

50 Python Exercises: Beginner to Intermediate

This list is designed to build your Python skills from the ground up. Start with the beginner exercises to master the fundamentals, then move to the intermediate ones to tackle more complex problems.

Part 1: Beginner Exercises (1-25)

Focus: Variables, data types, operators, loops, conditionals, functions, and basic data structures.

1. **Hello, World!:** Print the string "Hello, World!".
2. **User Input and Greeting:** Ask the user for their name and print a personalized greeting.
3. **Simple Calculator:** Create a program that takes two numbers as input and performs addition, subtraction, multiplication, and division.
4. **Even or Odd:** Check if a given number is even or odd.
5. **String Reversal:** Write a function that reverses a given string.
6. **Area of a Circle:** Calculate the area of a circle given its radius. $\text{Area} = \pi r^2$.
7. **Temperature Converter:** Convert a temperature from Celsius to Fahrenheit and vice versa.
8. **Largest of Three Numbers:** Find the largest number among three user-provided numbers.
9. **Vowel Counter:** Count the number of vowels in a given string.
10. **Factorial Calculator:** Calculate the factorial of a number using a loop.
11. **Fibonacci Sequence:** Generate the first 'n' numbers of the Fibonacci sequence.
12. **Palindrome Checker:** Check if a string is a palindrome (reads the same forwards and backward).
13. **List Sum:** Calculate the sum of all elements in a list.
14. **Find Maximum in List:** Find the largest element in a list without using the built-in `max()` function.
15. **Count List Elements:** Count the occurrences of a specific element in a list.
16. **Unique Elements:** Create a new list containing only the unique elements from an original list.
17. **Dictionary Basics:** Create a dictionary representing a person (name, age, city) and print its values.
18. **Word Count:** Count the number of words in a sentence.
19. **Simple Interest Calculator:** Calculate simple interest given principal, rate, and time.
20. **Guess the Number Game:** Generate a random number and have the user guess

it, providing "higher" or "lower" hints.

21. **Multiplication Table:** Print the multiplication table for a given number.
22. **Check for Prime Number:** Write a function to check if a number is prime.
23. **String Slicing:** Given a string, print the first 3 characters and the last 3 characters.
24. **List Comprehension:** Use list comprehension to create a list of squares of numbers from 1 to 10.
25. **FizzBuzz:** Print numbers from 1 to 100. For multiples of 3, print "Fizz". For multiples of 5, print "Buzz". For multiples of both, print "FizzBuzz".

Part 2: Intermediate Exercises (26-50)

Focus: File I/O, Object-Oriented Programming (OOP), APIs, error handling, and more advanced data manipulation.

26. **Read and Write to File:** Write a program to read content from a text file and write it to a new file.
27. **Count Lines in File:** Count the number of lines, words, and characters in a text file.
28. **CSV File Reader:** Read data from a CSV file and print it in a formatted way.
29. **JSON Data Handling:** Read data from a JSON file, parse it, and access specific values.
30. **Basic Class:** Create a Car class with attributes like make, model, and year, and methods like start_engine().
31. **Inheritance:** Create a ElectricCar class that inherits from the Car class and adds a battery_size attribute.
32. **Exception Handling:** Modify the Simple Calculator (exercise 3) to handle ZeroDivisionError and ValueError.
33. **Contact Book:** Create a simple command-line contact book application using a dictionary to store contacts. Save and load contacts from a file.
34. **To-Do List Application:** A command-line to-do list where users can add, view, and delete tasks. Persist tasks to a file.
35. **Caesar Cipher:** Implement the Caesar cipher for basic string encryption and decryption.
36. **Password Generator:** Create a program that generates a random password of a specified length, including a mix of letters, numbers, and symbols.
37. **Weather API Client:** Use the requests library to fetch weather data for a given city from a free weather API (like OpenWeatherMap).
38. **Web Scraper:** Use requests and BeautifulSoup to scrape and print all the headlines from a news website's homepage.

39. **Lambda Functions:** Rewrite a simple function (e.g., one that doubles a number) using a lambda function.
40. **Map, Filter, Reduce:** Use map to square all numbers in a list, filter to get all even numbers, and functools.reduce to find the product of all numbers.
41. **Recursive Factorial:** Re-implement the factorial calculator (exercise 10) using recursion.
42. **Dice Rolling Simulator:** Simulate rolling two dice and display the results.
43. **Hangman Game:** Create a command-line Hangman game.
44. **Tic-Tac-Toe Game:** Implement a two-player Tic-Tac-Toe game on the command line.
45. **Directory Tree Lister:** Write a script that lists all files and subdirectories within a given directory.
46. **Image Downloader:** Write a script that downloads an image from a URL and saves it locally.
47. **Simple Stopwatch:** Create a command-line stopwatch to measure elapsed time.
48. **Text-Based Adventure Game:** Create a simple adventure game where the user makes choices to navigate through a story.
49. **URL Shortener:** Connect to a URL shortener API (like tinyurl) to create a short URL for a given long URL.
50. **Regular Expression Validator:** Write a function that uses regex to validate if a string is a valid email address.

Part 3: Project Ideas

Build on the skills from the exercises to create a more substantial portfolio piece. Some are tailored to AI and Cybersecurity.

General Projects

- **Personal Blog with a Simple CMS:** A web application (using a framework like Flask or Django) where you can write, edit, and delete blog posts.
- **E-commerce Website Backend:** The backend logic for an online store, including product management, user accounts, and a shopping cart.
- **Data Visualization Dashboard:** Fetch data from an API or a file and use a library like Matplotlib or Seaborn to create various plots and charts.

AI-Focused Projects

- **Sentiment Analysis Tool:** A script that takes a block of text (e.g., a product review) and classifies its sentiment as positive, negative, or neutral. You can start with a library like NLTK or TextBlob.
- **Simple Chatbot:** Create a rule-based chatbot that can answer predefined

questions. For a more advanced version, integrate with a platform like Dialogflow.

- **Image Classifier with a Pre-trained Model:** Use a library like TensorFlow or PyTorch to load a pre-trained model (e.g., MobileNet) and build an application that can classify images you provide.

Cybersecurity-Focused Projects

- **Network Port Scanner:** A tool that scans a target IP address to identify open ports. This is a fundamental tool in network reconnaissance.
- **File Encryption/Decryption Tool:** An application that can encrypt a file using a symmetric encryption algorithm (like AES) and a user-provided password, and then decrypt it back.
- **Password Strength Checker:** A tool that analyzes a password to determine its strength based on criteria like length, character types (uppercase, lowercase, numbers, symbols), and common patterns.
- **Simple Keylogger:** (For educational purposes only and to be used on your own machine). A script that records keystrokes and saves them to a log file. This project helps in understanding malware behavior and system monitoring.