

SESSIONS 2



Connecting Python to MongoDB Database

Data Science Program

PyMongo

PyMongo is a Python distribution containing tools for working with MongoDB & the recommended way to work with MongoDB from Python.

- Install PyMongo :
 - Open terminal
 - write code `pip install pymongo`

1 Make a Connection

Make a Connection

```
Import pymongo as p  
Client = p. MongoClient  
("mongodb://localhost:27017/")
```

Show all database

```
print(myclient.list_database_names())
```

Sample code to make connection and show all databases in Python using pymongo

1 Make a Connection

Make a Connection

```
[3]: import pymongo as p  
     client = p.MongoClient('mongodb://localhost:27017')
```

Show all dbs name

```
[4]: print(client.list_database_names())  
     ['admin', 'config', 'local']
```

2 Create db & columns

```
Import pymongo as p  
Client = p. MongoClient ("mongodb://localhost:27017/")
```

```
# create database & collection
```

```
mydb = client["databaseku"]  
mycol = mydb["collectionku"]
```

```
# list of databases & collections
```

```
print(client.list_databases_names())  
print(mydb.list_collection_names())
```

Sample code in Python using pymongo:

- Create new database and collections
- Show list of databases and collections

2 Create db & columns

Create database & collection

```
[9]: mydb = client['databaseku']  
     mycol = mydb['collectionku']
```

List of databases & collections

```
[10]: print(client.list_database_names())  
      ['admin', 'config', 'databaseku', 'local']
```

```
[11]: print(mydb.list_collection_names())  
      ['collectionku']
```

3 Insert data

insert data to collection

```
mydata = {'nama': 'Andi', 'usia': '27'}
```

```
x = mycol.insert_one(mydata)
```

return inserted id

```
print(x.insert_id)
```

Sample code in Python using pymongo:

- Insert data to collection ('Collectionku')
- Return inserted id

3 Insert data

Insert Data to Collection

```
[12]: mydata = {'nama': 'Andi', 'usia': '27'}  
x = mycol.insert_one(mydata)  
x.inserted_id
```

```
[12]: ObjectId('601bad8114be8fb66ce6d4da')
```

```
[13]: mydata = {'nama': 'Budi', 'usia': '26'}  
x = mycol.insert_one(mydata)  
x.inserted_id
```

```
[13]: ObjectId('601bad8214be8fb66ce6d4db')
```

```
[14]: mydata = {'nama': 'Caca', 'usia': '29'}  
x = mycol.insert_one(mydata)  
x.inserted_id
```

```
[14]: ObjectId('601bad8214be8fb66ce6d4dc')
```


4 Find Data

```
# Find all data as list  
print(list(mycol.find()))
```

```
# print every data  
for x in mycol.find():  
    print(x)
```

Sample code in Python using pymongo:

- Find all data as list
- Print every data in collection

4 Find Data

Find All Data as List

```
[10]: print(list(mycol.find()))
```

```
[{'_id': ObjectId('601bb21c1911d2e9da27b4de'), 'nama': 'Andi', 'usia': '27'},  
{'_id': ObjectId('601bb21c1911d2e9da27b4df'), 'nama': 'Budi', 'usia': '26'},  
{'_id': ObjectId('601bb21d1911d2e9da27b4e0'), 'nama': 'Caca', 'usia': '29'}]
```

```
[11]: for x in mycol.find():  
       print(x)
```

```
{'_id': ObjectId('601bb21c1911d2e9da27b4de'), 'nama': 'Andi', 'usia': '27'}  
{'_id': ObjectId('601bb21c1911d2e9da27b4df'), 'nama': 'Budi', 'usia': '26'}  
{'_id': ObjectId('601bb21d1911d2e9da27b4e0'), 'nama': 'Caca', 'usia': '29'}
```

4 Find by Data Property

```
# find data property 'name' = 'Andi'
```

```
myquery = {'nama' : 'Andi'}
```

```
mydoc = mycol.find(myquery)
```

```
# print every data
```

```
for x in mydoc:
```

```
    print(x)
```

Sample code in Python using pymongo:

- Find data property
- Print data property

4 Find by Data Property

Find Data, name = 'Andi'

```
[18]: mydoc = mycol.find({'nama': 'Andi'})
```

```
for x in mydoc:  
    print(x)
```

```
{'_id': ObjectId('601927c4df7b3a9f387c49f7'), 'nama': 'Andi', 'usia': '27'}
```

5 Delete by data property

```
# Delete by data property 'name' = 'Andi'  
myquery = {'name' : 'Andi'}  
mycol.delete_one(myquery)  
  
# Print all data after delete  
print(list(mycol.find()))
```

Sample code in Python using pymongo:

- Delete by data property
- Print all data after delete

5 Delete by data property

Delete by Data Property

```
[26]: # Print all data before delete  
print(list(mycol.find()))
```

```
[{'_id': ObjectId('601bb7ba1911d2e9da27b4e4'), 'nama': 'Andi', 'usia': '27'},  
{'_id': ObjectId('601bb7bb1911d2e9da27b4e5'), 'nama': 'Budi', 'usia': '26'},  
{'_id': ObjectId('601bb7bb1911d2e9da27b4e6'), 'nama': 'Caca', 'usia': '29'}]
```

```
[27]: # Delete by data property 'name' = 'Andi'  
myquery = {'nama' : 'Andi'}  
mycol.delete_one(myquery)
```

```
# Print all data after delete  
print(list(mycol.find()))
```

```
[{'_id': ObjectId('601bb7bb1911d2e9da27b4e5'), 'nama': 'Budi', 'usia': '26'},  
{'_id': ObjectId('601bb7bb1911d2e9da27b4e6'), 'nama': 'Caca', 'usia': '29'}]
```

6 Update Data

Update data

```
myquery = {'nama' : "Budi"}  
newvalues = { '$set' : {"nama" : "Bisma"}}  
mycol.update_one(myquery, newvalues)
```

Print all data after update

```
print(list(mycol.find()))
```

Sample code in Python using pymongo:

- Update by data property
- Print all data after update

6 Update Data

Update Data

```
[36]: # Print all data before update
print(list(mycol.find()))
```

```
[{'_id': ObjectId('601bba9a1911d2e9da27b4e7'), 'nama': 'Budi', 'usia': '26'},
{'_id': ObjectId('601bba9a1911d2e9da27b4e8'), 'nama': 'Caca', 'usia': '29'}]
```

```
[37]: # Update data
myquery = {'nama': 'Budi'}
newvalues = {'$set' : {'nama' : 'Bisma'}}
mycol.update_one(myquery, newvalues)
```

```
# Print all data after update
print(list(mycol.find()))
```

```
[{'_id': ObjectId('601bba9a1911d2e9da27b4e7'), 'nama': 'Bisma', 'usia': '26'},
{'_id': ObjectId('601bba9a1911d2e9da27b4e8'), 'nama': 'Caca', 'usia': '29'}]
```