

100 Python Practice Problems

Python Fundamentals

1. Print 'Hello, World!'
2. Assign variables of type int, float, string, bool
3. Swap two variables without using a temporary variable
4. Find the maximum of three numbers
5. Compute factorial of a number using a loop
6. Compute factorial of a number using recursion
7. Check if a number is prime
8. Print all prime numbers up to n
9. Print the Fibonacci sequence up to n
10. Sum all elements in a list
11. Find the average of numbers in a list
12. Reverse a string
13. Check if a string is a palindrome
14. Count vowels in a string
15. Convert Celsius to Fahrenheit
16. Convert a number to binary, octal, and hex
17. Implement a simple calculator (input string like '5 + 3')
18. Use try/except to handle division by zero
19. Read input from the user and validate it
20. Use enumerate() to print list items with indices

Data Structures

21. Create a list of squares for numbers 1–10
22. Filter even numbers from a list using a loop
23. Filter even numbers from a list using list comprehension
24. Create a dictionary mapping numbers to their squares
25. Count frequency of characters in a string using a dictionary
26. Merge two dictionaries
27. Get keys with the maximum value from a dictionary
28. Create a set from a list (remove duplicates)
29. Find common elements in two lists using sets
30. Implement a stack using a list
31. Implement a queue using a list
32. Implement a queue using collections.deque
33. Implement a linked list node class
34. Traverse a linked list
35. Implement a simple hash table using a dictionary
36. Create a 2D list (matrix) 3x3 and print it
37. Transpose a matrix
38. Flatten a nested list
39. Sort a list of tuples by the second element
40. Implement BFS for a graph represented as an adjacency list

Functions & OOP

41. Write a function that returns the square of a number
42. Write a function that returns the max in a list
43. Write a function that accepts variable arguments *args
44. Write a function that accepts keyword arguments **kwargs
45. Create a Vector2D class with addition and subtraction
46. Add a magnitude method to Vector2D
47. Create an Agent class with name and state
48. Add a method choose_action() to Agent
49. Create a LearningAgent subclass with learn() method
50. Implement multiple inheritance with an EnemyAgent class
51. Override methods in a subclass
52. Implement __str__ and __repr__ for a class
53. Use @staticmethod in a class
54. Use @classmethod in a class
55. Implement operator overloading (e.g., + for Vector2D)
56. Track the number of created agents with a class variable
57. Implement encapsulation (private attributes with getters/setters)
58. Create a grid-world simulation with agents moving randomly
59. Implement a method to check for collisions between agents
60. Extend grid-world to track agent rewards

File Handling & Data Manipulation

61. Write a list of numbers to a text file
62. Read numbers from a text file and sum them
63. Read CSV file using csv module
64. Write CSV file using csv module
65. Read JSON file
66. Write JSON file
67. Merge multiple JSON files into one
68. Count how many times each key appears in JSON objects
69. Write a logger that appends messages with timestamps
70. Generate random data for agents and write to CSV
71. Read CSV of agent data and compute averages
72. Filter agents above a threshold skill level
73. Convert CSV to JSON
74. Convert JSON to CSV
75. Handle file errors using try/except

NumPy & Data Science Prep

76. Create a 1D NumPy array
77. Create a 2D NumPy array (matrix)
78. Perform element-wise addition, subtraction, multiplication
79. Compute dot product of two vectors
80. Compute Euclidean distance between points
81. Normalize a 1D array to [0,1]
82. Find max, min, mean, median using NumPy
83. Index and slice NumPy arrays

- 84. Reshape a 1D array to 2D
- 85. Use boolean masks to filter arrays

AI/ML Basics

- 86. Implement linear regression from scratch (using NumPy)
- 87. Implement gradient descent for linear regression
- 88. Create a simple neural network class (one hidden layer)
- 89. Implement sigmoid activation and derivative
- 90. Train network on a small dataset (predict $y=2x+3$)
- 91. Save network weights to a file
- 92. Load network weights from a file
- 93. Implement mini-batch gradient descent
- 94. Compute mean squared error for predictions
- 95. Plot predictions vs real values using matplotlib

Reinforcement Learning / Agent Simulation

- 96. Implement a Q-table for a simple grid-world
- 97. Update Q-values based on rewards
- 98. Implement epsilon-greedy action selection
- 99. Create multiple agents competing for the same goal
- 100. Track and plot cumulative rewards over episodes