

Exercise 1: Data Filtering and Transformation

Imagine you have a list of names and their corresponding scores for a class. Filter the list to keep only names of students who scored above 80 and then apply a function to format their names with a congratulatory message.

Steps:

1. Create a list of names and scores;
2. Define functions for filtering and formatting;
3. Apply filter and map to achieve the filtering and transformation;
4. Print the results.

Answer: You'll see congratulatory messages for students with scores above 80 (e.g., "Congratulations, Alice! You scored 85!").

Challenge: Combine steps 3 and 4 into a single line using list comprehension.

Exercise 2: Text Processing Pipeline

Imagine you need to process a text string without using any additional libraries. Your task is to implement a text processing pipeline that reads a provided string, converts it to lowercase, removes punctuation, and then splits it into individual words.

Steps:

1. Create a sample string.
2. Define a function to clean and tokenize the text without using external libraries.
3. Apply the function to the text.
4. Print the result.

Challenge: Extend the function to filter out basic stop words like "the", "a", "is", and calculate the frequency of each remaining word.

Exercise 3: Game Logic with Higher-Order Functions

Challenge: Simulate rolling two dice in a simple dice game. Use functions to define die values (1-6) and implement rolling them together to get a total score.

Steps:

1. Define functions for generating random die values;
2. Define a function to roll both dice and get the total score;
3. Use the function to roll and print the result.

Answer: You'll see a randomly generated total score between 2 and 12.

Challenge: Extend the game logic to check for winning conditions and handle multiple rolls.

Reading exercise

Read about list comprehension, list transformation and lazy-like evaluation.