Chapter 11: Python Basics

Introduction

Python is one of the most popular and beginner-friendly programming languages used in Artificial Intelligence and many other domains. It is easy to learn, has simple syntax, and comes with powerful libraries and tools. In this chapter, we will explore the basic elements of Python, laying the foundation for developing AI-based solutions.

11.1 What is Python?

Python is a **high-level**, **interpreted**, and **object-oriented** programming language developed by **Guido van Rossum** and first released in **1991**.

Key Features of Python:

- Simple and easy to learn
- Open-source and free to use
- Portable (can run on different operating systems)
- Interpreted language (no need for compilation)
- Large standard library
- Extensive third-party libraries for AI (e.g., NumPy, Pandas, Scikit-learn)

11.2 Python Environment Setup

You can write Python code using:

- **IDLE** (comes with Python installation)
- **Jupyter Notebook**
- **Online compilers** like Replit, Google Colab
- **VS Code / PyCharm**

To install Python:

- Download from python.org
- Follow the installation instructions for your OS

11.3 Python Syntax and Indentation

Python uses **indentation** to define blocks of code. No curly braces `{}` are used.

```
if 10 > 5:
    print("10 is greater than 5")
```

Note: Incorrect indentation will lead to errors.

11.4 Variables and Data Types

Variables

- Used to store values.
- No need to declare data type explicitly.

```
name = "AI"
age = 15
is_student = True
```

Common Data Types

Data Type	Example
`int`	10
`float`	3.14
`str`	"Python"
`bool`	True / False
`list`	[1, 2, 3]
`tuple`	(1, 2, 3)
`dict`	{"name": "AI"}

11.5 Input and Output

Input

To take user input:

```
name = input("Enter your name: ")
```

Output

To display output:

```
print("Welcome to Python")
```

11.6 Operators in Python

Arithmetic Operators

+, -, *, /, //, %, **

Comparison Operators

==, !=, <, >, <=, >=

Logical Operators

and, or, not

```
a = 10
b = 5
print(a > b and b < 10)  # True
```

11.7 Conditional Statements

Used to make decisions in code.

```
if condition:
    # code
elif another_condition:
    # code
else:
    # code
```

Example:

```
age = 18
if age >= 18:
    print("Eligible to vote")
else:
    print("Not eligible")
```

11.8 Loops in Python

For Loop

```
for i in range(5):
    print(i)
```

While Loop

```
i = 1
while i <= 5:
    print(i)
    i += 1
```

11.9 Lists and Strings

Lists

- Mutable

```
fruits = ["apple", "banana", "cherry"]
print(fruits[0])
fruits.append("orange")
```

Strings

- Immutable

```
name = "Python"
print(name[0])
print(name.upper())
```

11.10 Functions

Used to reuse code.

```
def greet():
    print("Hello, AI Student")

greet()
```

Function with Parameters:

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

result = add(3, 5)
print(result)
```

11.11 Introduction to Libraries

Python has rich libraries that make it powerful for AI:

- `math` – for mathematical operations
- `random` – for generating random numbers
- `statistics` – for mean, median, etc.

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

11.12 Error Handling

Try-Except Block

Used to handle runtime errors without crashing the program.

```
try:
    print(10 / 0)
except ZeroDivisionError:
    print("Cannot divide by zero")
```

Chapter Summary

- Python is a versatile and beginner-friendly language, perfect for AI applications.
- It uses indentation instead of braces.
- Python supports variables, different data types, and operators.
- You learned how to take inputs and display outputs using `input()` and `print()`.
- Conditional statements and loops allow decision-making and repetition.
- Lists and strings are basic data structures.
- Functions help organize and reuse code.
- Libraries like `math` and `random` are useful tools.
- Basic error handling ensures smooth execution of code.