

List Methods Practice

Assignment Overview:

In this assignment, you will practice using different list methods in Python. Lists are a fundamental data structure, and understanding how to manipulate them is essential for Python programming.

Instructions:

Complete the following tasks using Python list methods. Write your code in a single Python file.

Tasks:

1. Create and Modify Lists

- Create a list called `fruits` with the items: "apple", "banana", "orange"
- Add "grape" to the end of the list using an appropriate method
- Insert "mango" at position 2 in the list
- Remove "banana" from the list
- Print the final list

2. List Operations

- Create a list `numbers` with values: 10, 20, 30, 40, 50
- Create a second list `more_numbers` with values: 60, 70, 80
- Combine both lists into a new list called `all_numbers`
- Make a copy of `all_numbers` called `numbers_copy`
- Reverse the order of `numbers_copy`
- Print both `all_numbers` and `numbers_copy`

3. Sorting and Counting

- Create a list `scores` with these values: 85, 92, 78, 65, 92, 85, 74
- Sort the list in ascending order
- Sort the list in descending order

- Count how many times 92 appears in the list
- Find and print the highest and lowest scores
- Print the sorted list

4. List Manipulation

- Create a list letters with values: "a", "b", "c", "d", "e"
- Use slicing to create a new list with only the first three letters
- Use slicing to create another list with only the last two letters
- Find and print the index of "c" in the list
- Replace "d" with "z" in the original list
- Print the modified list

Dictionary Methods Practice

Assignment Overview:

In this assignment, you will practice using different dictionary methods in Python.

Dictionaries are essential data structures that store key-value pairs and are widely used in Python programming.

Instructions:

Complete the following tasks using Python dictionary methods. Write your code in a single Python file.

Tasks:

1. Create and Access Dictionaries

- Create a dictionary called student with these key-value pairs:
 - "name": "John"
 - "age": 20
 - "grade": "A"
 - "courses": ["Math", "Science", "History"]
- Print the student's name

- Print the list of courses
- Add a new key-value pair: "email": "john@example.com"
- Print the entire dictionary

2. Modify Dictionary Values

- Change the student's age to 21
- Add a new course "Computer Science" to the courses list
- Update the grade to "A+"
- Print the modified dictionary

3. Dictionary Operations

- Create a second dictionary `student_address` with these key-value pairs:
 - "street": "123 College Ave"
 - "city": "New York"
 - "zip": "10001"
- Combine both dictionaries into a new dictionary called `student_info`
- Remove the "zip" key from `student_info`
- Check if "phone" exists in the dictionary
- Print the final dictionary

4. Dictionary Methods

- Create a dictionary `word_count` with these key-value pairs:
 - "hello": 5
 - "world": 10
 - "python": 15
- Get all keys and print them as a list
- Get all values and print them as a list
- Get all key-value pairs and print them
- Make a copy of the dictionary
- Clear the original dictionary
- Print both dictionaries to verify one is empty and one is a copy

Shopping Discount Calculator

Assignment Overview

In this assignment, you will create a simple shopping discount calculator that determines discounts based on purchase amount, customer loyalty status, and seasonal promotions. This program will use if statements to make decisions similar to those in real-world retail systems.

Instructions

Create a Python program that calculates the final price after applicable discounts. Do not use any functions - just use variables and if statements.

Tasks

1. Set up the initial variables:

- `purchase_amount`: The total cost of items in the shopping cart
- `is_loyalty_member`: Boolean (True/False) indicating if the customer is a loyalty member
- `day_of_week`: A string with the current day (e.g., "Monday", "Tuesday", etc.)
- `is_holiday_season`: Boolean indicating if it's currently a holiday shopping season

2. Apply discount rules using if statements:

- If purchase amount is greater than \$100, apply a 10% discount
- If purchase amount is greater than \$50 (but less than or equal to \$100), apply a 5% discount
- If the customer is a loyalty member, apply an additional 5% discount
- If it's a weekend (Saturday or Sunday), apply an additional 2% discount
- If it's a holiday season, apply an additional 8% discount

3. Calculate and display:

- The original purchase amount

- Each discount that was applied and its amount
- The final price after all discounts

For Loop Assignment - Number Patterns

Assignment Overview:

In this assignment, you will create number patterns using for loops in Python.

Instructions:

Write a Python program that generates the following patterns using for loops:

1. Pattern 1: Counting Numbers

- Print numbers from 1 to 10 on a single line.

Pattern 2: Multiplication Table

- Create a multiplication table for the number 5 (from 1 to 10).

While Loop Assignment - Interactive Game

Assignment Overview:

Create a simple number guessing game using a while loop.

Instructions:

Write a Python program that:

1. Sets a "secret number" (you can choose any number between 1 and 50)
2. Asks the player to guess the number
3. Tells the player if their guess is too high or too low
4. Keeps asking until the player guesses correctly
5. Keeps track of how many guesses it took

Student Grade Management System

Assignment Overview:

In this assignment, you will create a simple student grade management system using Python lists and dictionaries. This will help you understand how these data structures work together to organize and manage information.

Requirements:

Create a Python program that does the following:

1. Create a list to store information about students
2. Each student should be represented as a dictionary with the following keys:
 - name: The student's full name
 - id: A unique student ID number
 - grades: Another dictionary containing course names as keys and grades as values
3. Your program should:
 - Add at least 3 students to the list
 - Add grades for at least 3 courses for each student
 - Display all students with their grades

- Calculate and display the average grade for each student
- Find and display the student with the highest average grade
- List all students with an average grade above 80
- Calculate and display the average grade for a specific course
- Add a new student to the system