# Module Python -Fundamentals of Python Language

Vrinda Mavadhiya 10th June Python

**Defining and calling functions in Python.**

** Function arguments (positional, keyword, default).**

** Scope of variables in Python.**

** Built-in methods for strings, lists, etc.**

**What is a Function?**

A function is a reusable block of code that performs a specific task. It helps make your code modular, organized, and efficient.

**Defining a Function**

def greet():

print("Hello, welcome to Python!")

* def: keyword to define a function
* greet: function name
* (): can contain parameters
* : and **indentation** define the function body

**✅ Calling a Function**

greet() # Output: Hello, welcome to Python!

**🔹 Function Arguments**

You can pass **data (arguments)** to functions.

**✅ 1. Positional Arguments**

* Values are assigned **based on position**.

def greet\_user(name, age):

print(f"Hello {name}, you are {age} years old")

greet\_user("Alice", 25)

**✅ 2. Keyword Arguments**

* Specify arguments by **name**.

greet\_user(age=25, name="Alice")

**✅ 3. Default Arguments**

* Provide a **default value** if none is passed.

def greet(name="Guest"):

print(f"Hello {name}!")

greet() # Output: Hello Guest!

greet("Alice") # Output: Hello Alice!

**✅ 4. Arbitrary Arguments**

* Use \*args for **multiple positional** values.

def add\_all(\*numbers):

return sum(numbers)

print(add\_all(1, 2, 3)) # Output: 6

* Use \*\*kwargs for **multiple keyword** arguments.

def print\_details(\*\*info):

for key, value in info.items():

print(key, ":", value)

print\_details(name="Alice", age=25)

**🔹 Scope of Variables in Python**

**Scope** refers to the region where a variable is accessible.

**✅ Types of Scope (LEGB Rule):**

| **Type** | **Description** |
| --- | --- |
| Local | Inside the function |
| Enclosing | In the outer function (for nested funcs) |
| Global | Defined at the top-level of the file |
| Built-in | Predefined by Python (e.g., len()) |

**✅ Example:**

x = 10 # Global scope

def my\_function():

x = 5 # Local scope

print("Inside function:", x)

my\_function()

print("Outside function:", x)

**Output:**

Inside function: 5

Outside function: 10

**🔹 Built-in Methods in Python**

**STRING METHOD**

| **Method** | **Description** | **Example** |
| --- | --- | --- |
| **.upper()** | **Converts to uppercase** | **"hello".upper() → HELLO** |
| **.lower()** | **Converts to lowercase** | **"HELLO".lower() → hello** |
| **.strip()** | **Removes spaces** | **" hello ".strip() → "hello"** |
| **.replace()** | **Replace part of string** | **"hi".replace("i", "ello") → hello** |
| **.split()** | **Split string into list** | **"a,b,c".split(",") → ['a','b','c']** |

**✅ List Methods**

| **Method** | **Description** | **Example** |
| --- | --- | --- |
| **.append()** | **Add item at the end** | **mylist.append(4)** |
| **.pop()** | **Remove and return last item** | **mylist.pop()** |
| **.remove(x)** | **Remove first occurrence of x** | **mylist.remove(2)** |
| **.sort()** | **Sort the list** | **mylist.sort()** |
| **.reverse()** | **Reverse the list** | **mylist.reverse()** |