent	size_m2	floor	building_flo
0	644949277a5088f5dbdfe238	36401812	월세	1000	90	47.11	3	
1	644949277a5088f5dbdfe25d	36296177	월세	1000	70	119.64	1	

UPDATE Document

In [73]:

```
db = client.mongo
collection = db.info
documents = collection.find({'subject': 'html'})
pd.DataFrame(list(documents))
```

Out[73]:

	_id	subject	level	comments
0	643ff4007304400a91102978	html	1	NaN
1	6446ac9ba524e8861d56d39c	html	2	[{'name': 'peter', 'msg': 'medium'}]
2	644946440610ae8503719176	html	2	[{'name': 'peter', 'msg': 'medium'}]
3	644947297a5088f5dbdfe224	html	2	[{'name': 'peter', 'msg': 'medium'}]

In [76]:

```
query = {'subject': 'html'}
update = {'$set': {'level': 3}}
collection.update_one(query, update)
```

Out[76]:

```
<pymongo.results.UpdateResult at 0x7f6c6eb03850>
```

In [77]:

```
documents = collection.find({'subject': 'html'})
pd.DataFrame(list(documents))
```

Out[77]:

	_id	subject	level	comments
0	643ff4007304400a91102978	html	3	NaN
1	6446ac9ba524e8861d56d39c	html	2	[{'name': 'peter', 'msg': 'medium'}]
2	644946440610ae8503719176	html	2	[{'name': 'peter', 'msg': 'medium'}]
3	644947297a5088f5dbdfe224	html	2	[{'name': 'peter', 'msg': 'medium'}]

In [78]:

```
query = {'subject': 'html'}
update = {'$set': {'level': 1}}
collection.update_many(query, update)
```

Out[78]:

```
<pymongo.results.UpdateResult at 0x7f6c6fe62a00>
```

In [79]:

```
documents = collection.find({'subject': 'html'})
pd.DataFrame(list(documents))
```

Out[79]:

	_id	subject	level	comments
0	643ff4007304400a91102978	html	1	NaN
1	6446ac9ba524e8861d56d39c	html	1	[{'name': 'peter', 'msg': 'medium'}]
2	644946440610ae8503719176	html	1	[{'name': 'peter', 'msg': 'medium'}]
3	644947297a5088f5dbdfe224	html	1	[{'name': 'peter', 'msg': 'medium'}]

DELETE Document

In [82]:

```
# level 1 데이터 삭제
documents = collection.find()
pd.DataFrame(list(documents))
```

Out[82]:

	_id	subject	level	comments
0	643ff4007304400a91102974	python	3	NaN
1	643ff4007304400a91102975	web	1	NaN
2	643ff4007304400a91102976	sql	2	NaN
3	643ff4007304400a91102977	java	3	NaN
4	643ff4007304400a91102978	html	1	NaN
5	643ff4007304400a91102979	css	2	NaN
6	6446ac98a524e8861d56d39a	css	1	NaN
7	6446ac9ba524e8861d56d39b	java	1	[{'name': 'peter', 'msg': 'easy'}]
8	6446ac9ba524e8861d56d39c	html	1	[{'name': 'peter', 'msg': 'medium'}]
9	6446ac9ba524e8861d56d39d	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
10	644946210610ae8503719174	css	1	NaN
11	644946440610ae8503719175	java	1	[{'name': 'peter', 'msg': 'easy'}]
12	644946440610ae8503719176	html	1	[{'name': 'peter', 'msg': 'medium'}]
13	644946440610ae8503719177	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
14	644947207a5088f5dbdfe222	css	1	NaN
15	644947297a5088f5dbdfe223	java	1	[{'name': 'peter', 'msg': 'easy'}]
16	644947297a5088f5dbdfe224	html	1	[{'name': 'peter', 'msg': 'medium'}]
17	644947297a5088f5dbdfe225	gulp	3	[{'name': 'peter', 'msg': 'hard'}]

In [85]:

```
# level 1 데이터 삭제
query = {'level': {'$eq': 1}}
collection.delete_many(query)
```

Out[85]:

```
<pymongo.results.DeleteResult at 0x7f6c6f69c790>
```

In [86]:

```
documents = collection.find()
pd.DataFrame(list(documents))
```

Out[86]:

	_id	subject	level	comments
0	643ff4007304400a91102974	python	3	NaN
1	643ff4007304400a91102976	sql	2	NaN
2	643ff4007304400a91102977	java	3	NaN
3	643ff4007304400a91102979	css	2	NaN
4	6446ac9ba524e8861d56d39d	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
5	644946440610ae8503719177	gulp	3	[{'name': 'peter', 'msg': 'hard'}]
6	644947297a5088f5dbdfe225	gulp	3	[{'name': 'peter', 'msg': 'hard'}]