Training Report Day-11

18 June 2024

➢ What are Functions?

Python Functions is a block of statements that return the specific task. The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can do the function calls to reuse code contained in it over and over again.

Some Benefits of Using Functions:-

- Increase Code Readability
- Increase Code Reusability

❖ FUNCTIONS:-

In Python, a function is defined using the def. keyword, followed by the function name, parentheses containing any parameters, and a colon. The function body is indented and contains the code to be executed. .

Types of Functions

1. Built-in Functions: These are pre-defined functions provided by a programming language or environment. Examples include print (), Len () in Python, and printf () in C.

2. User-defined Functions: These are functions created by the programmer to perform specific tasks. They follow a defined structure with a name, parameters, and a body.

3. Anonymous Functions (Lambda Functions): These are functions without a name, often used for short, simple operations

4. Recursive Functions: These functions call themselves within their definition, useful for tasks that can be divided into similar subtasks.

Example:-

- #define a function

  #syntax

  def function_name(paramenters):

```
        #statement
            Pass
```

- ```
  # A lambda function that adds 10 to the number passed as an argument
  add_ten = lambda x: x + 10
  # Using the lambda function
  result = add_ten(9)
  print(result)
  ```
- ```
  def factorial(n):
  # Base case: if n is 0, return 1
  if n == 0:
      return 1
  # Recursive case: n! = n * (n-1)!
  else:
      return n * factorial(n - 1)

  # Get user input
  number = int(input("Enter a number to calculate its factorial: "))

  # Ensure the input is a non-negative integer
  if number < 0:
      print("Please enter a non-negative integer.")
  else:
      # Using the recursive function
      result = factorial(number)
      print(f"The factorial of {number} is {result}")
  ```

- **Arguments in Functions:-**

Arguments are values passed to a function when it is called. They can be categorized as follows:

1. Positional Arguments: These are arguments that are passed to a function in a specific order.
2. Keyword Arguments: These are arguments passed to a function by explicitly naming each parameter and its corresponding value.
3. Default Arguments: These are arguments that assume a default value if no value is provided during the function call
4. Variable-length Arguments: *args (Non-keyword arguments): Allows a function to accept any number of positional arguments.
**kwargs (Keyword arguments): Allows a function to accept any number of keyword arguments