

# Functions

1. **Basic Function Definition:**  
Define a function greet that takes a person's name as a parameter and prints a greeting message like "Hello, [name]!".
2. **Function with Return Value:**  
Write a function square that takes a number as input and returns its square. Test the function by squaring the numbers 2, 3, and 4.
3. **Function with Multiple Parameters:**  
Create a function multiply that accepts two numbers and returns their product. Call the function with different pairs of numbers.
4. **Default Arguments:**  
Define a function add that adds two numbers but has the second parameter default to 0 if not provided. Test the function by calling it with one and two arguments.
5. **Keyword Arguments:**  
Write a function describe\_pet that takes two parameters: a pet\_name and an optional animal\_type (defaulting to "dog"). The function should print a sentence describing the pet.
6. **Variable-Length Arguments (Arbitrary Arguments):**  
Create a function average that accepts any number of numerical arguments and returns their average. Test the function with different sets of numbers.
7. **Lambda Functions:**  
Define a lambda function that multiplies a number by 10. Use this function to multiply the numbers 5, 7, and 9.
8. **Recursive Functions:**  
Write a recursive function factorial that calculates the factorial of a given number. Test the function by finding the factorial of 5.
9. **Higher-Order Functions:**  
Implement a function apply\_to\_list that takes a list of numbers and a function as arguments. It should return a new list with the function applied to each element. Use it with a lambda function to double the numbers in a list.
10. **Functions with Lists:**  
Define a function find\_max that takes a list of numbers and returns the largest number in the list. Test the function with different lists of numbers.
11. **Functions with Dictionaries:**  
Write a function build\_profile that accepts an arbitrary number of keyword arguments and builds a dictionary representing a user profile (e.g., name, age, location).
12. **Nested Functions:**  
Create a function outer\_function that defines and returns an inner\_function. The inner function should simply print a message. Call the outer function to demonstrate how nested functions work.
13. **Global vs Local Variables:**  
Define a function that modifies a global variable. Demonstrate the use of the global keyword to change the value of the global variable from inside the function.

## **Modules**

1. Create a user-defined module "distance" with a function to calculate the distance between two points using the Euclidean distance formula:  $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ . Then, within the main program, import this module and calculate the distance between two points (3, 4) and (6, 8).
2. Develop a user-defined module "trigonometry" with functions to calculate the sine, cosine, and tangent of an angle in degrees. Import this module into the main program and calculate the sine, cosine, and tangent of a given angle.
3. Create a user-defined module "statistics" with functions to calculate the mean, median, and mode of a given list of numbers. Import this module and calculate the mean, median, and mode of the list [12, 15, 10, 17, 15, 14, 12, 18, 13].
4. Develop a user-defined module "currency" with functions to convert currency from USD to EUR, GBP, and JPY. Import this module into the main program and convert \$100 to EUR, GBP, and JPY using the predefined exchange rates.
5. Create a user-defined module "matrix\_operations" with functions to perform matrix addition, subtraction, and multiplication. Import this module and demonstrate the operations on two given matrices.
6. Utilize the inbuilt datetime module to create a program that calculates the number of days between two specific dates provided by the user.
7. Develop a program that uses the inbuilt random module to generate a sequence of 10 random numbers between 1 and 100, displaying the result to the user.
8. Create a program that utilizes the inbuilt os and csv modules to read a CSV file containing student scores, calculate and print the average score, and display the highest scoring student.
9. Develop a program that uses the inbuilt re (regular expression) module to validate email addresses provided by the user, checking if they conform to a correct email format.
10. Utilize the inbuilt matplotlib.pyplot module to create a program that generates and displays a line graph representing the temperature variations over a week, using pre-defined temperature data.