🚀 21-Day Python Challenge

Day 10 – Functions

🎯 Day 10 Learning Objectives

By the end of today's class, students will be able to:

- Understand what a **Function** is.
- Create and call Functions.
- Use **Parameters**.
- Understand **Arguments**.
- Return values using **return**.
- Learn **Variable Scope**.
- Build simple real-world programs using functions.

📓 What is a Function?

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

Instead of writing the same code many times, we write it once inside a function and use it whenever needed.

Real-Life Example

Think about a **TV Remote**.

```
Power Button

↓

TV Turns ON
```

Whenever you press the **Power Button**, the same action happens.

A Python function works exactly like this.

📷 Picture 1 – Function Concept

Function Syntax

```
def function_name():
    print("Hello Python")
```

Explanation

Keyword	Meaning
`def`	Define a function
`function_name`	Name of the function
`()`	Parameters go here
`:`	Function starts here

Example 1 – Simple Function

```
def welcome():
    print("Welcome to Python")

welcome()
```

Output

```
Welcome to Python
```

Example 2 – Calling a Function Multiple Times

```
def hello():
    print("Hello Students")

hello()
hello()
hello()
```

Output

```
Hello Students
Hello Students
Hello Students
```

📓 Parameters

Parameters receive information.

Example

```
def greet(name):
    print("Welcome", name)

greet("Rahul")
greet("Anitha")
```

Output

```
Welcome Rahul
Welcome Anitha
```

📷 Picture 2 – Parameters

Think of it as:

```
Function

↓

Name

↓

Rahul

↓

Welcome Rahul
```

Example 3 – Two Parameters

```
def student(name, marks):
    print("Name :", name)
    print("Marks:", marks)

student("Rahul", 92)
```

Return Values

Sometimes a function calculates something and sends the result back.

Example

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

result = add(10, 20)

print(result)
```

Output

```
30
```

Picture Explanation

```
10

↓

Function

↓

20

↓

Addition

↓

30 Returned
```

Example 4 – Area of Rectangle

```
def area(length, width):
    return length * width

result = area(10, 5)

print("Area =", result)
```

Output

```
Area = 50
```

Variable Scope

Variables created inside a function exist only inside that function.

```
def test():
    x = 100
    print(x)

test()
```

Output

```
100
```

Trying to print `x` outside the function gives an error because `x` is a **local variable**.

Local Variable

```
def display():
    name = "Rahul"
    print(name)

display()
```

Global Variable

```
college = "GK InfoTech"

def show():
    print(college)

show()

print(college)
```

Real-Time Program 1

Student Result

```
def result(marks):

    if marks >= 35:
        print("PASS")
    else:
        print("FAIL")

marks = int(input("Enter Marks : "))

result(marks)
```

Real-Time Program 2

Electricity Bill

```
def bill(units):

    amount = units * 8

    return amount

units = int(input("Units : "))

print("Bill =", bill(units))
```

Real-Time Program 3

Simple Calculator

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

def sub(a,b):
    return a-b

print(add(20,10))
print(sub(20,10))
```

Practice Programs

1. Create a welcome function.
2. Create a function to print your name.
3. Create a function to add two numbers.
4. Create a function to calculate the square of a number.
5. Create a function to check pass or fail.
6. Create a function to calculate the area of a rectangle.

🏆 Mini Project – Student Report

```
def student(name, marks):

    print("Name :", name)

    print("Marks :", marks)

    if marks >= 35:
        print("Result : PASS")
    else:
        print("Result : FAIL")

name = input("Enter Name : ")

marks = int(input("Enter Marks : "))

student(name, marks)
```

Interview Questions

1. What is a Function?
2. Why do we use Functions?
3. What is `def`?
4. What are Parameters?
5. What is a Return Value?
6. What is Variable Scope?
7. Difference between Local and Global Variables?

🌟 Day 10 Success Message

Today you learned:

- ✅ Creating Functions
- ✅ Calling Functions
- ✅ Parameters
- ✅ Return Values
- ✅ Variable Scope

"Functions help us write cleaner, shorter, and reusable code. Professional programmers build software by combining many small functions."

🚀 21-Day Python Challenge

Day 11 – Modules

🎯 Day 11 Learning Objectives

Students will learn:

- What is a Module?
- Why Modules are useful.
- How to use `import`.
- Use the **math** module.
- Use the **random** module.
- Use the **datetime** module.

📓 What is a Module?

A **Module** is a Python file that contains ready-made functions and tools.

Instead of writing everything yourself, you can **import** a module and use its features.

Real-Life Example

Think of a **Tool Box**.

```
🧰 Toolbox
↓
Hammer
↓
Screwdriver
↓
Wrench
```

Instead of making your own tools, you simply use the toolbox.

Python Modules work the same way.

📷 Picture 3 – Module Concept

Importing a Module

```
import math
```

Now Python loads the **math** module.

📓 Math Module

Used for mathematical calculations.

Example 1

```
import math

print(math.sqrt(25))
```

Output

```
5.0
```

Example 2

```
import math

print(math.pow(2,5))
```

Output

```
32.0
```

Example 3

```
import math

print(math.ceil(4.2))
print(math.floor(4.8))
```

Output

```
5
4
```

Random Module

Used to generate random values.

Example 1

```
import random

print(random.randint(1,10))
```

Output

```
Random Number between 1 and 10
```

Example 2

```
import random

colors = ["Red","Green","Blue"]

print(random.choice(colors))
```

Output

```
Any one color
```

Picture 4 – Random Choice

Datetime Module

Used to work with dates and time.

Example 1

```
import datetime

today = datetime.datetime.now()

print(today)
```

Example 2

```
import datetime

today = datetime.date.today()

print(today)
```

Example 3

```
import datetime

current_time = datetime.datetime.now()

print(current_time.strftime("%H:%M:%S"))
```

Real-Time Program 1

Lucky Number

```
import random

number = random.randint(1,100)

print("Lucky Number :", number)
```

Real-Time Program 2

Today's Date

```
import datetime

print("Today's Date")

print(datetime.date.today())
```

Real-Time Program 3

Square Root Calculator

```
import math

number = int(input("Enter Number : "))

print("Square Root =", math.sqrt(number))
```

Mini Project

Student Utility Program

```
import random
import datetime
import math

print("Today's Date :", datetime.date.today())

marks = random.randint(35,100)

print("Random Marks :", marks)

print("Square Root of Marks :", math.sqrt(marks))
```

Common Modules

Module	Purpose	Example
`math`	Mathematical calculations	`math.sqrt(25)`
`random`	Random numbers and choices	`random.randint(1,10)`
`datetime`	Date and Time	`datetime.date.today()`

Assignment

Write programs to:

1. Find the square root of a number.
2. Generate a random number from 1 to 50.
3. Select a random fruit from a list.
4. Display today's date.
5. Display the current time.
6. Build a lucky number generator.

Interview Questions

1. What is a Module?
2. Why do we use `import`?
3. Name three built-in Python modules.
4. What does `math.sqrt()` do?
5. What does `random.randint()` do?
6. How do you display today's date?

🏆 Day 11 Challenge – Student ID Generator

Write a Python program that:

- Accepts the student's name.
- Generates a random Student ID between **1000 and 9999**.
- Displays today's date.
- Prints a welcome message.

Sample Output

```
===== STUDENT ID CARD =====
Name       : Rahul
Student ID : 4827
Date       : 2026-01-15

Welcome to GK InfoTech Solutions!
```

🌟 Day 11 Success Message

Congratulations! 🎉

Today you learned how to use Python's built-in **Modules**, making your programs more powerful without writing everything from scratch.

You explored:

- ✅ Importing Modules
- ✅ `math` Module
- ✅ `random` Module
- ✅ `datetime` Module

"Great programmers don't reinvent the wheel—they use modules to solve problems faster. Learning to use Python's built-in modules is an important step toward becoming a professional programmer."

🚀 Next Lesson (Day 12): File Handling – Reading Files, Writing Files, Appending Data, and Working with Text Files.