1. Defining and Calling Functions

A function is defined using the `def` keyword and is called by its name followed by parentheses `()`.

Example:

```
def say_hello():
    print("Hello, World!")

# Calling the function
say_hello()
```

Output:

```
Hello, World!
```

2. Parameters and Return Values

You can pass **parameters** (inputs) into a function, and use the `return` statement to **return a value**.

Example:

```
def add(a, b):
    return a + b

result = add(5, 3)
print("Result:", result)
```

Output:

```
Result: 8
```

Default Parameters:

```
def greet(name="Guest"):
    print("Hello,", name)

greet()         # Hello, Guest
greet("Raj")    # Hello, Raj
```

3. `*args` and `**kwargs`

These are used when you're not sure how many arguments a function will receive.

- *args: allows **multiple positional arguments**
- **kwargs: allows **multiple keyword arguments**

`*args` – Non-keyword variable-length arguments -Treats all extra values as a **tuple**

```
def add(*numbers):
    total = sum(numbers)
    print("Total:", total)

add(1, 2, 3, 4)
```

Output:

```
Total: 10
```

`**kwargs` – Keyword variable-length arguments-Treats all extra named arguments as a **dictionary**

```
def print_info(**info):
    for key, value in info.items():
        print(f"{key}: {value}")

print_info(name="Raj", age=24, city="Kathmandu")
```

Output:

```
name: Raj
age: 24
city: Kathmandu
```

4. Lambda Functions (Anonymous Functions)

A **lambda function** is a **short, one-line function** without a name, often used for simple operations.

Syntax: lambda arguments: expression

Example 1: Square a number

```
square = lambda x: x * x
print(square(5))  # Output: 25
```

Example 2: Add two numbers

```
add = lambda a, b: a + b
print(add(3, 7))  # Output: 10
```

Example 3: Use inside `sorted()`

```
names = ["Raj", "Anil", "Binita"]
# Sort by string length
sorted_names = sorted(names, key=lambda name: len(name))
print(sorted_names)
```

Output:

```
['Raj', 'Anil', 'Binita']
```

Mini Assignment:

1. Create a function that takes any number of numbers using `*args` and returns their average.
2. Create a lambda function that checks if a number is even or odd.