

Practice 2:











Python Development with MongoDB

Big Data System Design

Python Development Environment

❖ Python

- Object-oriented, high-level programming language
- It uses an interpreter
- It is platform-independent

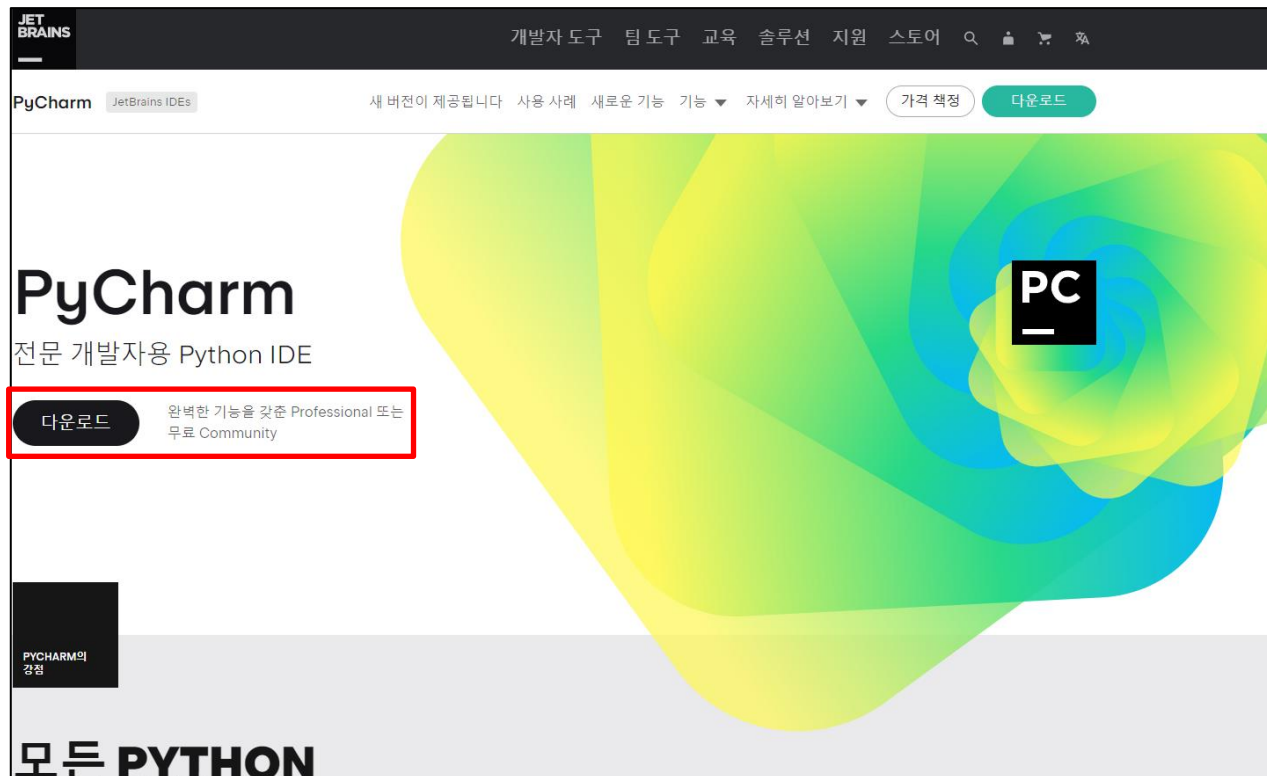
Jan 2024	Jan 2023	Change	Programming Language		Ratings	Change
1	1			Python	13.97%	-2.39%
2	2			C	11.44%	-4.81%
3	3			C++	9.96%	-2.95%
4	4			Java	7.87%	-4.34%
5	5			C#	7.16%	+1.43%
6	7	▲		JavaScript	2.77%	-0.11%
7	10	▲		PHP	1.79%	+0.40%
8	6	▼		Visual Basic	1.60%	-3.04%
9	8	▼		SQL	1.46%	-1.04%
10	20	▲		Scratch	1.44%	+0.86%

Source: <https://www.techrepublic.com/article/tiobe-index-language-rankings>

Python IDE

❖ PyCharm Installation

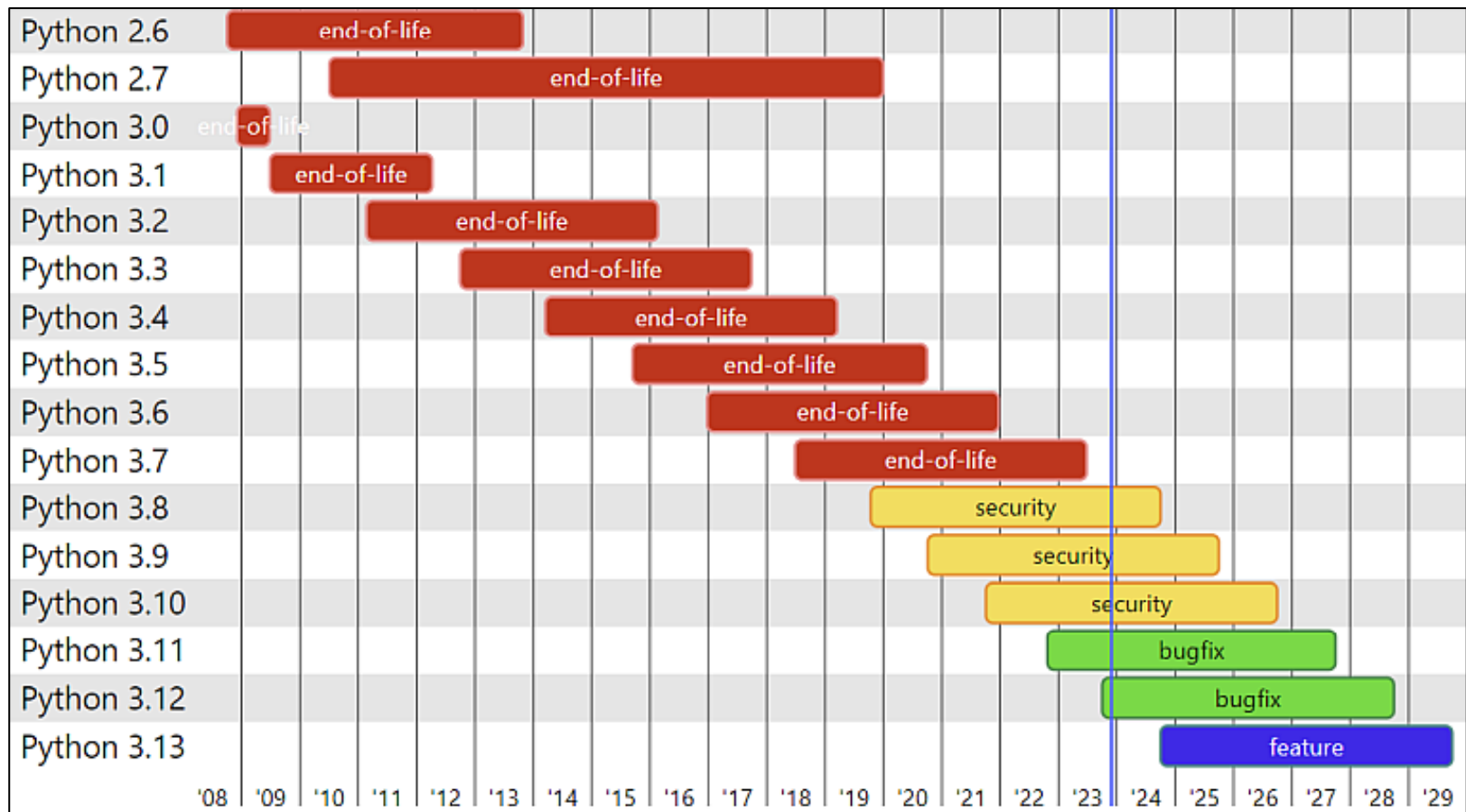
- Download Community Edition
- <https://www.jetbrains.com/ko-kr/pycharm>
- Alternative IDEs can also be utilized



Python Development Environment

❖ Before Python installation

- Python continues to be actively updated



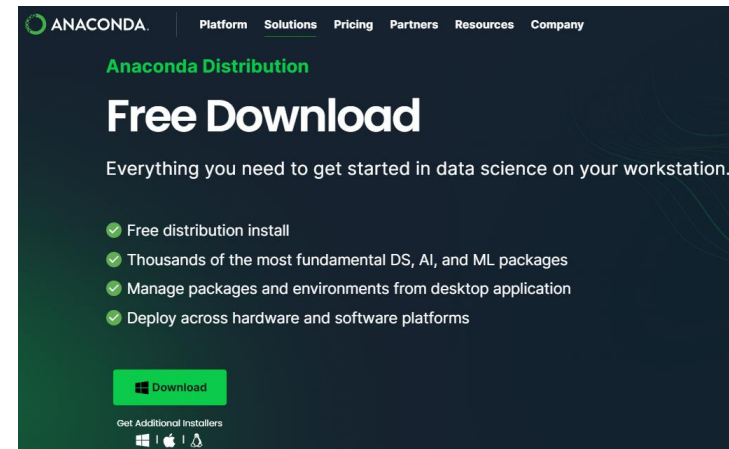
Python Development Environment

❖ Before Python installation

- Python is often installed in a virtual environment to manage its version

❖ Anaconda

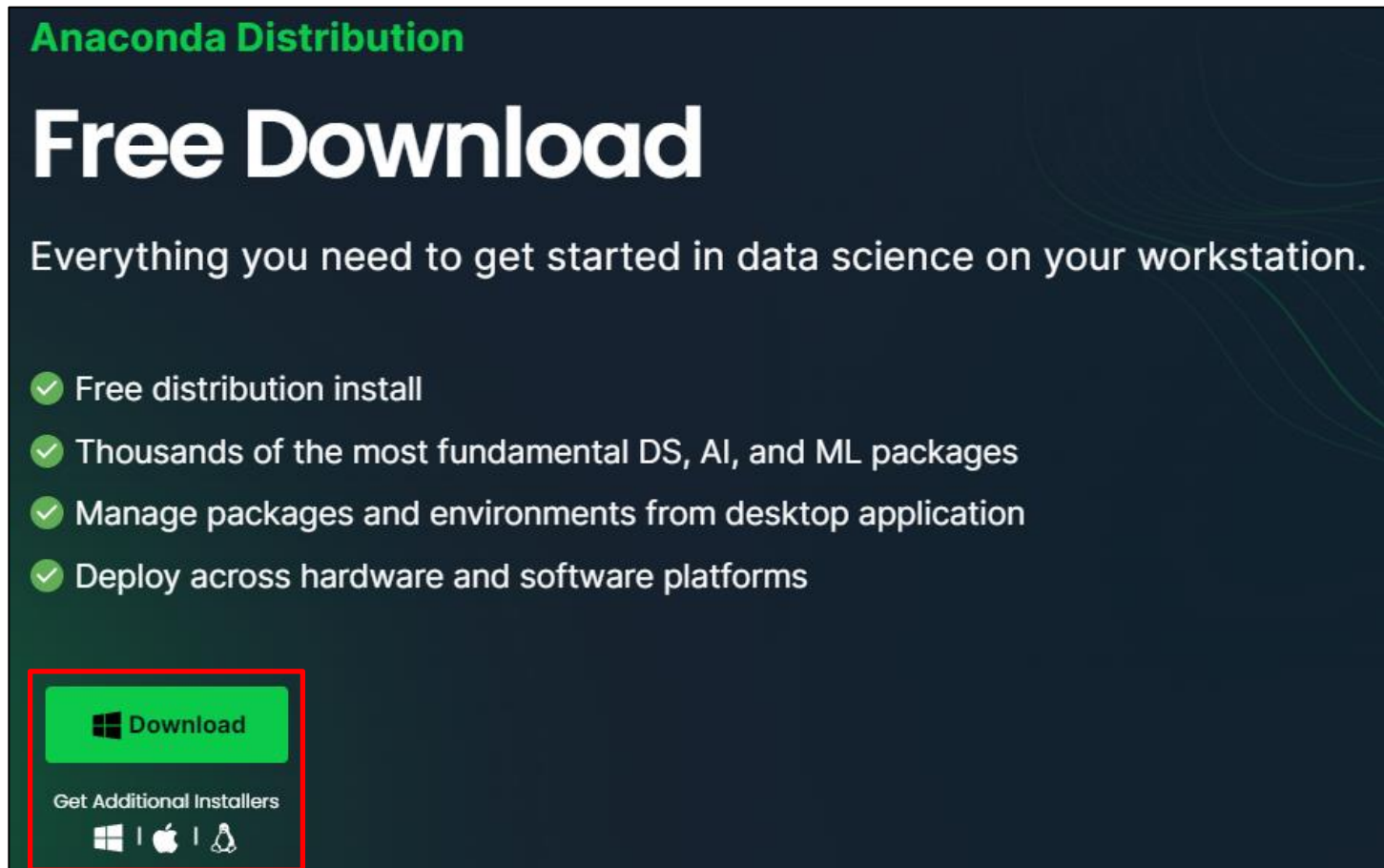
- A tool supporting package management, deployment, and virtual environment in Python
- Visit official website:
 - <https://www.anaconda.com/download>



Python Development Environment

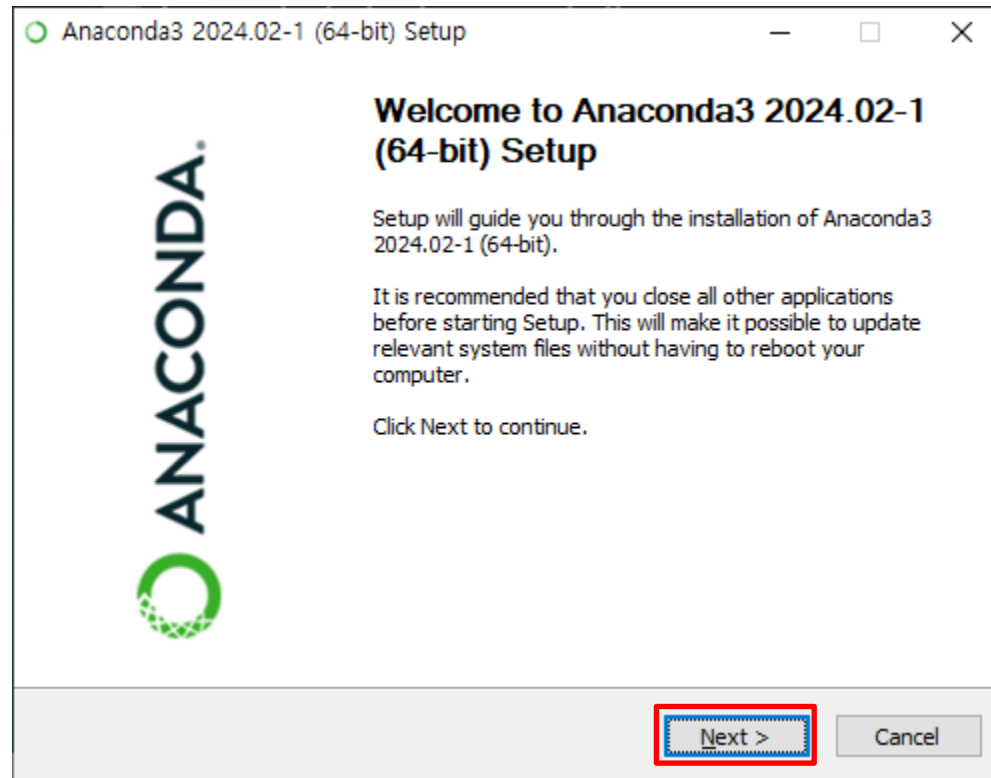
❖ Anaconda (cont'd)

- Download installation file



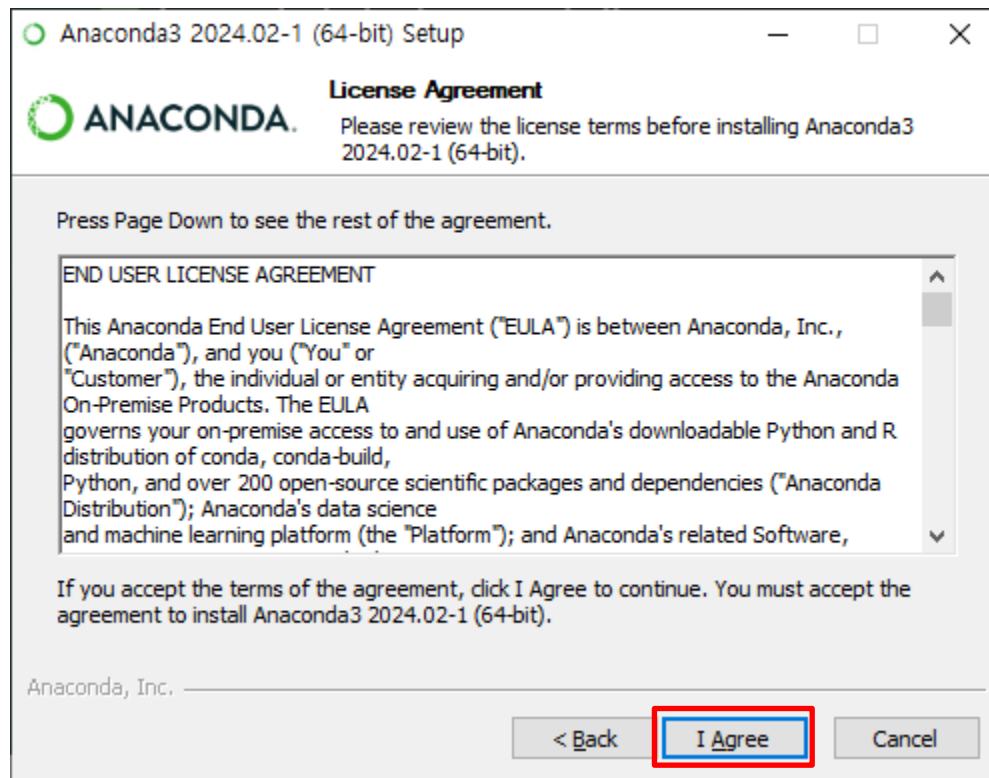
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



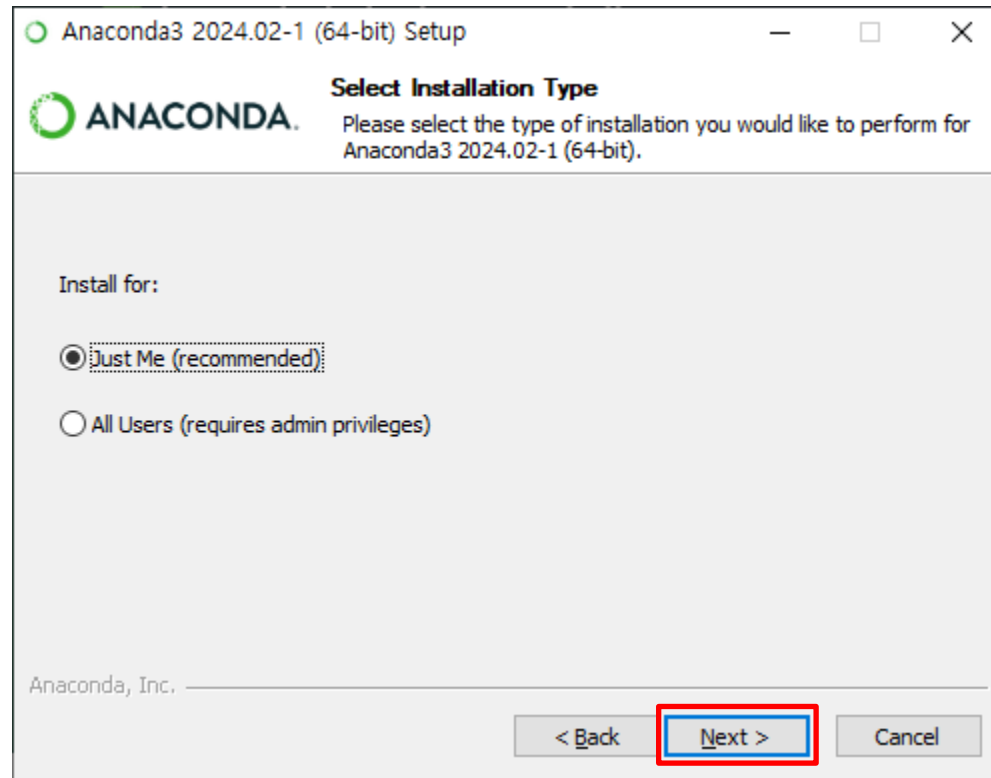
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



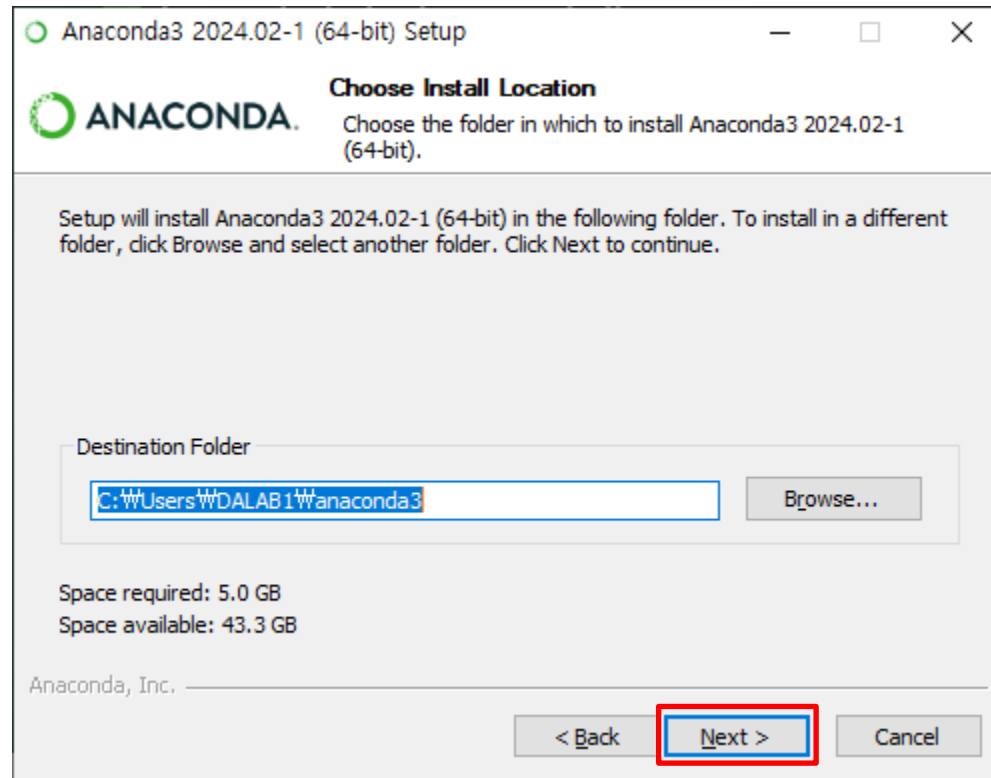
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



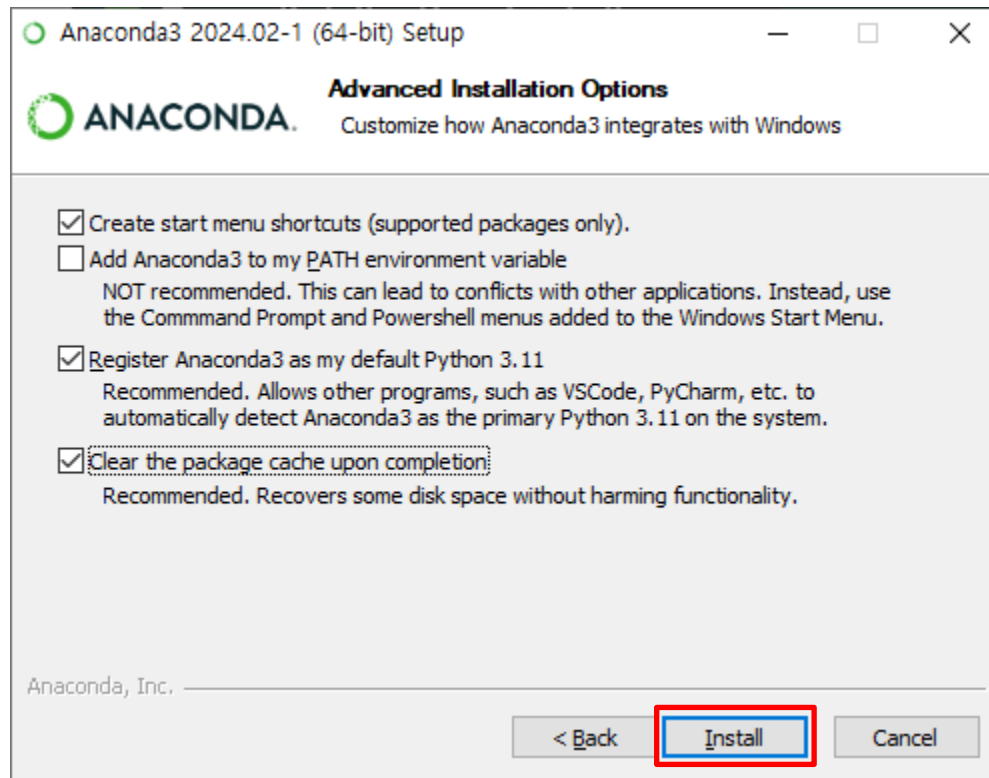
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



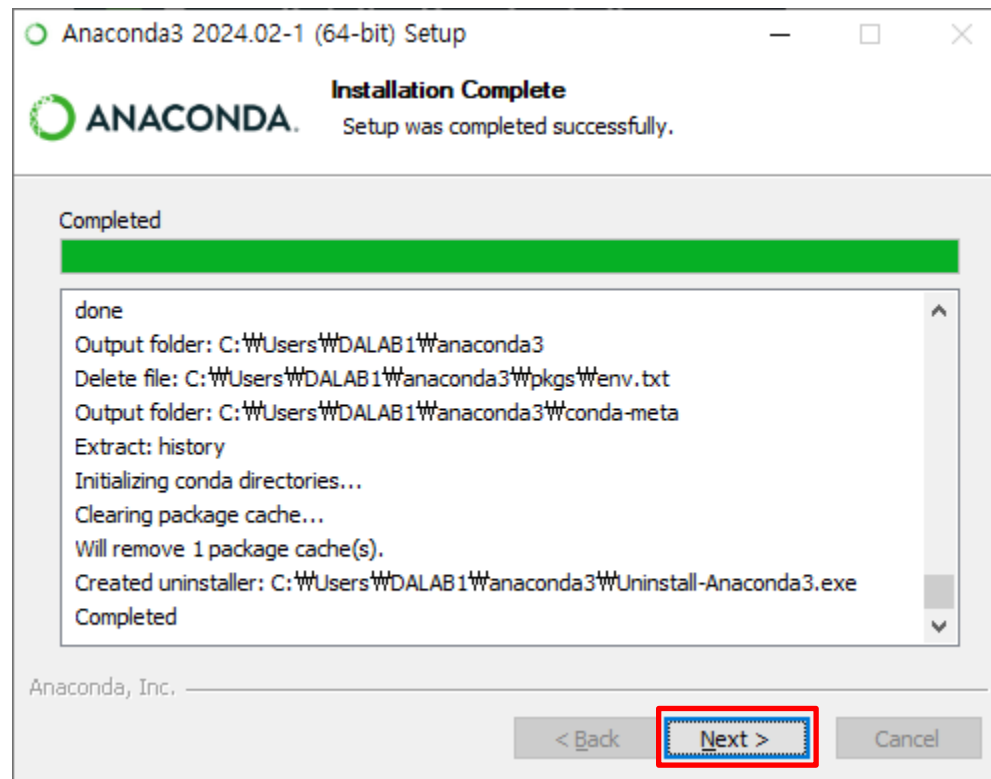
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



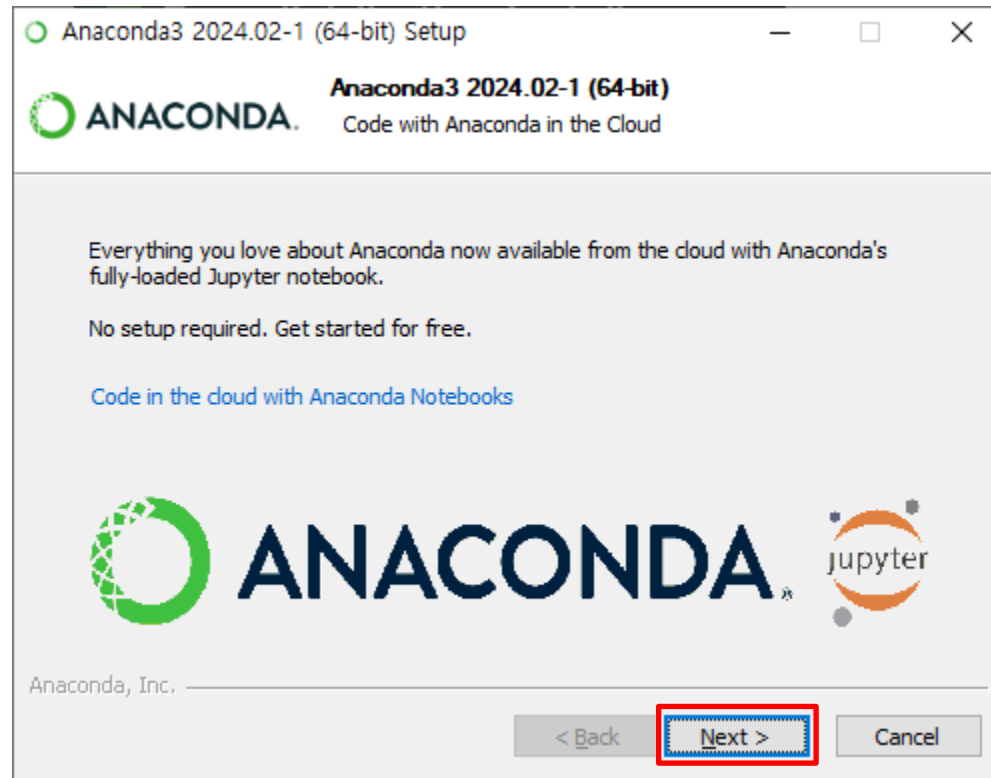
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



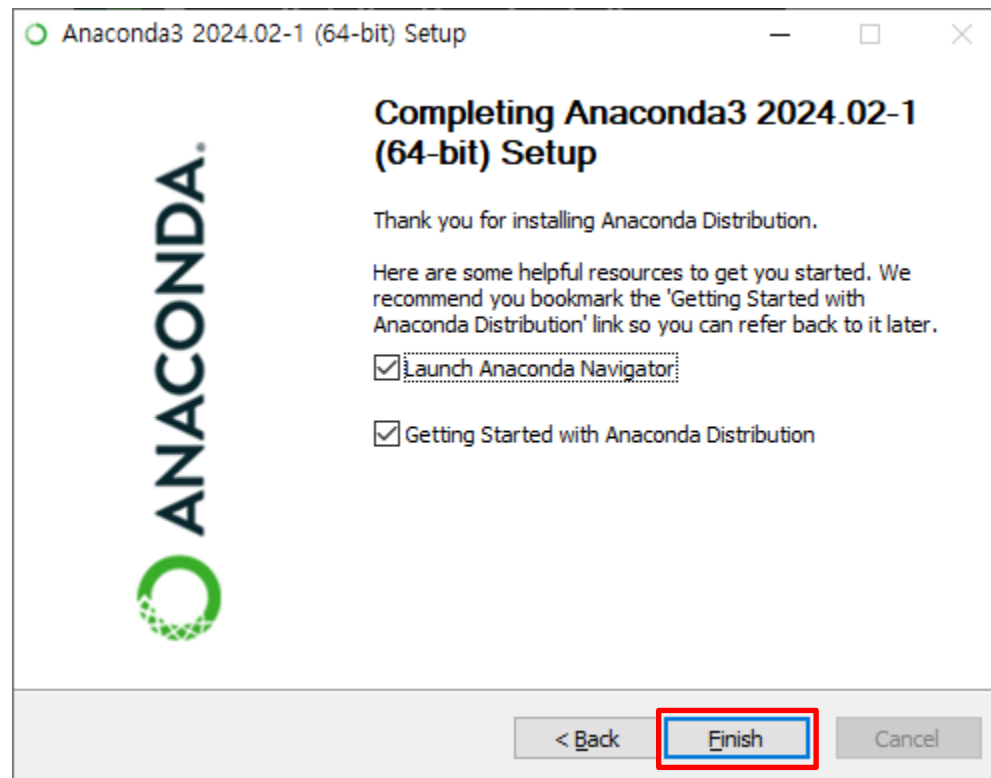
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



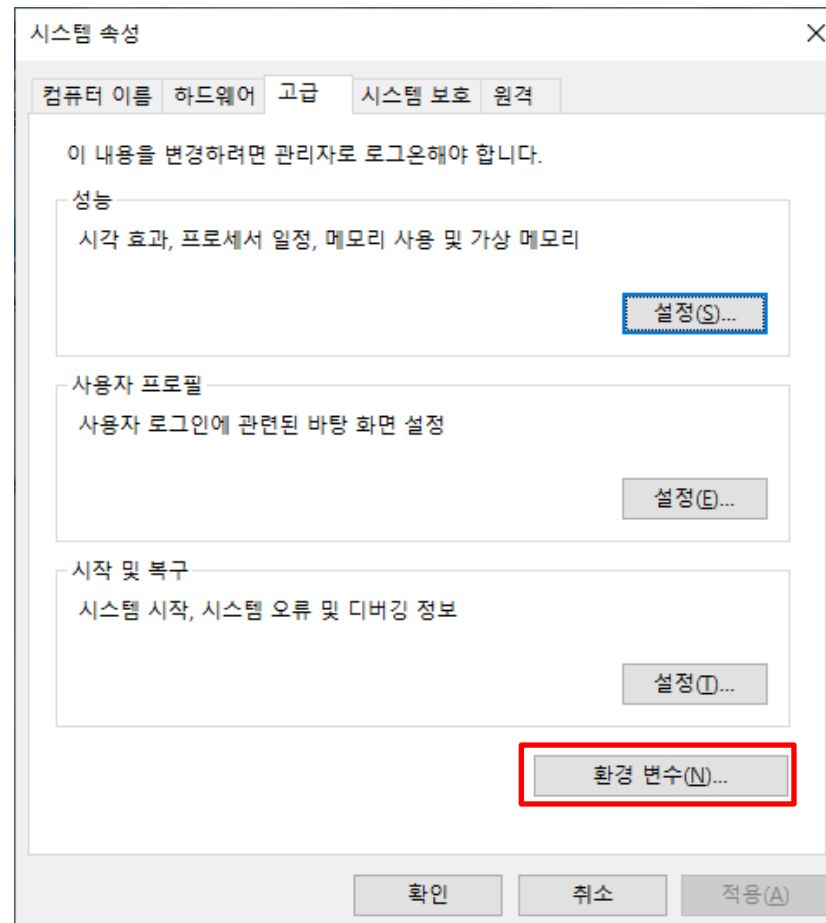
Python Development Environment

- ❖ Anaconda (cont'd)
 - Install Anaconda



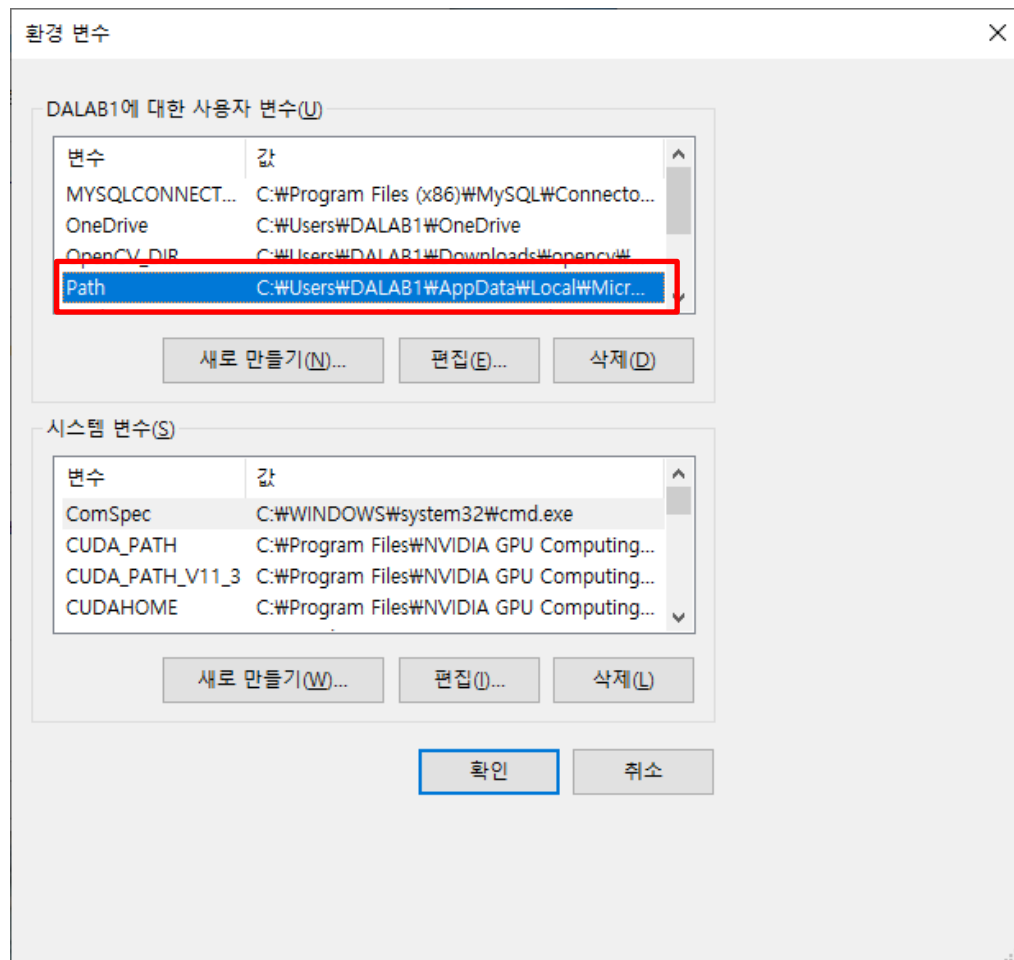
Python Development Environment

- ❖ Anaconda (cont'd)
 - PATH configuration



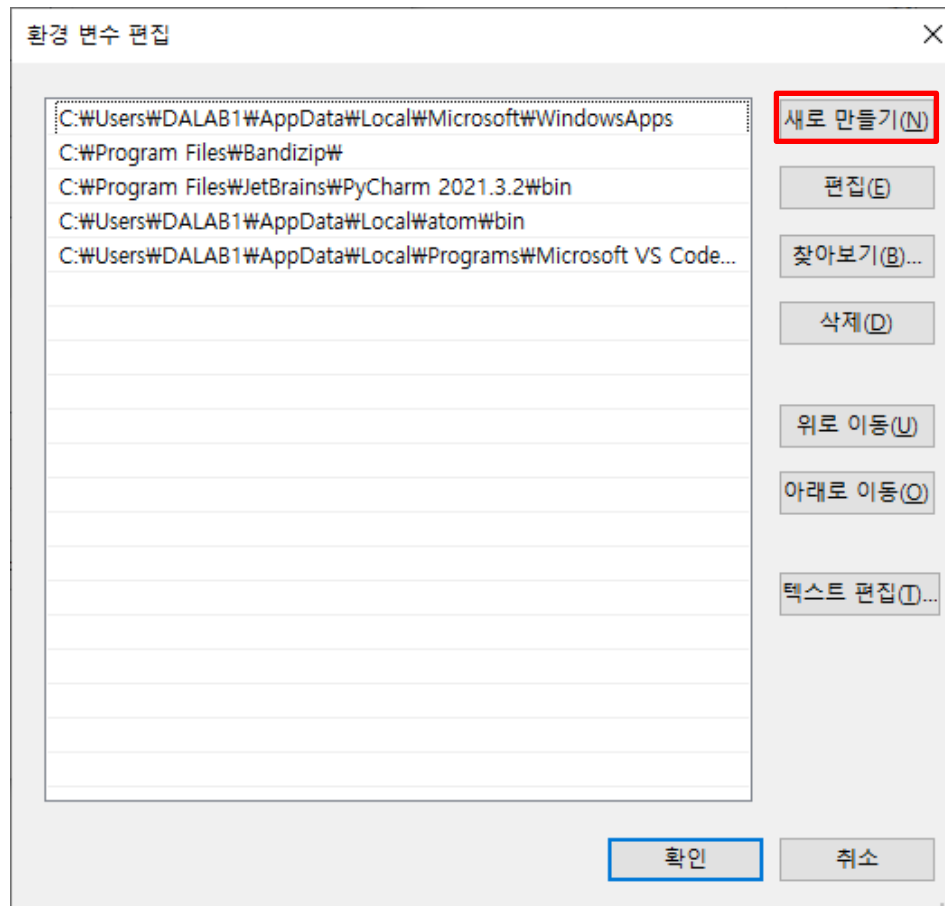
Python Development Environment

- ❖ Anaconda (cont'd)
 - PATH configuration



Python Development Environment

- ❖ Anaconda (cont'd)
 - PATH configuration

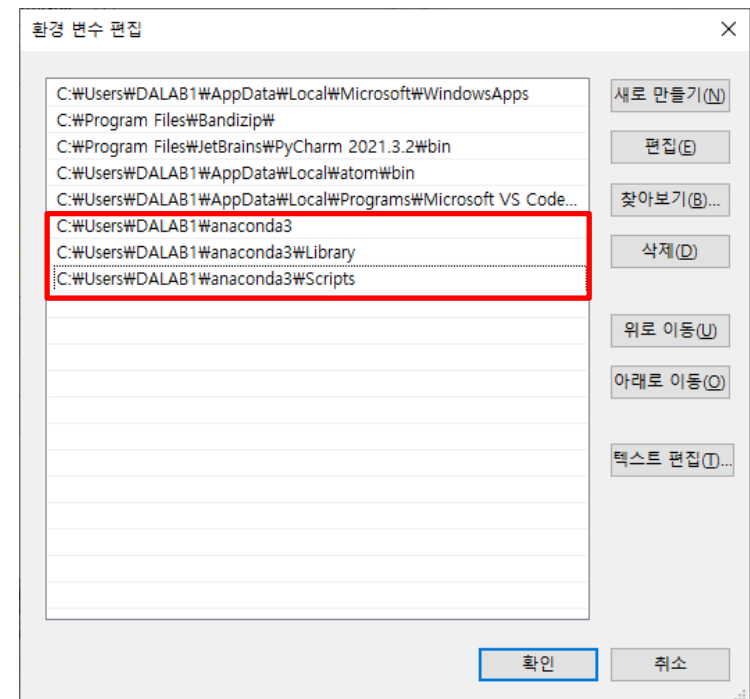


Python Development Environment

❖ Anaconda (cont'd)

▪ Add PATHs:

- C:\Users\%(User name)\anaconda3
- C:\Users\%(User name)\anaconda3\Library
- C:\Users\%(User name)\anaconda3\Scripts



Python Development Environment

❖ Anaconda (cont'd)

▪ Commands (1)

Command	Description
<code>conda -V</code> <code>conda --version</code>	Check Anaconda version
<code>python -V</code> <code>python --version</code>	Check Python version
<code>pip -V</code> <code>pip --version</code>	Check pip version
<code>conda update conda</code>	Update Anaconda to the latest version
<code>conda update python</code>	Update Python to the latest version
<code>python -m pip install --upgrade pip</code>	Update pip to the latest version
<code>conda update --all</code>	Update all packages to the latest version

Python Development Environment

❖ Anaconda (cont'd)

▪ Commands (2)

Command	Description
conda list	List of packages installed in the activated virtual environment
conda env list	List of virtual environments
conda search <package name>	Available package versions
conda create --name <virtual env name> python=3.x	Create a <virtual env name> virtual environment with Python 3.x
conda env remove --name <virtual env name>	Remove a <virtual env name> virtual environment
conda install <package name>	Install the <package name> package in the activated virtual environment
conda remove <package name>	Remove the <package name> package in the activated virtual environment
conda install --name <virtual env name> <package name>==[version]	Install the <package name> package [version] in the <virtual env name> virtual environment
conda remove --name <virtual env name> <package name>	Remove the <package name> package in the <virtual env name> virtual environment

Python Development Environment

❖ Anaconda (cont'd)

▪ Commands (3)

Command	Description
activate <virtual env name>	Activate <virtual env name> virtual environment
conda deactivate	Deactivate the activated virtual environment

Python Development Environment

❖ Anaconda (cont'd)

- Create virtual environment
 - Run Anaconda Prompt
 - `conda --name practice python=3.12`

```
(base) C:\Users\DALAB1>conda create --name practice python=3.12
Channels:
  - defaults
Platform: win-64
Collecting package metadata (repodata.json): done
Solving environment: done

## Package Plan ##

  environment location: C:\Users\DALAB1\anaconda3\envs\practice

added / updated specs:
  - python=3.12
```

Python Development Environment

- ❖ Anaconda (cont'd)
 - Create virtual environment
 - `conda env list`

```
(base) C:\Users\DALAB1>conda env list
# conda environments:
#
base                *  C:\Users\DALAB1\anaconda3
practice            C:\Users\DALAB1\anaconda3\envs\practice
```

Python Development Environment

- ❖ Anaconda (cont'd)
 - Create virtual environment
 - conda deactivate

```
(base) C:\Users\DALAB1>conda deactivate  
C:\Users\DALAB1>
```


Python Development Environment

- ❖ Anaconda (cont'd)
 - Create virtual environment
 - conda activate practice

```
C:\Users\DALAB1>conda activate practice  
(practice) C:\Users\DALAB1>
```

Python Development Environment

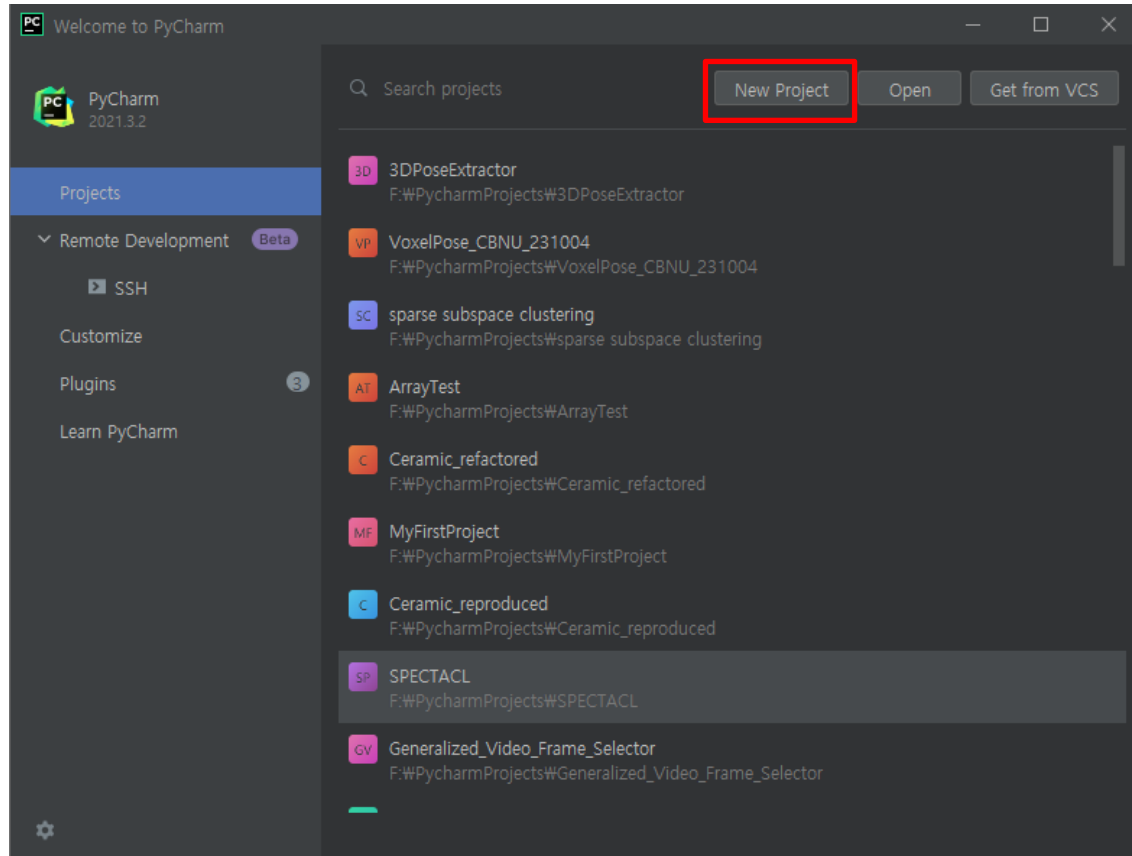
- ❖ Anaconda (cont'd)
 - Create virtual environment
 - `python --version`

```
(practice) C:\Users\DALAB1>python --version  
Python 3.12.2
```

Python Development Environment

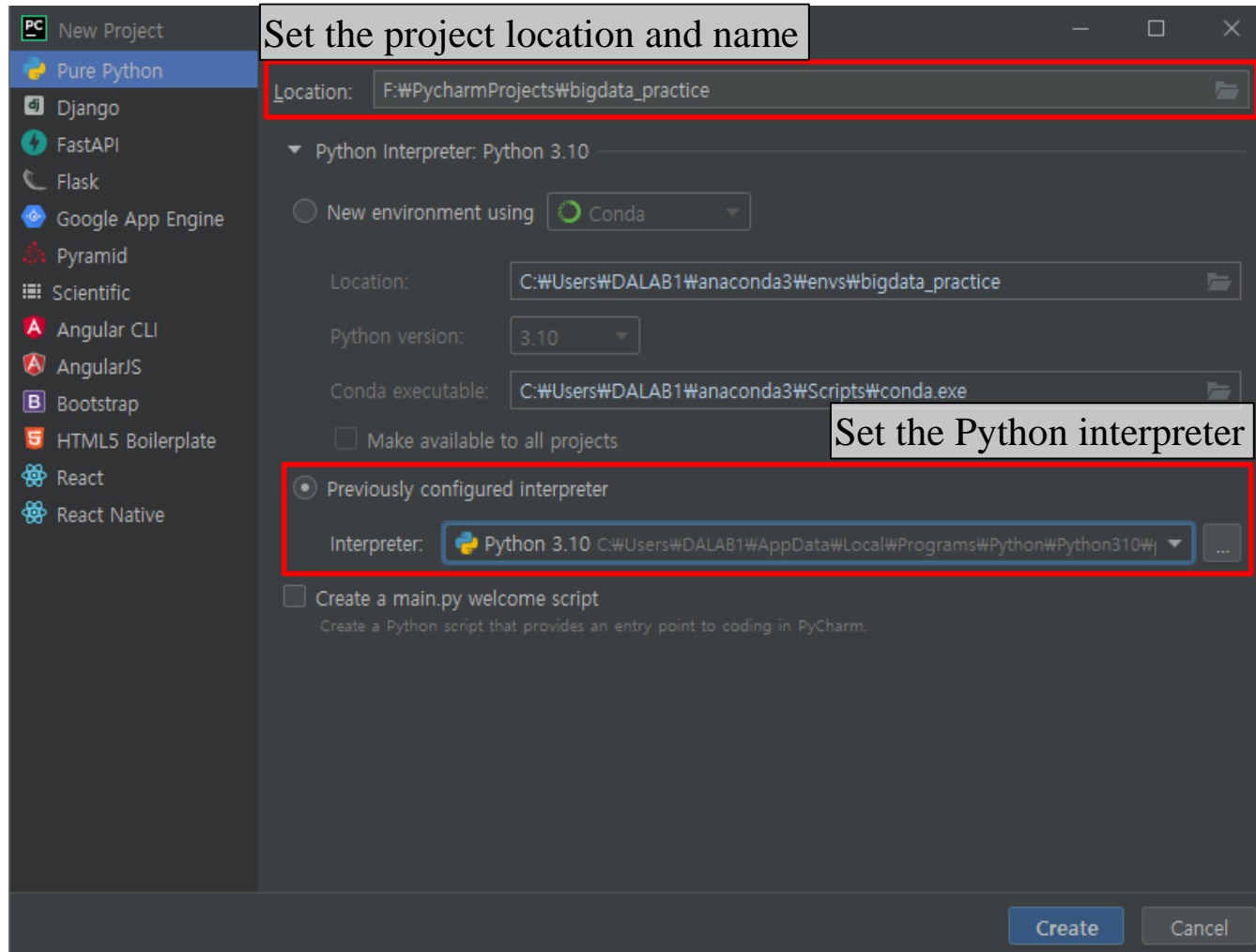
❖ Integrate PyCharm with Anaconda

- Run PyCharm
- Projects – New Project



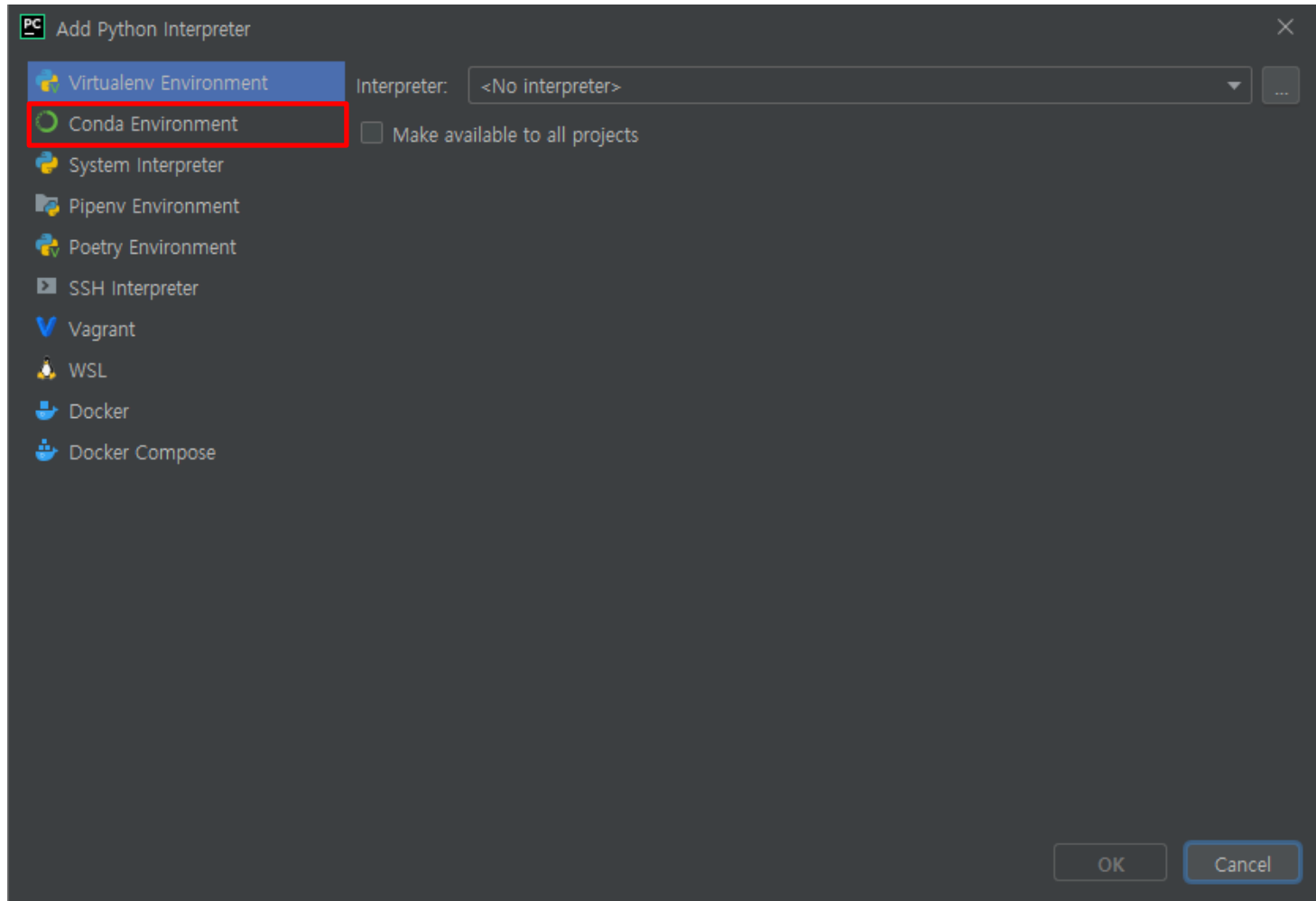
Python Development Environment

❖ Integrate PyCharm with Anaconda (cont'd)



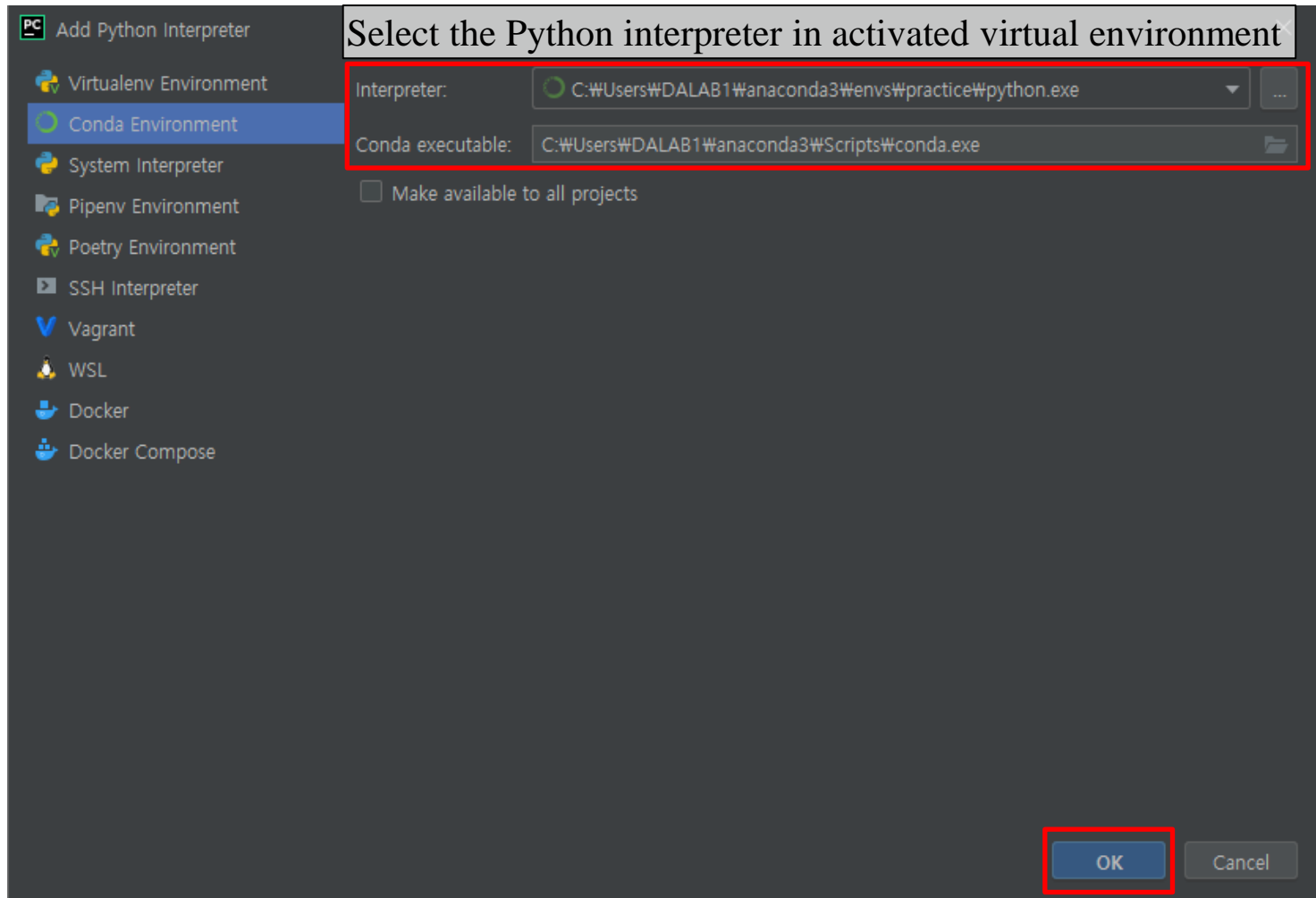
Python Development Environment

❖ Integrate PyCharm with Anaconda (cont'd)



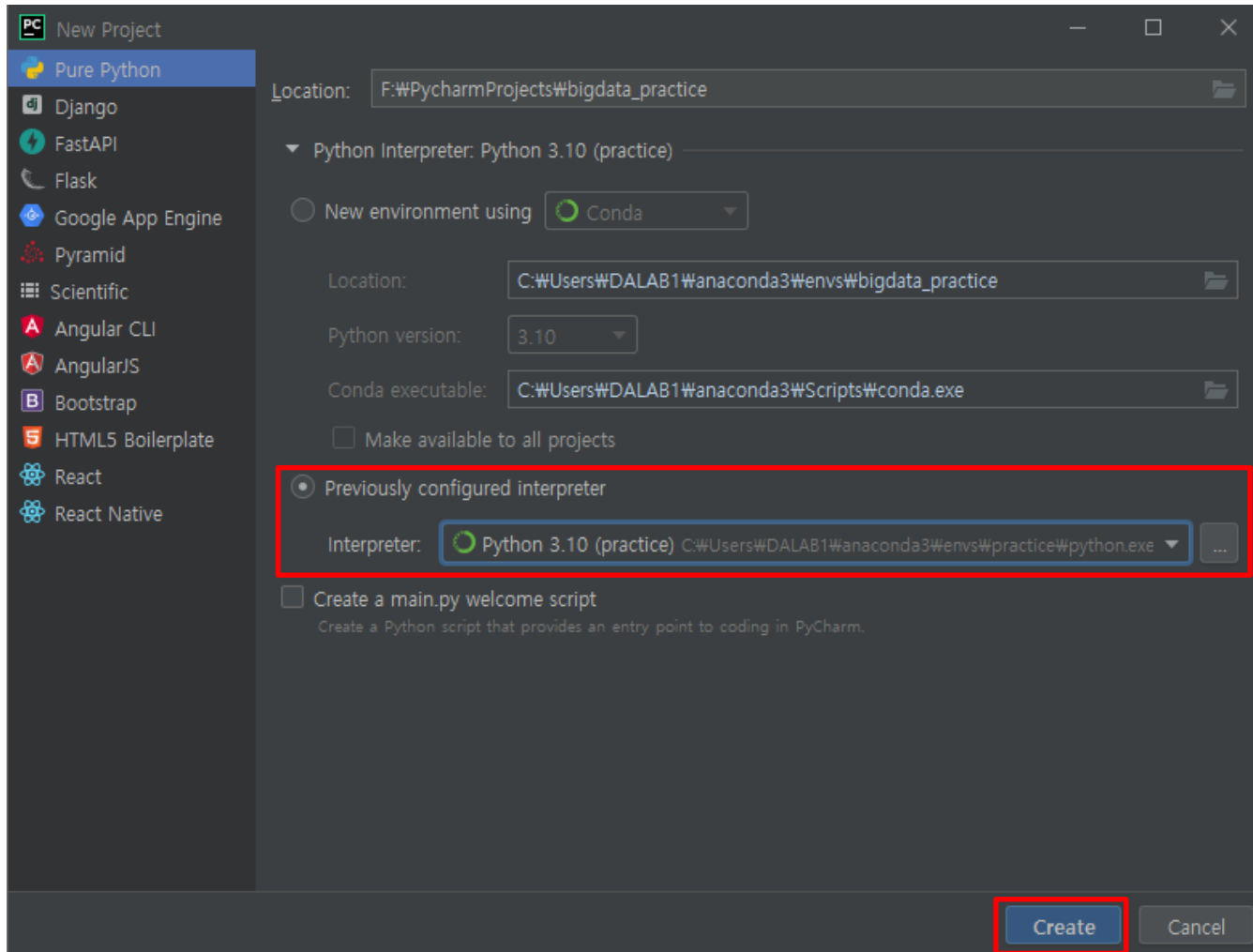
Python Development Environment

❖ Integrate PyCharm with Anaconda (cont'd)



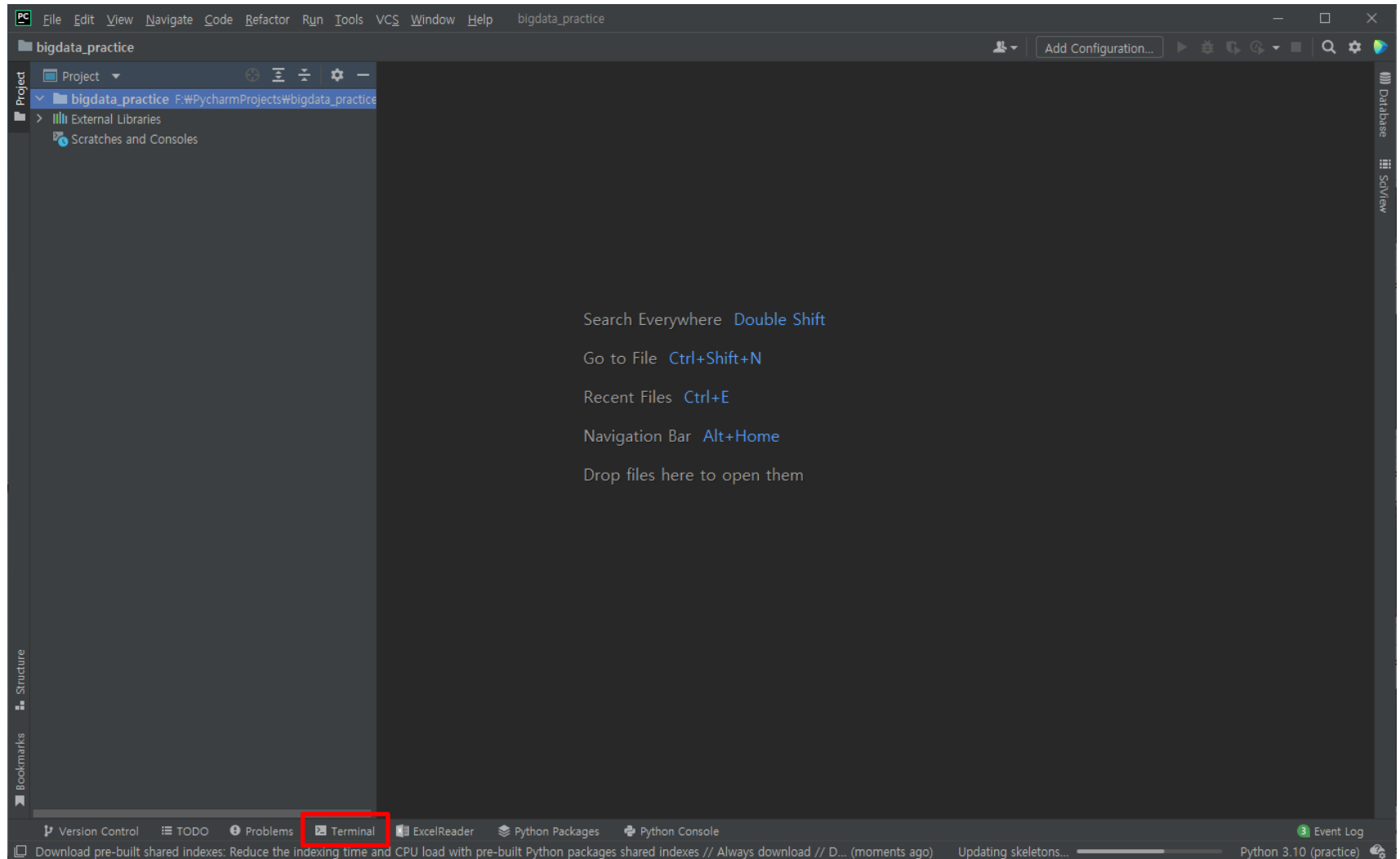
Python Development Environment

❖ Integrate PyCharm with Anaconda (cont'd)



Python Development Environment

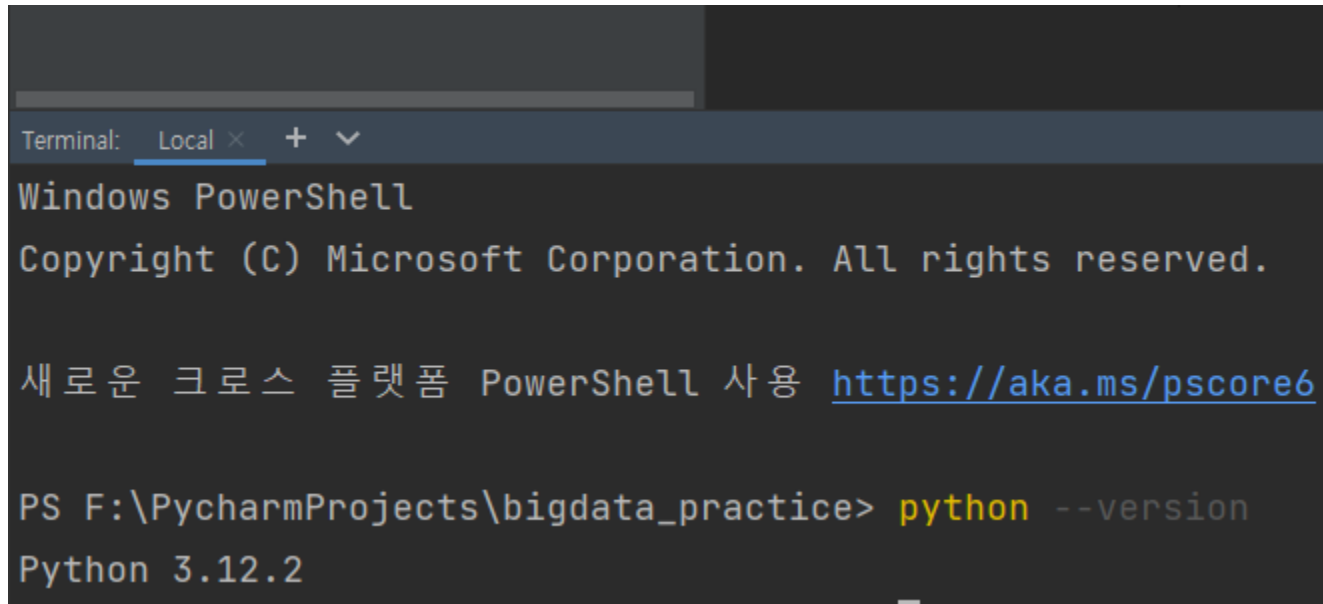
❖ Integrate PyCharm with Anaconda (cont'd)



Python Development Environment

❖ Integrate PyCharm with Anaconda (cont'd)

- `python --version`



```
Terminal: Local x + v
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS F:\PycharmProjects\bigdata_practice> python --version
Python 3.12.2
```

Python Development Environment

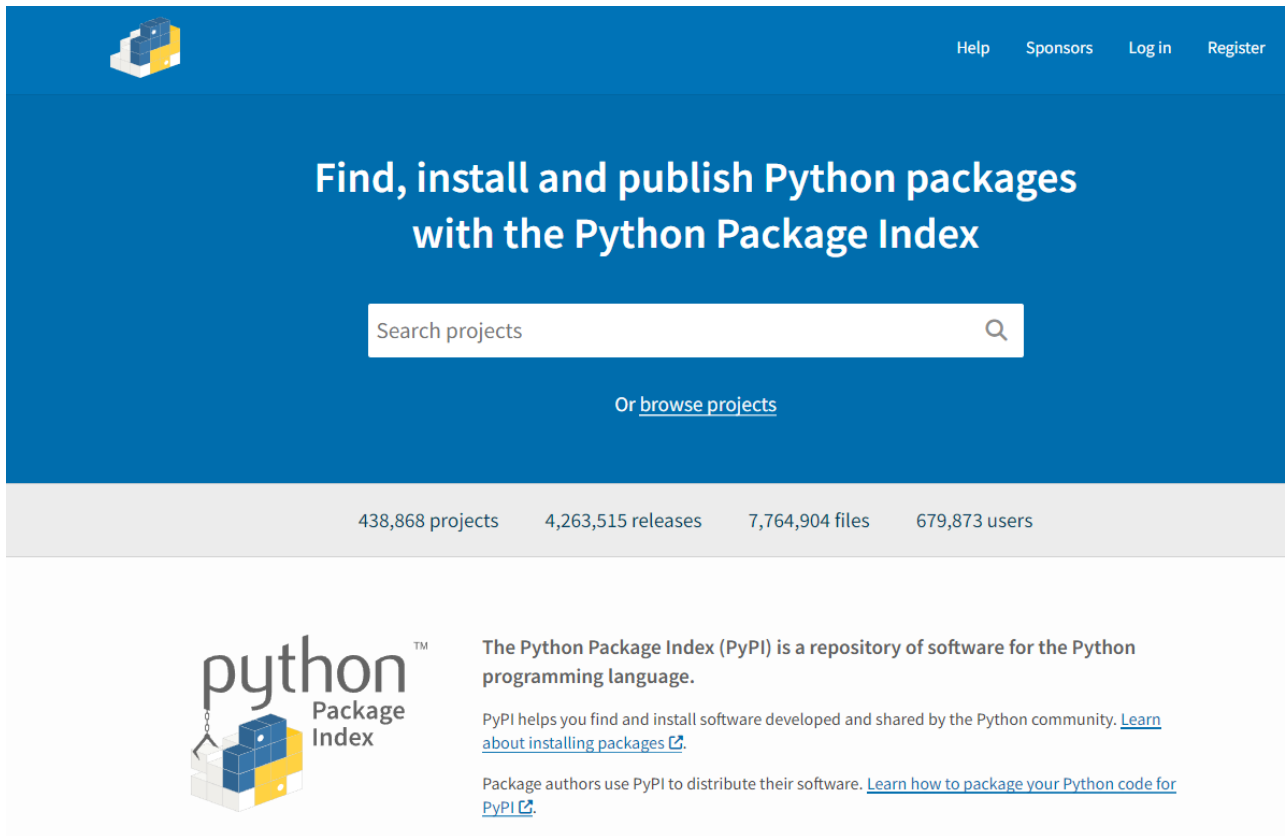
- ❖ First Python implementation with MongoDB
 - Install the interface library for accessing MongoDB
 - pip list

```
PS F:\PycharmProjects\bigdata_practice> pip list
Package      Version
-----
pip          23.3.1
setuptools   68.2.2
wheel        0.41.2
PS F:\PycharmProjects\bigdata_practice> 
```

Python Development Environment

❖ What is “pip”?

- Management tool for software packages developed in Python
- Python Package Index: PyPI (<https://pypi.org/>)



The screenshot shows the homepage of the Python Package Index (PyPI). The header is blue with the PyPI logo on the left and links for Help, Sponsors, Log in, and Register on the right. The main content area is also blue and features the text "Find, install and publish Python packages with the Python Package Index". Below this is a search bar with the placeholder text "Search projects" and a magnifying glass icon. Under the search bar is a link that says "Or [browse projects](#)". A light gray bar below the main content area displays statistics: "438,868 projects", "4,263,515 releases", "7,764,904 files", and "679,873 users". The footer is white and contains the "python Package Index" logo on the left. To the right of the logo, there is text explaining that PyPI is a repository of software for the Python programming language, followed by two links: "[PyPI helps you find and install software developed and shared by the Python community. \[Learn about installing packages\]\(#\)](#)" and "[Package authors use PyPI to distribute their software. \[Learn how to package your Python code for PyPI\]\(#\)](#)".

438,868 projects 4,263,515 releases 7,764,904 files 679,873 users

python Package Index™

The Python Package Index (PyPI) is a repository of software for the Python programming language.

PyPI helps you find and install software developed and shared by the Python community. [Learn about installing packages](#)

Package authors use PyPI to distribute their software. [Learn how to package your Python code for PyPI](#)

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Install the interface library for accessing MongoDB
 - `python -m pip install --upgrade pip`

```
2.1/2.1 MB 7.9 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.3.1
    Uninstalling pip-23.3.1:
      Successfully uninstalled pip-23.3.1
Successfully installed pip-24.0
```

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Install the interface library for accessing MongoDB
 - `pip install pymongo`

```
PS F:\PycharmProjects\bigdata_practice> pip install pymongo
Collecting pymongo
  Downloading pymongo-4.6.2-cp312-cp312-win_amd64.whl.metadata (22 kB)
Collecting dnspython<3.0.0,>=1.16.0 (from pymongo)
  Downloading dnspython-2.6.1-py3-none-any.whl.metadata (5.8 kB)
Downloading pymongo-4.6.2-cp312-cp312-win_amd64.whl (472 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 473.0/473.0 kB 5.9 MB/s eta 0:00:00
Downloading dnspython-2.6.1-py3-none-any.whl (307 kB)
  ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 307.7/307.7 kB 9.3 MB/s eta 0:00:00
Installing collected packages: dnspython, pymongo
Successfully installed dnspython-2.6.1 pymongo-4.6.2
PS F:\PycharmProjects\bigdata_practice> 
```

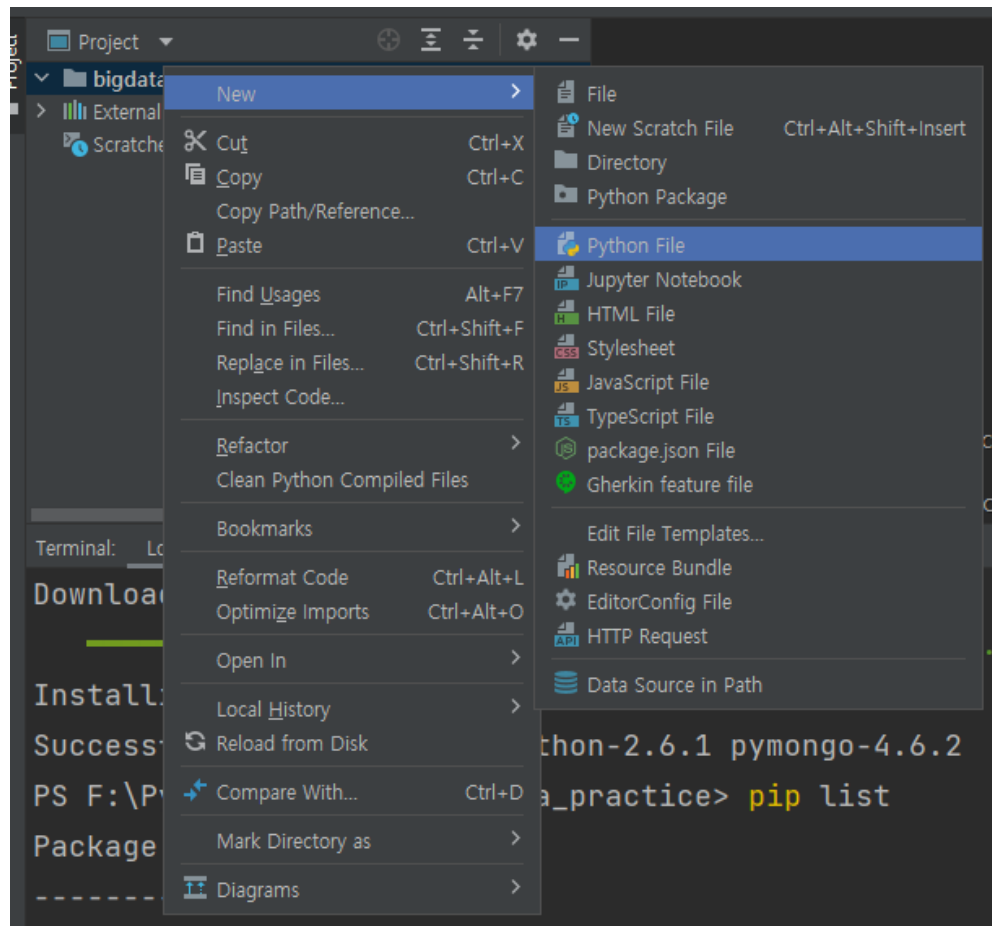
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Install the interface library for accessing MongoDB
 - (again) pip list

```
PS F:\PycharmProjects\bigdata_practice> pip list
Package      Version
-----
dnspython    2.6.1
pip          24.0
pymongo      4.6.2
setuptools   68.2.2
wheel        0.41.2
PS F:\PycharmProjects\bigdata_practice>
```

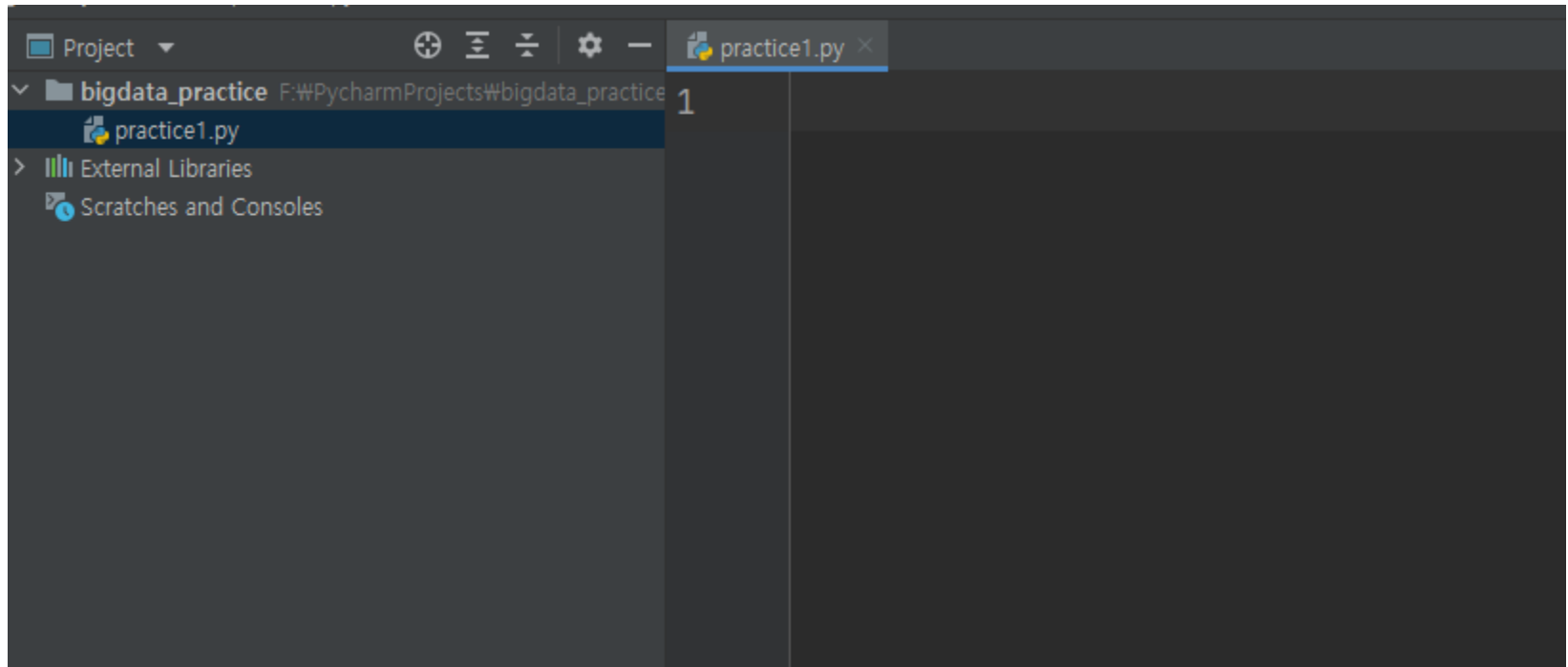
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - File – New – Python File



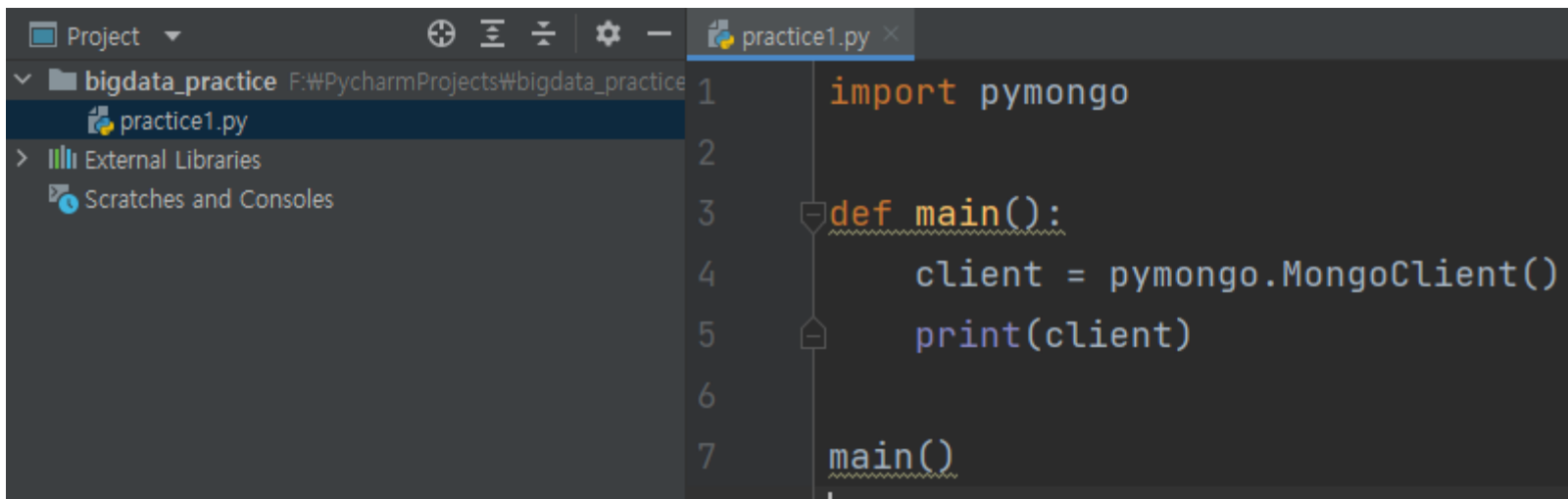
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Create 'practice1.py' file



Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Establish connection to localhost MongoDB server using the pymongo



The screenshot shows the PyCharm IDE interface. On the left, the 'Project' tool window displays a folder named 'bigdata_practice' with a file 'practice1.py' inside. Below it, 'External Libraries' and 'Scratches and Consoles' are visible. The main editor window shows the code for 'practice1.py' with line numbers 1 through 7. The code imports the 'pymongo' module, defines a 'main()' function that creates a 'MongoClient' and prints it, and then calls 'main()' at the bottom.

```
1 import pymongo
2
3 def main():
4     client = pymongo.MongoClient()
5     print(client)
6
7 main()
```

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - python practice1.py

```
PS F:\PycharmProjects\bigdata_practice> python practice1.py
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)
PS F:\PycharmProjects\bigdata_practice> 
```

Python Development Environment

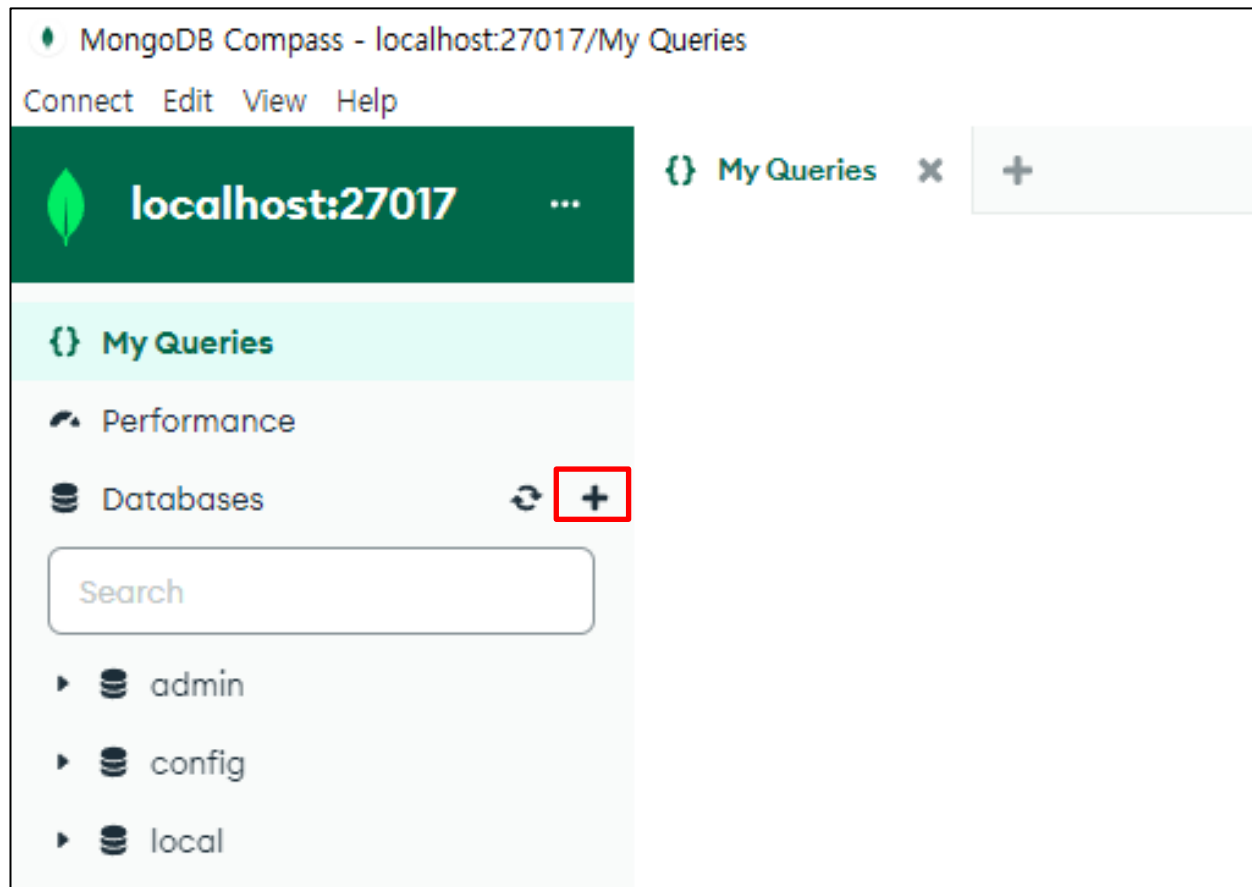
- ❖ First Python implementation with MongoDB (cont'd)
 - Show list of databases in localhost server

```
practice1.py x
1  import pymongo
2
3  def main():
4      client = pymongo.MongoClient()
5      print(client)
6
7      for db in client.list_databases():
8          print(db)
9
10  main()
```

```
PS F:\PycharmProjects\bigdata_practice> python practice1.py
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)
{'name': 'admin', 'sizeOnDisk': 40960, 'empty': False}
{'name': 'config', 'sizeOnDisk': 73728, 'empty': False}
{'name': 'local', 'sizeOnDisk': 73728, 'empty': False}
```

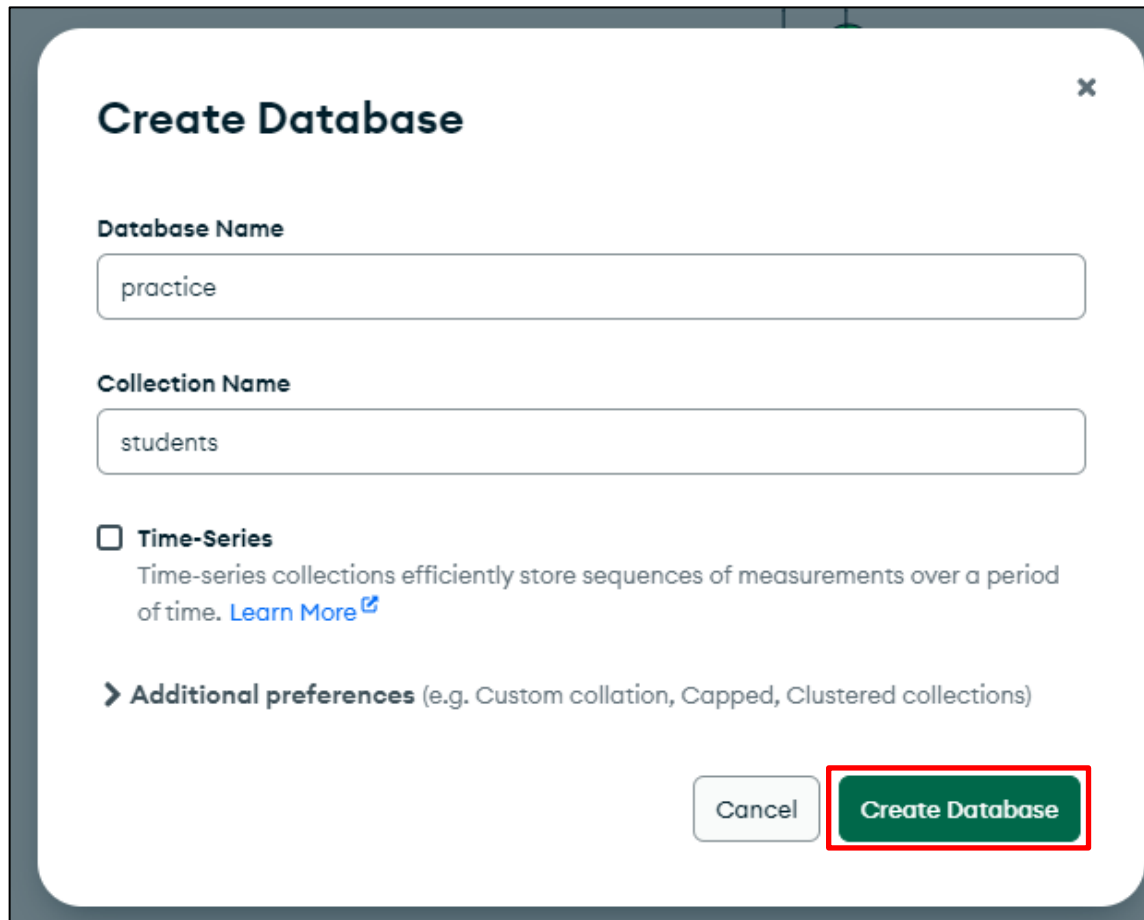
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Create a new database in localhost server



Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Create a new database in localhost server



The image shows a 'Create Database' dialog box from MongoDB. It has a title bar with a close button (X). The dialog contains two text input fields: 'Database Name' with the value 'practice' and 'Collection Name' with the value 'students'. Below these is a checkbox labeled 'Time-Series' which is unchecked. A description for 'Time-Series' is provided: 'Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)'. At the bottom, there is a section for 'Additional preferences' with a chevron icon and a description '(e.g. Custom collation, Capped, Clustered collections)'. At the bottom right, there are two buttons: 'Cancel' and 'Create Database'. The 'Create Database' button is highlighted with a red rectangular border.

Create Database

Database Name
practice

Collection Name
students

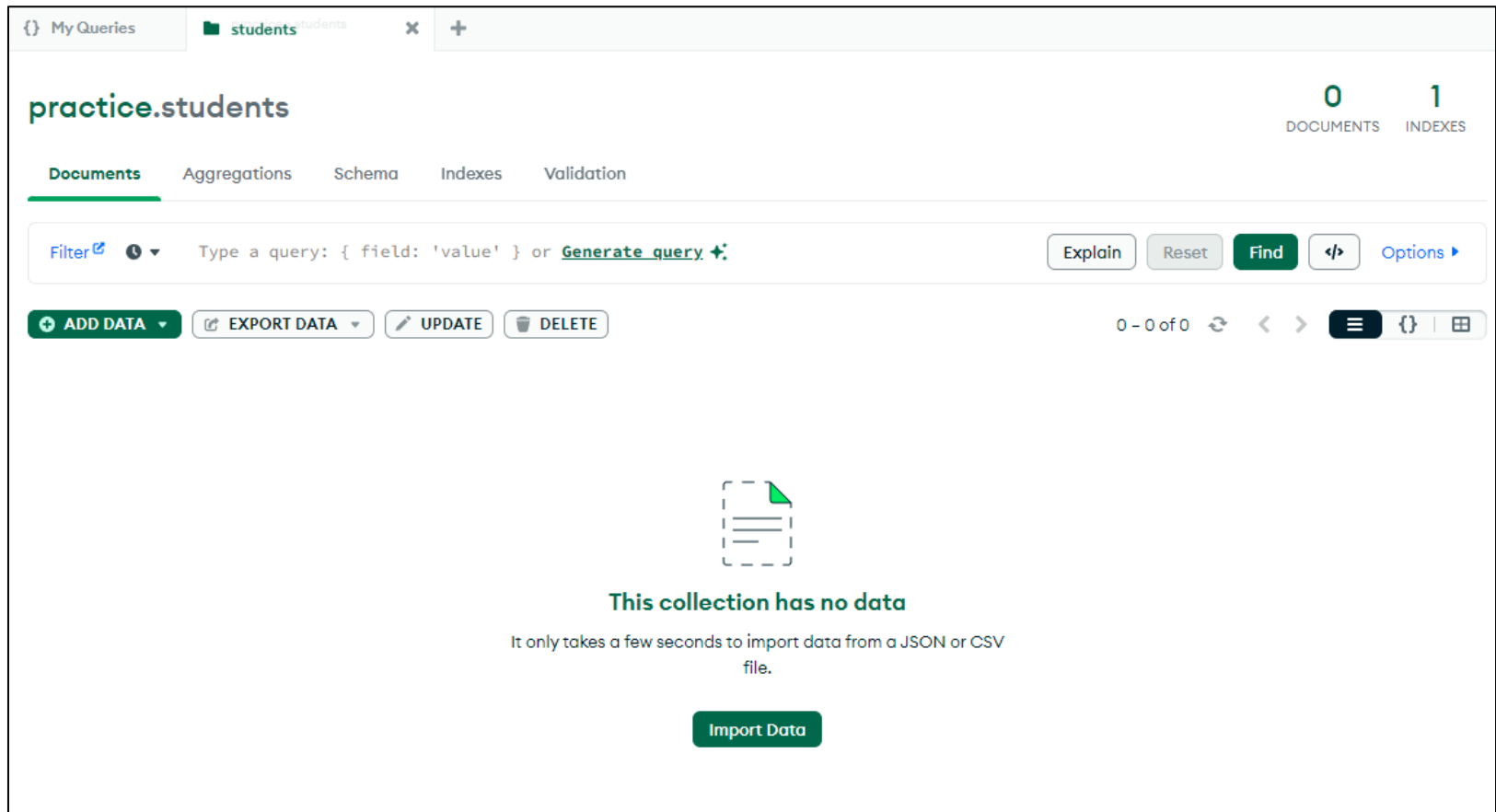
☐ **Time-Series**
Time-series collections efficiently store sequences of measurements over a period of time. [Learn More](#)

> **Additional preferences** (e.g. Custom collation, Capped, Clustered collections)

Cancel Create Database

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Create a new database in localhost server



Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - (again) Show list of databases in localhost server

```
practice1.py x
1  import pymongo
2
3  def main():
4      client = pymongo.MongoClient()
5      print(client)
6
7      for db in client.list_databases():
8          print(db)
9
10 main()
```

```
PS F:\PycharmProjects\bigdata_practice> python practice1.py
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)
{'name': 'admin', 'sizeOnDisk': 40960, 'empty': False}
{'name': 'config', 'sizeOnDisk': 110592, 'empty': False}
{'name': 'local', 'sizeOnDisk': 73728, 'empty': False}
{'name': 'practice', 'sizeOnDisk': 8192, 'empty': False}
```

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Access a specific database

```
practice1.py x
1      import pymongo
2
3      def main():
4          client = pymongo.MongoClient()
5          print(client)
6
7          for db in client.list_databases():
8              print(db)
9
10         db_conn = client.get_database("practice")
11         print(db_conn)
12
13     main()
```


Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Access a specific database

```
PS F:\PycharmProjects\bigdata_practice> python practice1.py
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True)
{'name': 'admin', 'sizeOnDisk': 40960, 'empty': False}
{'name': 'config', 'sizeOnDisk': 110592, 'empty': False}
{'name': 'local', 'sizeOnDisk': 73728, 'empty': False}
{'name': 'practice', 'sizeOnDisk': 8192, 'empty': False}
Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True), 'practice')
```

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Show list of collections in the selected database

```
practice1.py x
1  import pymongo
2
3  def main():
4      client = pymongo.MongoClient()
5      print(client)
6
7      for db in client.list_databases():
8          print(db)
9
10     db_conn = client.get_database("practice")
11     print(db_conn)
12
13     for col in db_conn.list_collection_names():
14         print(col)
15
16     main()
```

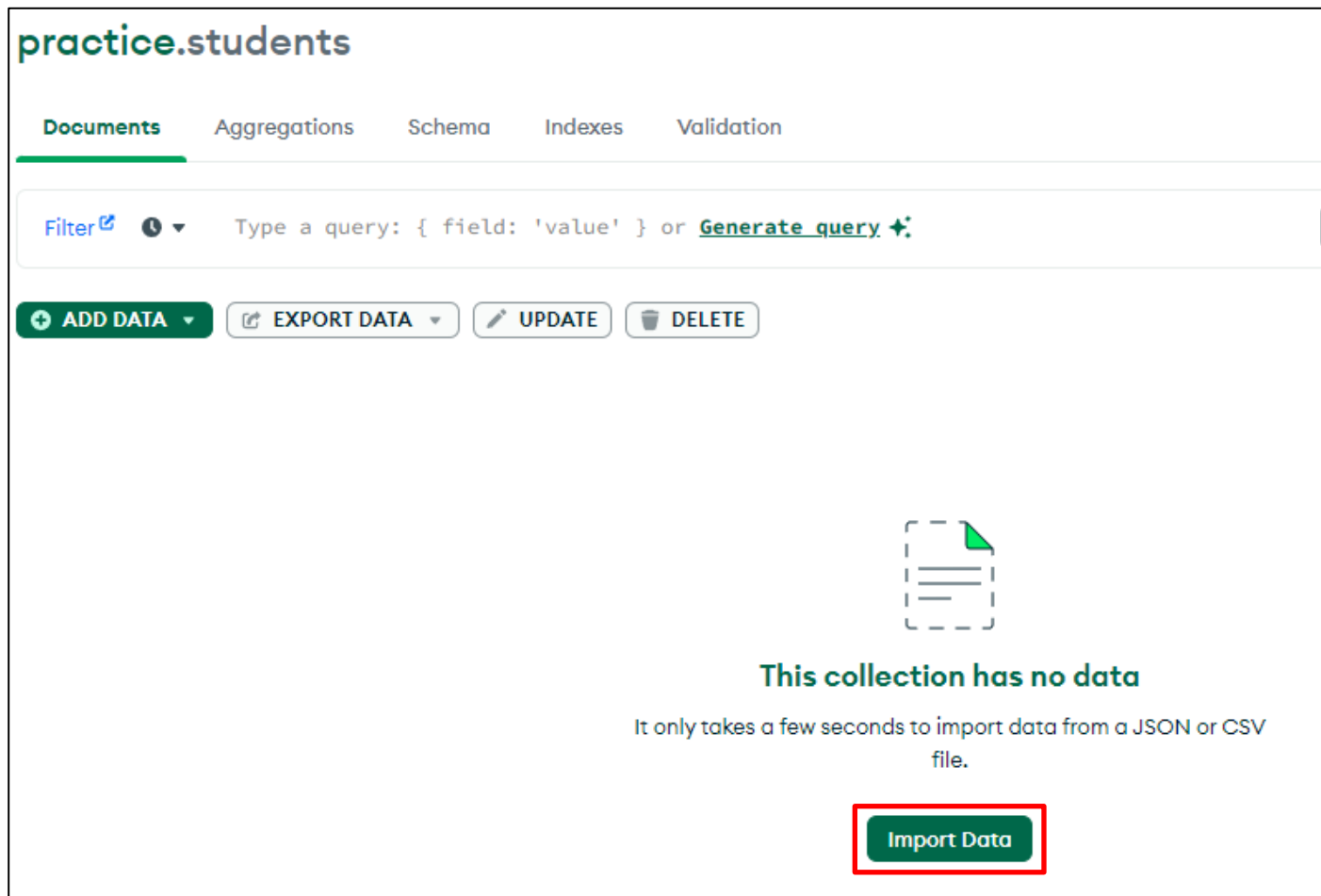
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Show list of collections in the selected database

```
PS F:\PycharmProjects\bigdata_practice> python practice1.py
MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False)
{'name': 'admin', 'sizeOnDisk': 40960, 'empty': False}
{'name': 'config', 'sizeOnDisk': 110592, 'empty': False}
{'name': 'local', 'sizeOnDisk': 73728, 'empty': False}
{'name': 'practice', 'sizeOnDisk': 8192, 'empty': False}
Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False), 'practice')
students
```

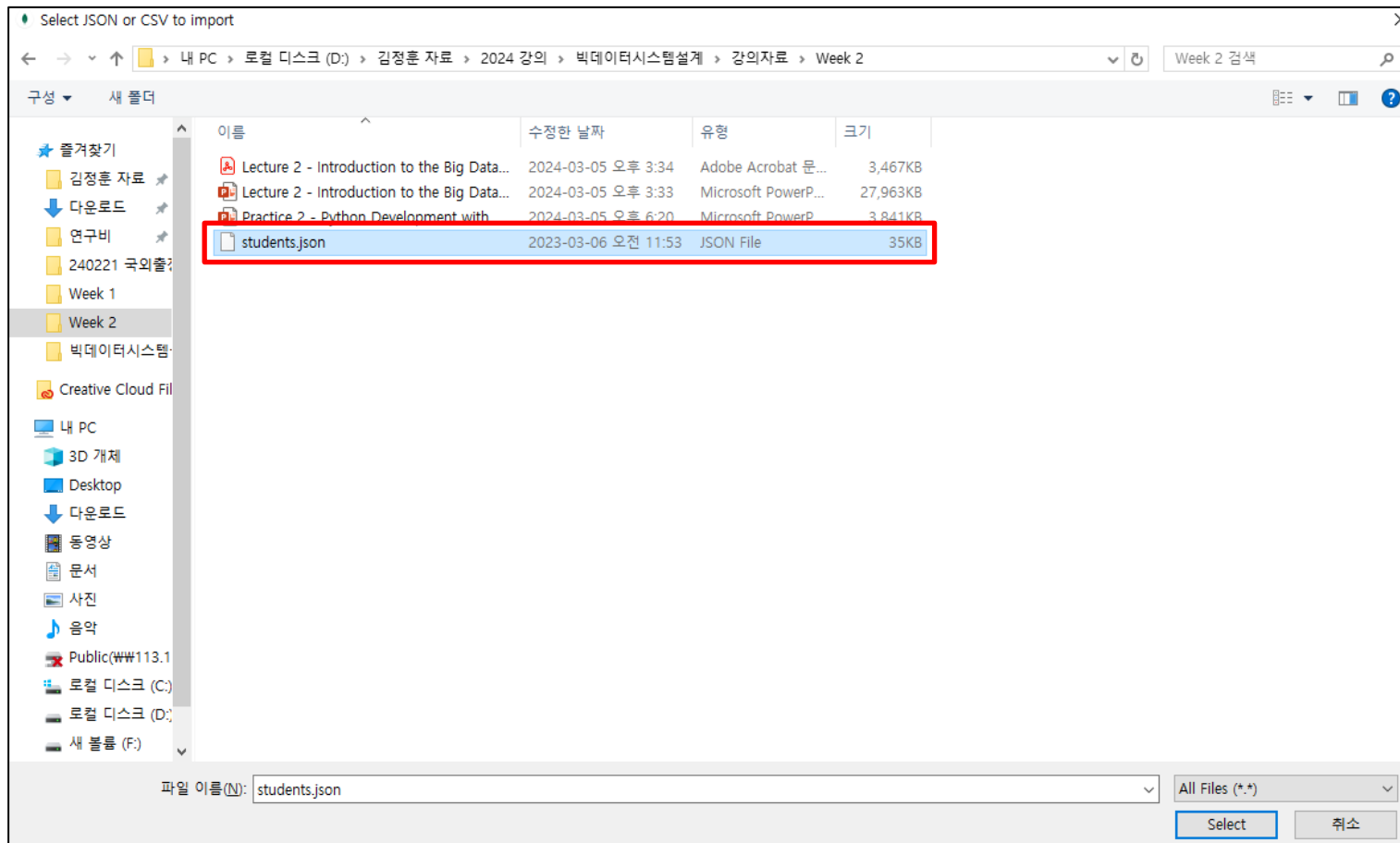
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Import "student.json" dataset into the students collection



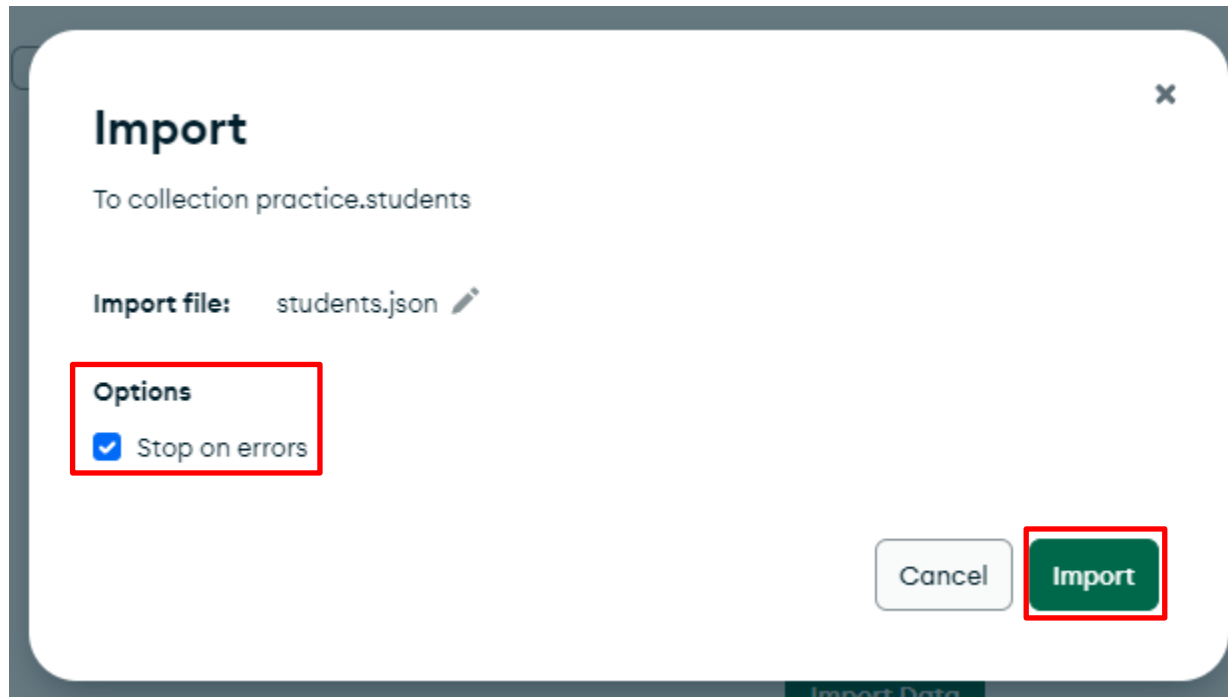
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Import "student.json" dataset into the students collection



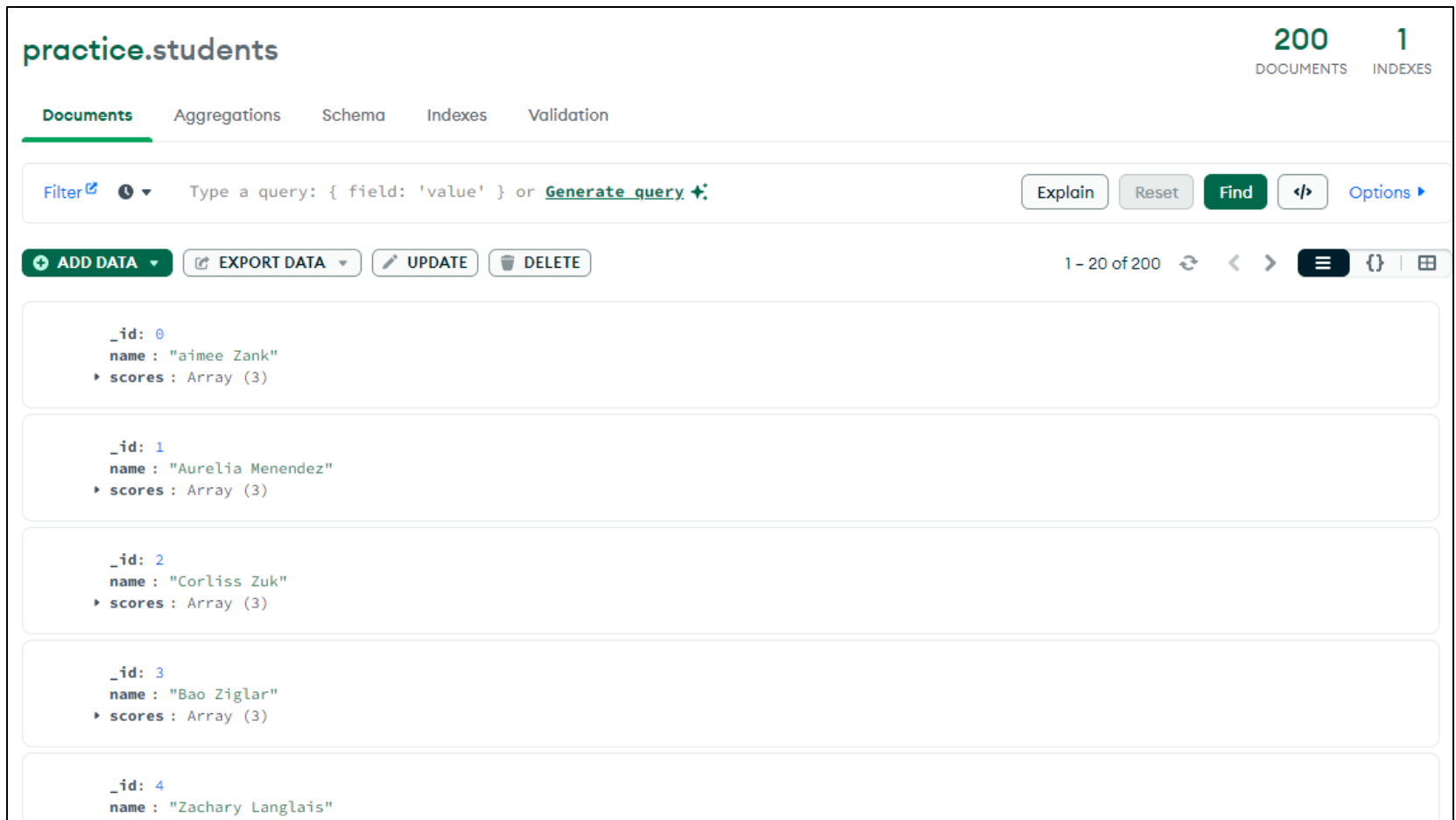
Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Import "student.json" dataset into the students collection



Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Import "student.json" dataset into the students collection



The screenshot shows the MongoDB Compass interface for the 'practice.students' collection. The top right corner indicates 200 documents and 1 index. The 'Documents' tab is selected, showing a list of student records. Each record contains an '_id', a 'name', and a 'scores' array.

_id	name	scores
0	"aimee Zank"	Array (3)
1	"Aurelia Menendez"	Array (3)
2	"Corliss Zuk"	Array (3)
3	"Bao Ziglar"	Array (3)
4	"Zachary Langlais"	Array (3)

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Access a specific collection in the selected database

```
practice1.py x
1  import pymongo
2
3  def main():
4      client = pymongo.MongoClient()
5      print(client)
6
7      for db in client.list_databases():
8          print(db)
9
10     db_conn = client.get_database("practice")
11     print(db_conn)
12
13     for col in db_conn.list_collection_names():
14         print(col)
15
16     collection = db_conn.get_collection("students")
17     print(collection)
18
19  main()
```


Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Access a specific collection in the selected database

```
Terminal: Local × + ▾
{'name': 'admin', 'sizeOnDisk': 40960, 'empty': False}
{'name': 'config', 'sizeOnDisk': 110592, 'empty': False}
{'name': 'local', 'sizeOnDisk': 73728, 'empty': False}
{'name': 'practice', 'sizeOnDisk': 49152, 'empty': False}
Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True), 'practice')
students
Collection(Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True), 'practice'), 'students')
```

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Retrieve data from the selected collection

```
16     collection = db_conn.get_collection("students")
17     print(collection)
18
19     results = collection.find()
20     ds = list(results)
21     print("Number of data: {}".format(len(ds)))
22
23     for data in ds:
24         print(data)
25
26     main()
```

Python Development Environment

- ❖ First Python implementation with MongoDB (cont'd)
 - Retrieve data from the selected collection

```
Terminal: Local x + v
students
Collection(Database(MongoClient(host=['localhost:27017'], document_class=dict, tz_aware=False, connect=True), 'practice'), 'students')
Number of data: 200
{'_id': 0, 'name': 'aimee Zank', 'scores': [{'score': 1.463179736705023, 'type': 'exam'}, {'score': 11.78273309957772, 'type': 'quiz'}, {'score': 35.8740349954354, 'type': 'homework'}]}
{'_id': 1, 'name': 'Aurelia Menendez', 'scores': [{'score': 60.06045071030959, 'type': 'exam'}, {'score': 52.79790691903873, 'type': 'quiz'}, {'score': 71.76133439165544, 'type': 'homework'}]}
{'_id': 2, 'name': 'Corliss Zuk', 'scores': [{'score': 67.03077096065002, 'type': 'exam'}, {'score': 6.301851677835235, 'type': 'quiz'}, {'score': 12.121212121212121, 'type': 'homework'}]}
```

Questions?

SEE YOU NEXT TIME!