

Learning Python and how to implement it with a database.

Mar 8th:

It is unclear to me how I go about writing code for it to be used in the database. I am also still learning python so this is what my focus is for today.

Task 1: 15 minutes

- https://www.youtube.com/watch?v=-bt_y4Loofg <Completed>

Task 2: 45 minutes

Realized that I still don't fully understand how to code in Python using pycharm. So spent 20 mins learning how to use pycharm fully.

- <https://www.youtube.com/watch?v=56bPIGf4us0> <Completed>
 - Command D is duplicate line

Task 3: 45 minutes

- How is python working with mongo DB
- <https://www.youtube.com/watch?v=UpsZDGutpZc> <Browsed through main sections>

Mar 9th:

Task 4: 1.5 hours

- Learning python by reviewing and implementing code from class (State, Adaptor, Observer, Singleton)
- Understanding how to make a method in python.

Task 5: 3 hours

- Testing and keyboard short cuts
- Multiline comment → `"""` on either side of the code
- Shifting code (multiline or single line):
 - To Right: tab
 - To Left: shift+tab
- Option + command + L → reformat the code in pycharm
- Reviewing unit testing for Java
- Watched the following videos (1 hour):
 - <https://www.youtube.com/watch?v=K4WTqO4-5RM> (5 min)
 - <https://www.youtube.com/watch?v=2sb5JJLpzw8> (12 min)
- UNIT TESTING:
 - Use Pytest in pycharm OR unittest library

Good video → using “unittest” <https://www.youtube.com/watch?v=6tNS--WetLI>

- Need unit testing for maintaining and automating testing of code.

- ALL TESTS must start with "def test_"
- assertRaises → we can test exceptions
- List of python asserts → <https://docs.python.org/3/library/unittest.html#assert-methods>
- Make code DRY → don't repeat yourself
- Doing def setUp(self): is a method to use at the beginning that allows you to declare instances of variables etc. Which is like the "begin" item in java Eclipse.
 - Then we can also add a tearDown(self)
 - These setup and tear down methods will occur before and after every test method in the file.
- Can also use SETUPCLASS OR TEARDOWNCLASS
 - setUp class runs before everything, tearDown class runs after everything.
- Mocking
 - Say you have a method that pulls from a website. If that website is down, your code will fail. But this is out of your control.
 - So we mock the items from the website.
 - At top of testing page → from unittest.mock import patch

Learning a bit about pytest:

<https://www.youtube.com/watch?v=mzIH8lp4ISA>

- Went through this to see the additional functionality of pytest. May be beyond scope/ goals at present.

Comparison of unittest vs pytest:

<https://www.browserstack.com/guide/pytest-vs-unittest#:~:text=Python%20Testing%20Frameworks,-A%20Python%20testing&text=unittest%3A%20A%20testing%20framework%20included,testing%20discovery%2C%20and%20expressive%20syntax>.

- Similar to unittest, just a little fancier with more options. Will stick with unittest framework for now. Will expand if needed.

Task 6:

- More of a focus on the building of the page needed now.