**Functions in Python – Detailed Explanation**

In Python, a **function** is a block of organized, reusable code that is used to perform a single, related action. Functions help make programs shorter, easier to read, and more efficient by allowing the programmer to break a large program into smaller, manageable parts.

Functions play a very important role in Python programming because they promote **code reusability** and **modularity**. Instead of writing the same code multiple times, we can write it once inside a function and call it whenever needed.

**🧩 Definition of a Function**

A function is defined using the **def keyword**, followed by the **function name**, a pair of **parentheses** (which may contain parameters), and a **colon**. The body of the function is written with an **indentation**.

**Syntax:**

def function\_name(parameters):

# statements

return value

* **def** → keyword used to define a function
* **function\_name** → name given to the function (should be meaningful)
* **parameters** → optional; used to pass data to the function
* **return** → optional; used to send a result back to the caller

**💬 Why Functions Are Used**

1. To divide a large program into smaller parts.
2. To avoid repetition of code.
3. To make the program easy to read, debug, and maintain.
4. To reuse the same code in different programs or at different times.
5. To improve the structure and organization of the code.

**⚙️ Types of Functions in Python**

1. **Built-in Functions**
   * These are functions already provided by Python.
   * Examples: print(), len(), sum(), input(), max(), min(), type()
2. **User-defined Functions**
   * These are functions created by the programmer to perform specific tasks.
   * Example:
   * def add(a, b):
   * return a + b
3. **Lambda (Anonymous) Functions**
   * These are short, one-line functions that do not have a name.
   * They are created using the keyword lambda.
   * Example:
   * square = lambda x: x \* x
   * print(square(5))

**🔁 Types of User-defined Functions**

1. **Function with No Arguments and No Return Value**
   * Example:
   * def greet():
   * print("Hello, welcome to Python!")
2. **Function with Arguments but No Return Value**
   * Example:
   * def add(a, b):
   * print("Sum =", a + b)
3. **Function with Arguments and Return Value**
   * Example:
   * def multiply(a, b):
   * return a \* b
4. **Function with Default Arguments**
   * Example:
   * def greet(name="Guest"):
   * print("Hello,", name)

Bottom of Form