# 

# 

# Section 1: Python Basics

**Introduction to Python:**

* What is Python?
* Why Python for backend development?
* Basic syntax and data types (numbers, strings, lists, dictionaries)

**Variables and Operators:**

* Declaring and assigning variables
* Arithmetic, comparison, and logical operators

**Control Flow:**

* Conditional statements (if, else, elif)
* Loops (for, while)

## **Python Basics**

### **Introduction to Python**

Python is a high-level, interpreted programming language known for its readability and versatility. Its simplicity makes it an excellent choice for beginners, while its powerful libraries and frameworks make it suitable for complex applications.

**Why Python for Backend Development?**

* **Readability:** Python's clean syntax promotes code understandability.
* **Large community:** Extensive support and resources are available.
* **Versatility:** Suitable for various backend tasks (web development, data science, machine learning).
* **Productivity:** Rapid development due to its interpreted nature and high-level abstractions.

### **Basic Syntax and Data Types**

Python uses indentation to define code blocks, making it visually appealing.

**Data Types:**

* **Numbers:**
  + Integers (e.g., 4, 5 , 99)
  + Floating-point numbers (e.g., 3.14, 2.73)
* **Strings:** Sequences of characters (e.g., "Hello, world!", “Bye bye”)
* **Lists:** Ordered collections of items (e.g., [1, 2, 3, "four"])
* **Dictionaries:** Unordered collections of key-value pairs (e.g., {"name": "Alice", "age": 30})

# Example of data types

**x = 10 # Integer**

**y = 3.14 # Float**

**name = "Python" # String**

**fruits = ["apple", "banana", "orange"] # List**

**person = {"name": "Bob", "city": "New York"} # Dictionary**

### **Variables and Operators**

Variables are used to store data values.

# Assigning values to variables

**a = 5**

**b = "Hello"**

**c = True**

**# Arithmetic operators**

**result = a + 3 # Addition**

**result = a - 2 # Subtraction**

**result = a \* 4 # Multiplication**

**result = a / 2 # Division**

**# Comparison operators**

**is\_equal = (a == 5) # Checks if a is equal to 5**

**is\_greater = (a > 3) # Checks if a is greater than 3**

**# Logical operators**

**condition1 = True**

**condition2 = False**

**result = condition1 and condition2 # Logical AND**

**result = condition1 or condition2 # Logical OR**

**result = not condition1 # Logical NOT**

### **Control Flow**

Conditional statements and loops allow you to control the program's flow.

**# Conditional statements**

**if a > 0:**

**print("a is positive")**

**else:**

**print("a is non-positive")**

**# Loops**

**for i in range(5):**

**print(i)**

**count = 0**

**while count < 3:**

**print(count)**

**count += 1**

**Remember to run each code block to see the output.**

This Jupyter Notebook provides a foundation for Python basics.