# Python for AI and Automation

# Chapter 2: Python Functions

## Learning Objectives

By the end of this lecture, students will:

1. Understand what functions are and why they are useful.
2. Learn how to define and call functions in Python.
3. Explore function parameters and return values.
4. Understand different types of functions (built-in, user-defined, lambda).
5. Practice writing efficient functions with exercises.

## 🔹 Introduction to Functions

### What is a Function?

A function is a block of reusable code that performs a specific task. Functions help in organizing code, avoiding repetition, and making programs easier to understand.

### Why Use Functions?

✅ Reduces code repetition  
✅ Improves code readability  
✅ Makes debugging easier

### 

### 

### Built-in vs. User-defined Functions

1. Built-in Functions: Predefined functions like print(), len(), type(), etc.
2. User-defined Functions: Functions created by programmers for custom tasks.

## 🔹 Defining and Calling Functions

### Syntax of a Function

# Defining a function

def greet():

print("Hello, students!")

# Calling a function

greet()

### Function with Parameters

def greet(name):

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

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

### Function with Return Value

def add(a, b):

return a + b

result = add(5, 3)

print(result) # Output: 8

## 🔹 Types of Function Arguments

### 1️⃣ Positional Arguments

def power(base, exponent):

return base \*\* exponent

print(power(2, 3)) # Output: 8

### 2️⃣ Default Arguments

def greet(name="Student"):

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

greet() # Output: Hello, Student!

greet("Emma") # Output: Hello, Emma!

### 3️⃣ Keyword Arguments

def describe\_pet(animal, name):

print(f"I have a {animal} named {name}.")

describe\_pet(animal="dog", name="Buddy")

describe\_pet(name="Whiskers", animal="cat")

### 4️⃣ Arbitrary Arguments (\*args, \*\*kwargs)

def sum\_all(\*numbers):

return sum(numbers)

print(sum\_all(1, 2, 3, 4, 5)) # Output: 15

## 🔹 Lambda (Anonymous) Functions

### What is a Lambda Function?

A lambda function is a small, anonymous function that can have any number of arguments but only one expression.

### Syntax:

lambda arguments: expression

### Example:

square = lambda x: x \* x

print(square(4)) # Output: 16

### Lambda with map()

numbers = [1, 2, 3, 4, 5]

squared = list(map(lambda x: x \*\* 2, numbers))

print(squared) # Output: [1, 4, 9, 16, 25]

## 🔹 Scope of Variables (Local & Global)

### Local Variable Example

def example():

x = 10 # Local variable

print(x)

example()

# print(x) # Error! x is not accessible outside the function

### Global Variable Example

global\_var = 50

def example():

print(global\_var) # Accessing global variable

example()

## 

## 

## 

## 📝

## Exercises for Students

### 1️⃣ Basic Function Creation

Write a function called welcome() that prints "Welcome to Python Programming!"

### 2️⃣ Function with Parameters

Create a function multiply(a, b) that takes two numbers and returns their product.

### 3️⃣ Function to Check Even or Odd

Write a function is\_even(n) that returns True if n is even and False otherwise.

### 4️⃣ Factorial Function

Write a function factorial(n) that returns the factorial of a given number.

### 5️⃣ Reverse a String

Write a function reverse\_string(s) that takes a string and returns it in reverse.

### 6️⃣ Use map() with Lambda

Use a lambda function inside map() to square all numbers in a list: [2, 4, 6, 8].

### 7️⃣ Find the Maximum Using reduce()

Use the reduce() function with a lambda to find the largest number in a list.

from functools import reduce

numbers = [3, 7, 1, 9, 2]

# Your code here

## 🎯 Summary

✔ Functions help in organizing and reusing code.  
 ✔ Python provides built-in and user-defined functions.  
 ✔ Parameters allow functions to accept values.  
 ✔ Functions can return values using return.  
 ✔ Lambda functions are useful for small, quick tasks.  
 ✔ Variable scope determines where a variable can be used.

### 📢 Homework Challenge

🔹 Write a function that takes a list of numbers and returns the sum of only the even numbers.