Training Day 01 Report

23 June, 2025

Topic: Introduction to Python Programming – Basics

Objectives of the Day

• To introduce students to the fundamentals of Python programming.
• To develop an understanding of basic programming constructs including variables, operators, data types, conditional statements, loops, typecasting, operator overloading, and basic pattern printing.

Topics Covered

1. Introduction to Python

• Python is a high-level, interpreted, and general-purpose programming language.
• It emphasises code readability and allows programmers to express concepts in fewer lines of code.
• Widely used in domains such as web development, data analysis, artificial intelligence, automation, and machine learning.

2. Variables

• Variables are containers used to store data values.
• In Python, variable declaration does not require explicitly defining the data type (dynamic typing).
• Example: name = "John" age = 25

3. Operators

• Arithmetic Operators: +, -, *, /, //, %, **
• Assignment Operators: =, +=, -=, *=, /=
• Comparison Operators: ==, !=, >, <, >=, <=
• Logical Operators: and, or, not
• Bitwise Operators: &, |, ^, ~, <<, >>

4. Data Types

• Numeric: int, float, complex
• Text Type: str
• Boolean: bool
• Sequence Types: list, tuple, range
• Set Types: set, frozenset
• Mapping Type: dict Example: x = 10 # Integer pi = 3.14 # Float name = "Alice" # String values = [1, 2, 3] # List

5. Conditional Statements (if-else)

• Used to execute code blocks based on specified conditions. Example: if age >= 18: print("Eligible to vote") else: print("Not eligible")

6. Loops (For Loop)

• A for loop is used to iterate over sequences like lists, strings, or a range of numbers. Example: for i in range(5): print(i)

7. Operator Overloading

• Python allows customization of standard operators for user-defined classes. Example: a = 10 b = 20 c = “hello” d = “world” a+b c+d

8. Typecasting

• Implicit Typecasting: Done automatically by Python.

• Explicit Typecasting: Manually converting data types using functions like int(), float(), and str(). Example: x = "10" y = int(x) + 5 # Output: 15

9. Pattern Printing

• Helps build logic and control flow understanding through nested loops. Example: for i in range(1, 6): print("*" * i)

Output:

```
    *
   **
  ***
 ****
*****
```

programs covered :

http://localhost:8888/files/day1(23%20june%2C%202025).ipynb?_xsrf=2%7C361123c9%7C23002d7340ebc6a2a92afe9f14528c92%7C17507696844

Summary

The first day focused on the foundational concepts of Python. Students learned about variable declarations, operators, control structures, data types, and loops. Practical demonstrations and small coding tasks were given to reinforce each concept.