User-defined Functions and Lambda Functions in Python

Student: Nachiket Prajapati

C19 (NCPL)

Instructor :- Mallikarjun sir

# Introduction

This document demonstrates the usage of user-defined functions and lambda functions in Python. It includes examples of both types of functions and shows how to replicate the functionality of user-defined functions using lambda functions.

# 1. User-defined Functions

User-defined functions are functions that are defined by users to perform specific tasks. They are defined using the 'def' keyword in Python.

## Example 1: Addition Function

def add(a, b):  
 return a + b  
  
# Usage  
result = add(3, 5)  
print("Addition Result:", result)

In this example, 'add' is a user-defined function that takes two parameters and returns their sum.

## Example 2: String Concatenation

def concatenate(str1, str2):  
 return str1 + " " + str2  
  
# Usage  
result = concatenate("Hello", "World")  
print("Concatenation Result:", result)

This 'concatenate' function takes two string parameters and combines them with a space in between.

# 2. Lambda Functions

Lambda functions, also known as anonymous functions, are small, single-expression functions that do not require a name. They are defined using the 'lambda' keyword in Python.

## Example 1: Addition Function with Lambda

# Lambda function for addition  
add\_lambda = lambda a, b: a + b  
  
# Usage  
result = add\_lambda(3, 5)  
print("Addition Result (Lambda):", result)

## Example 2: String Concatenation with Lambda

# Lambda function for concatenation  
concatenate\_lambda = lambda str1, str2: str1 + " " + str2  
  
# Usage  
result = concatenate\_lambda("Hello", "World")  
print("Concatenation Result (Lambda):", result)

# 3. Replicating User-defined Functions with Lambda Functions

It’s possible to replicate simple user-defined functions with lambda functions for quick, single-line operations. Here are the same examples replicated using lambda functions to illustrate this.

## Example 1: Replicating 'add' function with 'lambda'

# Lambda function equivalent  
add\_lambda = lambda a, b: a + b

## Example 2: Replicating 'concatenate' function with 'lambda'

# Lambda function equivalent  
concatenate\_lambda = lambda str1, str2: str1 + " " + str2

# Conclusion

User-defined functions provide flexibility for complex operations, while lambda functions offer a quick and concise way to perform single-expression tasks. This document demonstrates how to use both types of functions and shows how to replicate user-defined functions using lambda functions in Python.