

Chapter 1: Introduction to Python

1.1 What is Python?

Python is a general-purpose, dynamic, high-level and interpreted programming language.

- It is designed to be simple and easy to learn, making it an ideal choice for beginners.
- One of the key strengths of Python is its versatility.
- Python supports the object-oriented programming approach, allowing developers to create applications with organized and reusable code.

1.2 Features of Python

Readability:

Python's syntax is designed to be clear and readable, making it easy for both beginners and

1.5 First Python Program

Writing your first Python program is an exciting step toward learning the language. Here's a simple example to get you started:

```
# Printing Hello World Using Python  
print("Hello World")
```

Let's break down the code:

- The `print()` function is used to display the specified message or value on the console.
- In this case, we pass the string "Hello World" as an argument to the `print()` function.

M	T	W	T	F	S	S
Page No.:						YOUVA
Date:						

- The string is enclosed in double quotes.
- The `#` symbol indicates a comment in Python. Comments are ignored by the interpreter and are used to provide explanations or notes to the code.

Chapter 3 : Variables.

Data Types & Keywords

3.1 Variables in Python

In Python, variables are used to store values that can be used later in a program. You can think of variables as containers that hold data. What makes Python unique is that you don't need to explicitly declare the type of a variable. You simply assign a value to a variable using the "=" operator.

- For example, you can create a variable called "name" and assign it a value like this:

```
name = "Yadnyesh"
```

Here, "name" is the variable name, and "Yadnyesh" is the value assigned to it. Python will automatically determine the type of the variable.

Chapter : 4 Operators in Python

4.1 Operators in Python

Operators in Python are symbols or special characters that are used to perform specific operations on variables and values.

Python provides various types of operators to manipulate and work with different data types. Here are some important categories of operators in Python:

- Arithmetic operators
- Comparison operators
- Assignment operators
- Logical operators
- Bitwise operators
- Membership operators
- Identity operators
- Arithmetic operators.

Chapter 5 Python Basics to Advanced Control Flow

Control Flow directs program execution through structures like loops, conditionals, and functions, determining the order and path of operations.

IF statements:

An if statement in Python checks whether a condition is true or false. If the condition is true, the code inside the if block runs. If False, the code is skipped. It's used to make decisions in the program, executing specific actions based on conditions.

example:

```
age = 18
if age >= 18:
    print("You are an adult")
else:
    print("You are a minor")
```