

Mongodb 활용 + pymongo

nosql = not sql

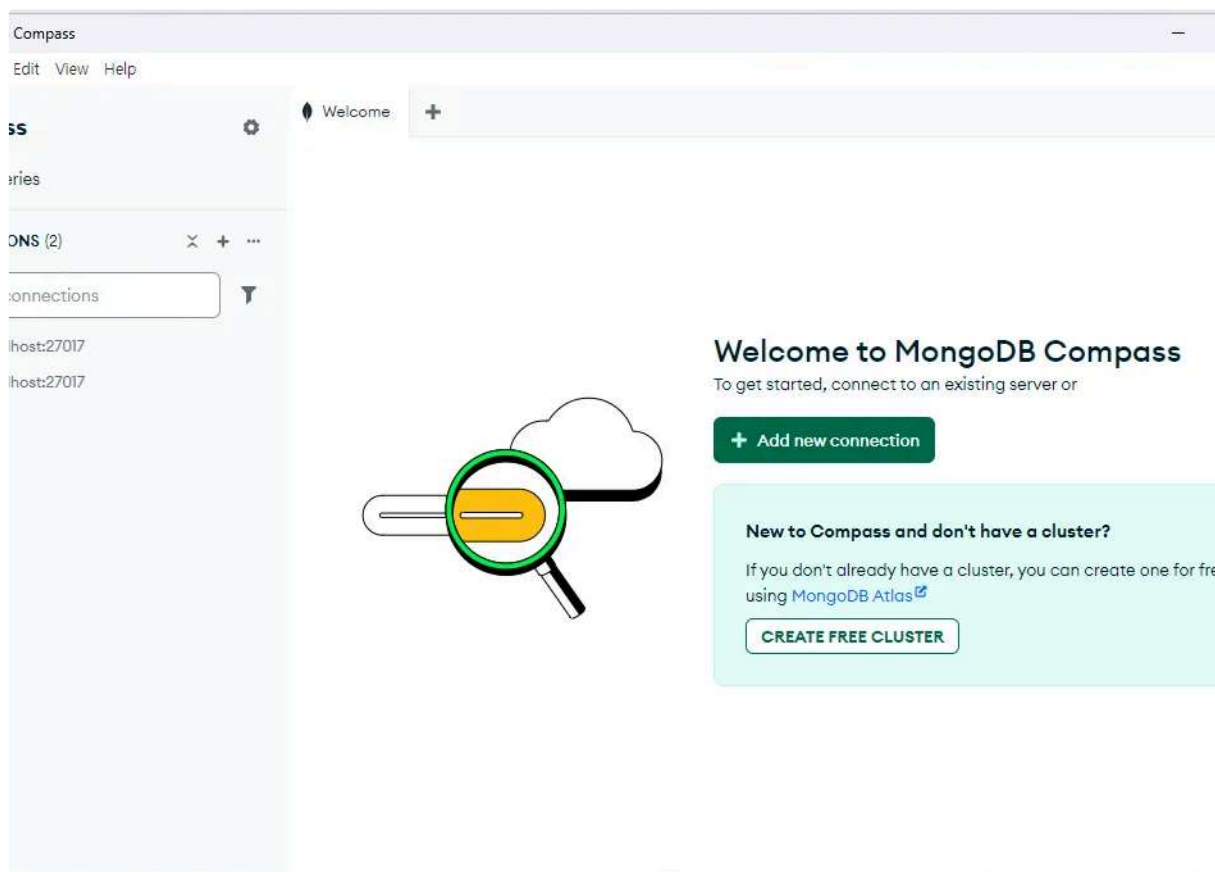
key : value 형식으로 저장을 함!!!

웹서버(Apache, IIS) —————> WAS(Tomcat, JBoss...) —————> DB

<https://www.mongodb.com/try/download/community>



설치를 하면 mongodb compass와 같이 설치가 됨



데이터베이스 기본 설정

```
from pymongo import MongoClient # MongoDB 서버에 연결 (로컬호스트, 기본 포트 27017)
client = MongoClient('mongodb://localhost:27017/') # 데이터베이스 선택 (없으면 자동 생성)
db = client['school_db'] # 컬렉션 선택 (없으면 자동 생성)
collection = db['students']
print("MongoDB에 성공적으로 연결되었습니다!")
```

데이터 생성(Create)

```

from pymongo import MongoClient client =
MongoClient('mongodb://localhost:27017/') db = client['school_db']
collection = db['students'] # 단일 문서 삽입 student = { "name": "김철수",
"age": 20, "grade": "A", "subjects": ["수학", "영어"] } result =
collection.insert_one(student) print(f"삽입된 문서 ID:
{result.inserted_id}") # 여러 문서 삽입 students = [ {"name": "이영희",
"age": 19, "grade": "B", "subjects": ["과학", "국어"]}, {"name": "박민수",
"age": 21, "grade": "A", "subjects": ["수학", "과학"]} ] result =
collection.insert_many(students) print(f"삽입된 문서 ID들:
{result.inserted_ids}")

```

데이터 조회(Read)

```

from pymongo import MongoClient client =
MongoClient('mongodb://localhost:27017/') db = client['school_db']
collection = db['students'] # 모든 문서 조회 print("모든 학생 데이터:") for
student in collection.find(): print(student)
print("=====") result =
collection.find_one({"name": "김철수"}) print(result)
print("=====") # 조건에 맞는 문서 조회
(나이가 20 이상인 학생) print("\n나이가 20 이상인 학생:") for student in
collection.find({"age": {"$gte": 20}}): print(student)

```

데이터 업데이트

```

from pymongo import MongoClient client =
MongoClient('mongodb://localhost:27017/') db = client['school_db']
collection = db['students'] # 단일 문서 업데이트 (김철수의 학년을 A+로 변경)
result = collection.update_one( {"name": "김철수"}, {"$set": {"grade":
"A+"}} ) print(f"수정된 문서 수: {result.modified_count}") # 여러 문서 업데
이트 (수학 과목을 수강하는 학생들의 학년을 B로 변경) result =
collection.update_many( {"subjects": "수학"}, {"$set": {"grade": "B"}} )
print(f"수정된 문서 수: {result.modified_count}")

```

데이터 삭제

```

from pymongo import MongoClient client =
MongoClient('mongodb://localhost:27017/') db = client['school_db']
collection = db['students'] # 단일 문서 삭제 (이름이 박민수인 학생 삭제)
result = collection.delete_one({"name": "박민수"}) print(f"삭제된 문서 수:
{result.deleted_count}") # 여러 문서 삭제 (나이가 19인 학생들 삭제) result =
collection.delete_many({"age": 19}) print(f"삭제된 문서 수:
{result.deleted_count}")

```

이전 (참고만)

```

#pip install pymongo from pymongo import MongoClient client =
MongoClient('mongodb://localhost:27017') db = client['mydatabase'] #컬렉션
collection = db['mycollection'] post = {"title":"두번째 포스트",
"content":"안녕하세요.11", "author":"John"} collection.insert_one(post)
print("오케이") #post = {"title":"세번째 포스트", "content":"안녕하세
요.3333", "author":"jeong"} #collection.insert_one(post) #print("OK") #실습
03 result = collection.find_one({"author":"John"}) print(f"검색된 문서
{result}") # 문서 수정 query = {"author": "John"} new_values = {"$set":
{"content": "수정된 내용입니다."}} collection.update_one(query, new_values)
print("문서가 성공적으로 수정되었습니다.") # 수정된 문서 확인 result =
collection.find_one(query) print("수정된 문서:", result) # 문서 삭제
delete_query = {"author": "John Doe"} collection.delete_one(delete_query)
print("문서가 성공적으로 삭제되었습니다.")

```

faker를 이용한 가짜 데이터 생성 사례

```

from faker import Faker fake = Faker("ko_KR") for _ in range(20):
print(fake.name()) print(fake.address()) print(fake.email())
print(fake.phone_number()) print("=====")

```

가짜 데이터 mongodb에 생성

```
from pymongo import MongoClient from faker import Faker # MongoDB에 연결
client = MongoClient('mongodb://localhost:27017/') # 데이터베이스 선택 db =
client['mydatabase_faker'] # 컬렉션 선택 collection = db['people'] # Faker
객체 생성 fake = Faker("ko_KR") # 가짜 데이터 생성 및 입력 for _ in
range(20): person = { 'name': fake.name(), 'address': fake.address(),
'email': fake.email(), 'phone': fake.phone_number() }
collection.insert_one(person) print("가짜 데이터가 성공적으로 입력되었습니다.")
```

이제 플라스크로 만들어보자.

```
from flask import Flask, redirect, render_template from pymongo import
MongoClient from bson import ObjectId app = Flask(__name__) # 연결 정보
client = MongoClient('mongodb://localhost:27017/') db =
client['mydatabase_faker'] collection = db['people'] # 기본 라우트 설정
@app.route('/') def index(): people = collection.find() return
render_template('list.html', people=people) # 기본 라우트 설정
@app.route('/delete_user/<user_id>') def delete_user(user_id):
collection.delete_one({'_id': ObjectId(user_id)}) return redirect('/') if
__name__ == '__main__': app.run(debug=True)
```

list.html

```
<!DOCTYPE html> <html lang="en"> <head> <meta charset="UTF-8"> <meta
name="viewport" content="width=device-width, initial-scale=1.0"> <link
rel="stylesheet" type="text/css" href="{{url_for('static',
filename='style.css')}}"> <title>Document</title> </head> <body> <table>
<tr> <td>이름</td> <td>주소</td> <td>이메일</td> <td>전화번호</td> <td>삭제
</td> </tr> {% for person in people%} <tr> <td>{{person.name}}</td> <td>
{{person.address}}</td> <td>{{person.email}}</td> <td>{{person.phone}}
</td> <td><a href="{{url_for('delete_user', user_id=person._id)}}">삭제</a>
</td> </tr> {%endfor%} </table> </body> </html>
```

style.css

```
table { width: 100%; border-collapse: collapse; margin: 20px 0; } th, td {
```