TOPS Technology

Python Fundamentals

Presented By:

Nandni Vala

Core Python Concepts

- 1.Understanding data types: integers, floats, strings, lists, tuples, dictionaries, sets.
- **Integers (int)**
- **Description**: Whole numbers, positive or negative, without a fractional component.
- **Example**:
- x = 42
- y = 40
- **Operations**: Addition, subtraction, multiplication, division, modulus, etc.
- result = x + y # Output: 35
- **Floats (float)**
- **Description**: Numbers with decimal points, representing real numbers.

- **Example**:
- pi = 3.14
- temperature = -4.5
- **Operations**: Support arithmetic operations like integers but include decimal precision.
- result = pi * 2 # Output: 6.28
- **Strings (str)**
- **Description**: Text data, enclosed in single, double, or triple quotes.
- **Example**:
- greeting = "Hello, World!"
- multiline = """This is
- a multiline string."""
- **Operations**:
  - Concatenation: "Hello" + " " + "World"
  - Slicing: greeting[:5] # Output: Hello

- **Lists (list)**
- **Description**: Ordered, mutable collection of items (can contain mixed types).
- **Example**:
- fruits = ["apple", "banana", "cherry"]
- mixed = [1, "two", 3.0]
- **Tuples (tuple)**
- **Description**: Ordered, immutable collection of items.
- **Example**:
- coordinates = (10, 20)
- single = (42,) # Single-element tuple requires a trailing comma
- **Dictionaries (dict)**
- **Description**: Unordered collection of key-value pairs.
- **Example**:
- person = {"name": "Alice", "age": 25}

- **Sets (set)**
- **Description**: Unordered collection of unique items.
- **Example**:
- unique_numbers = {1, 2, 3, 3} # Duplicates are removed

2.Python variables and memory allocation.

- **variables** are references to objects stored in memory. Python manages memory efficiently through its **dynamic typing** and **automatic memory management** features.
- **Variables in Python**
- **Definition**: A variable is a name that points to an object in memory.
- **Dynamic Typing**: Variables can change their type during execution.
- x = 10 # `x` is an integer
- x = "hello" # Now `x` is a string

- **Memory Allocation**
- Python's memory management involves two main components:
- **Heap Memory**: Stores objects and data (managed by Python's memory manager).
- **Stack Memory**: Stores references to objects and function calls.

3.Python operators: arithmetic, comparison, logical, bitwise.

- Python operators: arithmetic, comparison, logical, bitwise.

Operator	Description	Example
+	Addition	1+2
-	Subtraction	4-2
*	Multiplication	4*5
/	Division	8/2
%	Modulus	8%2
//	Floor Division	8//2

- **Comparison Operators :**
- Used to compare two values, returning True or False.

Operator	Description	Example
==	Equal to	5 == 3
>	Greater than	5 > 3
<	Less than	5 < 3
<=	Greater than or equal	5 <=3
>=	Less than or equal	5 >=3
!=	Not equal to	5 != 3

- **Logical Operators**
- Used to combine conditional statements.

Operator	Description	Example
&&	AND	(5>2) && (2 < 4)
||	OR	(5>2) ||(2 < 4)

- **Bitwise Operators**
- Operate at the bit level, manipulating binary representations of integers.

Operator	Description	Example
&	AND	5 & 3
^	XOR	5 ^ 3
~	NOT	5 ~ 3
>>	RIGHT SHIFT	5 >> 3
<<	LEFT SHIFT	5 << 3