**Que 1 : Defining functions in Python.**

A **function** in Python is a reusable block of code that performs a specific task.

Basic Syntax:

def function\_name(parameters):

# code block

return result

**1 : A simple function with no parameters**

def greet():

print("Hello, welcome to Python!”)

#Call the Function

greet()

**2 : Function with parameters**

def add(a, b):

return a + b

#Calling the Function

result = add(5, 3)

print(result) # Output: 8

**3 : Function with default parameters**

def greet(name="User"):

print("Hello", name)

#Output

greet("Krishna") # Output: Hello Krishna

greet() # Output: Hello User

**4 : Function with no return**

def say\_hello():

print(“Hello!")

**5 : Return multiple values**

def calculate(a, b):

return a + b, a \* b

sum\_, product = calculate(4, 5)

print("Sum:", sum\_, "Product:", product)

**Que 2 : Different types of functions: with/without parameters, with/without return values.**

**1. Without Parameters & Without Return Value**

Function takes no input and gives no output.  
It simply performs an action (like printing something).

Example:

def greet():

print("Hello! Welcome to Python.")

greet()

Output : Hello! Welcome to Python.

**2. With Parameters & Without Return Value**

Takes input but doesn't return anything (just performs some action).

Example :

def greet\_user(name):

print("Hello", name, "!")

greet\_user(“Krishna")

Output : Hello Krishna !

**3. Without Parameters & With Return Value**

No input, but gives an output using return.

Example :

def get\_pi():

return 3.14159

pi\_value = get\_pi()

print("Value of PI is:", pi\_value)

Output : Value of PI is: 3.14159

**4. With Parameters & With Return Value**

Takes input and gives output using return.

Example :

def add(a, b):

return a + b

result = add(5, 3)

print("Sum:", result)

Output : Sum: 8

**Que 3 : Anonymous functions (lambda functions).**

In Python, anonymous functions are functions without a name.  
They are created using the keyword lambda, and are also called lambda functions.

Syntax : lambda arguments : expression

Example 1: Add two numbers

add = lambda a, b: a + b

print(add(3, 5)) # Output: 8

When to Use Lambda Functions?

For short and temporary functions  
 When passing a function as an argument  
 Not good for complex operations (use def instead)