Python Functions - Detailed Notes

5. FUNCTIONS IN PYTHON

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

Defining & Calling Functions

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

greet()  # Output: Hello, World!
```

Function Arguments

---

1. Positional Arguments:

```
def greet(name, age):
    print(f"Hello {name}, you are {age} years old.")
greet("Alice", 25)
```

2. Keyword Arguments:

```
greet(age=30, name="Bob")
```

3. Default Arguments:

```
def greet(name="Guest"):
    print(f"Hello, {name}")
greet()        # Hello, Guest
greet("Sam")   # Hello, Sam
```

4. *args (Variable Positional Arguments):

```
def total(*args):
    print(sum(args))
total(1, 2, 3, 4)  # 10
```

5. **kwargs (Variable Keyword Arguments):

```
def print_info(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")
print_info(name="Alice", age=25)
```

Return Values

```
def add(a, b):
    return a + b
result = add(3, 4)
print(result)  # 7
```

Returning multiple values:

```
def operations(a, b):
    return a + b, a * b
sum_, product = operations(2, 3)
```

Recursion

```
def factorial(n):
    if n == 0 or n == 1:
        return 1
    return n * factorial(n - 1)
print(factorial(5))  # 120
```

Python Functions - Detailed Notes

Lambda Functions

```
add = lambda a, b: a + b
print(add(3, 4))  # 7

nums = [1, 2, 3, 4]
squared = list(map(lambda x: x**2, nums))
print(squared)  # [1, 4, 9, 16]
```

Summary Table

Feature	Example	Notes
def	def greet():	Defines a named function
*args	def fun(*args):	Accepts any number of positional args
**kwargs	def fun(**kwargs):	Accepts any number of keyword args
return	return a + b	Sends result back to caller
lambda	lambda x: x * 2	One-line anonymous function
Recursion	factorial(n): return ...	Function calls itself