Main Themes and Important Ideas/Facts:

1. PyMongo as a Python Driver:

- PyMongo serves as the interface between Python applications and MongoDB databases. It allows Python developers to use MongoDB features in their projects seamlessly.
- **Key point**: PyMongo is crucial for bridging Python applications with MongoDB's functionality.

2. Establishing a Connection:

- To interact with MongoDB, a connection to the database is required. This can be established using the `MongoClient`class from the `pymongo` library.

The basic connection string includes the server address, port (default is 27017), and optional authentication credentials:

```
from pymongo import MongoClient
client = MongoClient('mongodb://user_name:pw@localhost:27017')
```

-

3. Accessing Databases and Collections:

Once a connection is established, the database and collections can be accessed. Databases are accessed through the `client` object, and collections through the database object:

```
db = client['ds4300']  # Or client.ds4300
collection = db['myCollection']  # Or db.myCollection
```

-

4. Inserting Documents:

The `insert_one()` method is used to insert a document into a collection. Documents in MongoDB are stored as Python dictionaries:

```
post = { "author": "Mark", "text": "MongoDB is Cool!", "tags": ["mongodb", "python"] }
post_id = collection.insert_one(post).inserted_id
print(post_id)
```

- **Key point**: The `insert_one()` method returns an `InsertOneResult` object, which includes the `inserted_id` of the newly added document.

5. Querying Data (Finding Documents):

Documents can be queried using the `find()` method, which retrieves all documents matching specified criteria:

```
movies_2000 = db.movies.find({"year": 2000})
```

The query results are returned as a cursor, which can be iterated over. The results can be pretty-printed using `bson.json_util.dumps`:

```
from bson.json_util import dumps
print(dumps(movies_2000, indent = 2))
```

-

6. Setting up a Development Environment with JupyterLab:

- The document emphasizes setting up a proper development environment for practicing with PyMongo:
    1. Activate a Python environment (using `conda` or `venv`).

Install PyMongo using:

```
pip install pymongo
```

Install JupyterLab using:

```
pip install jupyterlab
```

Navigate to the folder in the terminal and run:

```
jupyter lab
```