**✅ 100 Days Python Coding Challenge (Beginner to Moderate)**

**📅 Day 1 - Day 10: Basics (Syntax, Variables, Conditionals, Loops)**

1. **Print "Hello World" and your name.**
2. **Swap two numbers without using a third variable.**
3. **Check if a number is even or odd.**
4. **Find the largest of three numbers.**
5. **Check if a number is prime.**
6. **Print all prime numbers in a given range.**
7. **Generate Fibonacci series up to n terms.**
8. **Find factorial of a number using loop.**
9. **Sum of digits of a number.**
10. **Reverse a number and a string.**

**📅 Day 11 - Day 20: Loops, Strings, Lists**

1. **Count vowels, consonants, digits, and special characters in a string.**
2. **Check if a string is a palindrome.**
3. **Find the length of a string without using len().**
4. **Find the second largest number in a list.**
5. **Sort a list in ascending and descending order (without built-in methods).**
6. **Sum of all elements in a list.**
7. **Remove duplicates from a list.**
8. **Find common elements between two lists.**
9. **Find maximum and minimum in a list without built-in functions.**
10. **List comprehension to create a list of squares of numbers from 1 to 10.**

**📅 Day 21 - Day 30: Functions and Recursion**

1. **Write a function to check for prime numbers.**
2. **Write a function to calculate factorial using recursion.**
3. **Write a function to find GCD of two numbers.**
4. **Write a function to reverse a string.**
5. **Write a function to count words in a string.**
6. **Function to find sum and average of list elements.**
7. **Recursive function to calculate Fibonacci series.**
8. **Function to find the largest number from a list.**
9. **Function to check if a string is a palindrome.**
10. **Function to calculate the power of a number using recursion.**

**📅 Day 31 - Day 40: Dictionaries, Tuples, Sets**

1. **Find frequency of each character in a string using a dictionary.**
2. **Find frequency of each word in a string.**
3. **Write a Python program to combine two dictionaries.**
4. **Sort dictionary by values and keys.**
5. **Write a program to convert a list of tuples into a dictionary.**
6. **Write a program to merge two sets.**
7. **Find common elements in two sets.**
8. **Find the difference between two sets.**
9. **Tuple unpacking and swapping values using tuples.**
10. **Write a program to count occurrences of an element in a tuple.**

**📅 Day 41 - Day 50: File Handling and Error Handling**

1. **Read a file and print its contents.**
2. **Write a list of numbers to a file.**
3. **Read numbers from a file and calculate their sum.**
4. **Count number of lines and words in a file.**
5. **Append data to an existing file.**
6. **Handle file not found error.**
7. **Handle division by zero error.**
8. **Create a custom exception for age validation.**
9. **Try-except block to handle multiple exceptions.**
10. **Write a program to copy content from one file to another.**

**📅 Day 51 - Day 60: Object-Oriented Programming (OOP)**

1. **Create a class Student with attributes and methods to display details.**
2. **Class BankAccount with deposit and withdraw methods.**
3. **Class Rectangle to calculate area and perimeter.**
4. **Inheritance example: Person and Employee classes.**
5. **Method overriding in derived class.**
6. **Static method to convert units (like km to miles).**
7. **Class Circle with method to calculate area and circumference.**
8. **Create Car class using \_\_init\_\_ and \_\_str\_\_.**
9. **Demonstrate encapsulation with private attributes.**
10. **Write a class to represent a book (title, author, price).**

**📅 Day 61 - Day 70: Intermediate Practice (Mix Topics)**

1. **Lambda function to find square of a number.**
2. **Lambda function to find even numbers from a list.**
3. **Use map() to double all numbers in a list.**
4. **Use filter() to get odd numbers from a list.**
5. **Use reduce() to multiply all numbers in a list (import functools).**
6. **Sort a list of dictionaries by a value.**
7. **Check if all elements in a list are unique.**
8. **Find common characters between two strings.**
9. **Check if two strings are anagrams.**
10. **Simple calculator using functions (add, subtract, multiply, divide).**

**📅 Day 71 - Day 80: Modules, Packages, Date & Time**

1. **Create and import your own Python module.**
2. **Use built-in math module (factorial, sqrt, gcd).**
3. **Use random module to generate random numbers.**
4. **Use datetime to print current date and time.**
5. **Format date and time using strftime.**
6. **Calculate difference between two dates.**
7. **Use calendar to display a month and year.**
8. **Build a simple countdown timer.**
9. **Simple stopwatch using time module.**
10. **Create a package with two modules and import them.**

**📅 Day 81 - Day 100: Focus on Streamlit + Final Recap**

**📊 Day 81 - Day 90: Introduction to Streamlit & Basic Apps**

1. **Day 81: Setup Streamlit**

* Install Streamlit (pip install streamlit).
* Run first Streamlit app (st.title("Hello, Streamlit!")).

1. **Day 82: Building First Interactive App**

* Create an input form that takes a name and age and displays them using st.text\_input and st.number\_input.

1. **Day 83: Create BMI Calculator**

* Take user weight & height as input.
* Calculate BMI and display the result.

1. **Day 84: Simple Calculator App using Streamlit**

* Add two number inputs.
* Let users select operation (+, -, \*, /) via st.selectbox.
* Display result on button click.

1. **Day 85: To-Do List App**

* Take user task input.
* Add to a task list and display updated list.

1. **Day 86: Weather App UI (Dummy Data)**

* Create a UI for Weather App with city input.
* Display static/dummy temperature, humidity, and weather condition.

1. **Day 87: Student Marks Input & Grade Calculator (with visualization)**

* Input marks for 3-4 subjects.
* Calculate percentage and grade.
* Visualize subject-wise marks using st.bar\_chart.

1. **Day 88: Expense Tracker (Simple Version)**

* Add expense title and amount.
* Display table of expenses and total sum.

1. **Day 89: Data Display App (with CSV upload)**

* Upload CSV file.
* Display data in a table using st.dataframe.

1. **Day 90: Survey Form (Interactive Form)**

* Take user feedback on 4-5 questions (using radio, checkbox, text input).
* Display summary of inputs.

**📊 Day 91 - Day 95: Intermediate Streamlit Projects**

1. **Day 91: Movie Recommendation App (Basic Static Version)**

* Input favorite movie.
* Show 3-5 static recommendations (for now).

1. **Day 92: Dictionary App**

* Input word.
* Return meaning (static or using a dictionary package).

1. **Day 93: Random Quote Generator App**

* Show a random motivational quote on button click.

1. **Day 94: Countdown Timer App (Interactive)**

* Input minutes/seconds.
* Show live countdown timer using time.sleep and st.empty.

1. **Day 95: Flashcards App (for learning)**

* Input Q&A pairs.
* Show random flashcards on button click.

**📊 Day 96 - Day 100: Final Review and Complex Apps**

1. **Day 96: Combine File Handling with Streamlit (Note Saving App)**

* Input a note.
* Save to a text file.
* Show saved notes in Streamlit app.

1. **Day 97: Mini Contact Book with Streamlit (OOP + File)**

* Input name and number.
* Save to file (or in-memory).
* Show contact list.

1. **Day 98: Finalize Your Expense Tracker App (Add file save/load features)**

* Extend Day 88 project.
* Save expenses to file and load on app start.

1. **Day 99: Portfolio / Personal Profile App**

* Build a personal portfolio app with about, contact info, skills, and project links.

1. **Day 100: Final Day Recap: Build and Share a Complete Mini-Project on GitHub**

* Choose one idea from earlier days (Expense Tracker, BMI Calculator, Flashcards).
* Polish the UI and functionality.
* Upload code to GitHub.
* Share a summary post of your 100 Days journey on LinkedIn or GitHub README.

**✅ Optional: Bonus Ideas (If you want to extend further)**

* **Blog App (Static Pages with Streamlit)**.
* **Markdown-based Documentation Viewer.**
* **Daily Planner / Habit Tracker App.**