Python Colloquium 2025

Exercise 1: Creating Property Listings with Python Dictionaries (at home?)

Goal: Practice core data structures and build foundational knowledge for the recommender system.

Concepts Covered:

- Variables, types, and basic I/O
- Lists and dictionaries
- Looping and conditional logic

Task:

1. Define 5 sample property listings as Python dictionaries. Each dictionary should include:
    - property_id (unique string or integer)
    - location (string)
    - type (e.g., "cabin", "condo")
    - price_per_night (numeric)
    - features (list of strings)
    - tags (list of strings)
2. Store these dictionaries in a list called property_listings.
3. Write a loop that:
    - Prints each listing in a readable format.
    - Filters and prints properties under a given budget.

Bonus:

- Add an input prompt to allow the user to search by tag (e.g., "beachfront", "family-friendly").

Exercise 2: Object-Oriented Design + File & DB Integration (during tutorial 1)

Goal: Apply OOP principles to the project and introduce persistent storage.

Concepts Covered:

- Classes and methods
- JSON file handling
- SQLite with sqlite3

Task Part A: Classes

1. Define a Property class with attributes:
    - property_id, location, type, price_per_night, features, tags
2. Define a User class with:
    - user_id, name, group_size, preferred_environment, budget
    - Method: matches(self, property_obj) – returns True if property is a good fit

Task Part B: JSON File I/O

1. Create several Property and User objects and convert them to dictionaries.
2. Save these to JSON files using json.dump().
3. Load and recreate the objects from the files using json.load().

Task Part C: SQLite

1. Create a SQLite database rentals.db with two tables: users and properties.
2. Insert the sample data from Part A.
3. Write a query to retrieve all properties under a user's budget.

Exercise 3: Ranking with NumPy and LLM Integration (during tutorial 2)

Goal: Build and test the matching and scoring logic; optionally enhance with LLM-based recommendations.

Concepts Covered:

- NumPy arrays and vectorized operations
- Pandas for tabular data manipulation
- **Calling APIs with requests (tutorial only)**

Task Part A: Vector Matching

1. Load a DataFrame with sample properties and one with sample users.
2. Create a scoring function using NumPy to rank how well a property matches a user's preferences.
    - For example: similarity in environment, group size fit, price match.

Task Part B: Pandas Ranking

1. Sort the top 3 properties for each user based on the scoring function.
2. Display results in a friendly format.

Task Part C: LLM Integration

1. Use requests to query a free LLM via OpenRouter or similar endpoint.
2. Prompt example:

   prompt = f"Generate a fun description for a summer rental in {location} under ${budget} with features: {features}"
3. Parse the response and add it as a new field (llm_summary) in the listing dictionary.