TOPS TECHNOLOGY

Python – Collections, functions and Modules

Presented By:
Nandni Vala

Functions

1.Defining functions in Python.

- functions are defined using the def keyword, followed by the function name and parentheses for parameters.
- Syntax:
- def function_name(parameters):
- return value
- Example:
- def add(a, b):
- return a + b
- result = add(3, 5)
- # Output: 8
- **Parameters**: Input to the function.
- **Return**: Sends a result back to the caller.

2.Different types of functions: with/without parameters, with/without return values.

- **Function Without Parameters and Without Return Value**
- A function that doesn't take any inputs and doesn't return anything.
- def greet():
- print("Hello, World!")
- greet()
- Output: Hello, World!
- **Function With Parameters and Without Return Value**
- A function that accepts parameters but doesn't return anything.
- def greet(name):
- print(f"Hello, {name}!")
- greet("Alice")
- Output: Hello, Alice!

- **Function Without Parameters but With Return Value**
- A function that doesn't take any parameters but returns a value.
- def get_five():
- return 5
- result = get_five()
- Output: 5
- **Function With Parameters and With Return Value**
- A function that accepts parameters and returns a value.
- def add(a, b):
- return a + b
- result = add(3, 5)
- Output: 8

3.Anonymous functions (lambda functions).

- A lambda function is a small anonymous function.
- a lambda function can tack any number of arguments,but can only have one expression.
- Example :
- s = lambda a,b,c:a+b+c
- print(s(1,3,9))
- Output :
- S = 13

3.Counting occurrences of characters in a string using dictionaries.

- Explanation:
- **Initialization**: An empty dictionary char_count is created to store each character as a key and its count as the value.
- **Iteration**: Loop through each character in the string.
- **Updating the dictionary**:
- If the character is already a key in the dictionary, its value is incremented.
- If not, the character is added as a new key with the value set to 1.
- **Output**: The dictionary containing the character counts is returned.

- Example :
- def count_characters(string):
- char_count = {}
- for char in string:
- char_count[char] = char_count.get(char, 0) + 1
- return char_count
- print(count_characters("hello world"))
- Output:
- {'h': 1, 'e': 1, 'l': 3, 'o': 2, ' ': 1, 'w': 1, 'r': 1, 'd': 1}
-