



## The Prime Step

Holding Your Hand Until You Learn

---

# Becoming The Python Expert

Embrace challenges as opportunities, and knowledge as your greatest asset. Learning is the key to unlocking your full potential.

---

## 1. Decision Making

1. Write a Python program using an if-else statement to check if a number is positive, negative, or zero.
2. Create a program to determine if a given number is even or odd using if-else.
3. Write a Python program that takes a user's age as input and determines if they are eligible to vote (age  $\geq 18$ ).
4. Implement a program to find the maximum of three numbers using nested if statements.
5. Create a program that checks if a year is a leap year (divisible by 4 but not by 100 or divisible by 400) using logical operators.
6. Write a Python program to determine if a given character is a vowel or a consonant using if-else.



7. Create a program that calculates the grade of a student based on their score (A, B, C, D, or F).
8. Write a Python program to determine if a number is within a given range (between 1 and 100).
9. Implement a program that checks if a string is a palindrome (reads the same forwards and backwards).
10. Create a program to calculate the factorial of a number using an if-else loop.
11. Write a Python program that determines if a given year is a century year (ends with '00').
12. Implement a program to check whether a number is even and a multiple of 3.
13. Create a program that compares two numbers and prints the larger number using an if-else statement.
14. Write a Python program that checks if a person is eligible for a senior citizen discount (age  $\geq 60$  or age  $\geq 55$  and a member).
15. Implement a program that determines if a triangle is equilateral, isosceles, or scalene based on the side lengths.
16. Create a program to check if a given string contains both uppercase and lowercase letters.
17. Write a Python program to determine if a person is a teenager (age between 13 and 19) and not a child or an adult.
18. Implement a program that determines if a year is a leap year and also checks if it is a multiple of 5.
19. Create a program that checks if a given number is positive, even, and a multiple of 7.



20. Write a Python program using logical operators to determine if a given character is an alphabet, a digit, or a special character.
21. Write a Python program that takes three numbers as input and determines if they can form a valid triangle based on the triangle inequality theorem.
22. Create a program to check if a given year is a leap year and also determine the next leap year.
23. Write a Python program to determine if a given number is a prime number.
24. Implement a program to check if a given string is a palindrome without considering spaces, punctuation, or letter casing.
25. Create a program that calculates the roots of a quadratic equation based on the values of  $a$ ,  $b$ , and  $c$ , handling cases where the equation has real or complex roots.

## 2. Loops

### For loops:

1. Write a Python program to print the numbers from 1 to 10 using a for loop.
2. Create a program that calculates the sum of all even numbers between 1 and 100 using a for loop.
3. Write a Python program to print a multiplication table for a given number (e.g., 5).
4. Implement a program that counts the number of vowels in a given string using a for loop.
5. Create a program that prints the Fibonacci sequence up to the 10th term using a for loop.



6. Write a Python program to find the factorial of a number using a for loop.
7. Implement a program that prints a pattern of stars in a right-angled triangle using nested for loops.
8. Create a program that generates a list of squares for the numbers from 1 to 10 using a for loop.
9. Write a Python program to find the largest element in a list using a for loop.
10. Implement a program to check if a given number is prime using a for loop.

### **While Loops:**

1. Create a Python program to find the sum of natural numbers from 1 to N using a while loop.
2. Write a program to reverse a given number using a while loop.
3. Implement a program to find the GCD (Greatest Common Divisor) of two numbers using a while loop.
4. Create a program to check if a string is a palindrome using a while loop.
5. Write a Python program to find the first N terms of the geometric progression (GP) using a while loop.



## **Nested Loops:**

1. Implement a program to print a multiplication table for numbers 1 to 5 using nested for-loop
2. Create a program to print a pattern of stars in a diamond shape using nested for loop
3. Write a Python program to generate a 2D matrix of numbers from 1 to 9 using nested for-loop
4. Implement a program to find the common elements between two lists using nested for-loop
5. Create a program that generates a Pascal's Triangle for a given number of rows using nested for loops.

## **3. Data Structures:**

### **Lists:**

1. Create a program that prints all the even numbers from a given list using a for loop and conditional statements.
2. Write a Python program that finds the sum of all elements in a list using a while loop.
3. Implement a program that removes duplicates from a list and stores the result in a new list.
4. Create a program to find the maximum and minimum values in a list using the max() and min() functions.
5. Write a program that counts the number of occurrences of a specific element in a list using a for loop.



## **Tuples:**

1. Create a program to find the product of all elements in a tuple using a for loop.
2. Write a Python program that checks if a given element exists in a tuple using conditional statements.
3. Implement a program that finds the index of the first occurrence of a specific element in a tuple.
4. Create a program that combines two tuples into a single tuple.
5. Write a program to count the number of elements in a tuple.

## **Dictionaries:**

1. Implement a program that iterates through a dictionary and prints all keys and values.
2. Create a program to check if a key exists in a dictionary using conditional statements.
3. Write a Python program that calculates the sum of all values in a dictionary.
4. Implement a program to find the key with the maximum value in a dictionary.
5. Create a program that merges two dictionaries into a new dictionary.

## **Sets:**

1. Write a Python program that finds the intersection of two sets and prints the result.
2. Implement a program to check if a specific element exists in a set using conditional statements.



3. Create a program that finds the union of two sets and stores the result in a new set.
4. Write a program that removes duplicate elements from a set and returns the result.
5. Implement a program that checks if one set is a subset of another set using conditional statements.

## 4. Function & Recursion :

1. Create a Python function that adds two numbers and returns the result.
2. Write a function to find the square of a number.
3. Implement a function that checks if a number is even.
4. Create a function that multiplies two numbers.
5. Write a function to calculate the area of a rectangle.

### Recursion:

1. Implement a recursive function to calculate the sum of natural numbers from 1 to N.
2. Create a recursive function to find the nth Fibonacci number.
3. Write a recursive function to reverse a string.
4. Implement a recursive function to calculate the factorial of a number.
5. Create a recursive function to compute the power of a number ( $a^b$ ).
6. Function and Recursion Combination:
7. Write a recursive function to find the factorial of a number.
8. Implement a function that calculates the sum of even numbers in a list using recursion.
9. Create a recursive function to calculate the product of elements in a list.
10. Write a Python function to check if a word is a palindrome using recursion.
11. Implement a recursive function to calculate the greatest common divisor (GCD) of two numbers.



**Advanced:**

1. Create a recursive function to find the binary representation of a decimal number.
2. Write a function that calculates the  $n$ th term of the geometric progression (GP) using recursion.
3. Implement a recursive function to generate the  $n$ th term of an arithmetic progression (AP).
4. Create a recursive function to solve the Towers of Hanoi problem for three pegs.
5. Write a Python function to calculate the number of ways to climb a staircase with  $N$  steps, considering you can take 1 or 2 steps at a time.