Functions in Python

UNDERSTANDING FUNCTIONS WITH EXAMPLES

What is a Function?

- A function is a block of organized, reusable code that performs a single, related action.
- Functions help in organizing the code and improving readability.
- They allow code to be called multiple times without repetition.

Syntax of a Function

- def function_name(parameters):

  '''docstring''‘

  statement(s)

- function_name: Name of the function.
- parameters: Inputs to the function (optional).
- docstring: Describes the function's purpose (optional).
- statement(s): The block of code that runs when the function is called.

Example of a Simple Function

- def greet():

  '''This function greets the user.'''

  print('Hello, World!')

- Here, 'greet' is the function name.
- It prints a message when called.

Calling a Function

- Once defined, you can call a function using its name:

- greet()

- Output:

Hello, World!

Function with Parameters

- def greet(name):

  '''This function greets a person by name.'''

  print(f'Hello, {name}!')

- 'name' is the parameter passed to the function.
- The function prints a personalized greeting based on the input.

Calling a Function with Parameters

- greet('Alice')

- • Output:

Hello, Alice!

- You can call the function with different names to greet other people.

Return Statement in Functions

- Functions can return a value using the 'return' statement.

- def add(a, b):

  '''This function returns the sum of two numbers.''’

  return a + b

Example of a Function with Return

- sum = add(5, 3)
  print(sum)

- Output:
  8

- The 'add' function returns the sum of 5 and 3.

Conclusion

- Functions are a fundamental part of Python programming.
- They help to make code reusable, modular, and easier to maintain.
- Use functions to break down your code into smaller, manageable parts.