Training Report Day-6

12 June 2024

Functions in python:

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

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.

```
#define a function
#syntax
def function_name(paramenters):
  #statement
  pass
```

```
#example
def greet(name):
```

```
  print(f"welcome {name}! ")
#call a function
greet("taran")

#Lambda function
add=lambda x,y:x+y
print(add(2,3))
```

```
# 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)
```

```
# 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)
```