14/05/25

* Function in Python:

Functions are reusable block of Code

→ Types of Functions:

a.) Pre-defined functions → Print(), Sum(), input(), append().

b.) User-defined functions →

1.) Step-1 = using def keyword.

2.) Step2 - Name of function.

3.) Step 3- logic or body of function

4.) Step 4- Call the function (Name of function).

→ Syntax:

```
def function name():
    logic
name-function() - calling.
```

1.) function to add two numbers?

```
def add():  — function declaration
    a = 10
    b = 5
    c = a+b
    print(f"Sum is : {c}")
```

logic code block

add() - calling the function.

2.) function to add square of a number till given value.

~~i = 1~~

```
def my_squ():
    i = 1
    n = 3
    sum = 0
    while (i <= n):
        sum += i**2
        i += 1
    print(f"total square value is : {sum}")
my_squ()
```

3.) ~~n = 1234~~ Function to add digits:

```
def digit_add():

    n = 1234
    sum = 0

    while(n>0):

        digit = n % 10
        sum += digit
        n //= 10

    print(f" total sum is: {sum}")

digit_add()
```

* Arguments and Parameters :-

i.) Parameters :

These are values which we pass when we create a function.

ii.) Arguments :

These are values which we pass when we call a function.

D-1.) Program / Function of Addition.

```
def my_add(a,b): # (a,b) are parameters
    c = a+b
    print("Sum is:" c)

my_add(12, 10) # (12,10) are arguments
```

0.2.) Function to check even or odd ?

```
def even_or_odd(a):
    a = int(input("Enter a Number: "))
    if(a%2==0):
        print("Even Number")
    else:
        print("Odd Number")

even_or_odd()
```

DATE
PAGE

* Lambda function :-

A Lambda function can take any number of arguments but can have only one expression.

Syntax :

lambda arguments : expression

1.) Ex-1 :

```
x = Lambda a : a+5
print(f" total is : {a(8)} ")
```

2.)

```
x = Lambda a, b : a * b
print(f" total is : {a(5,3)} ")
```

3.)

```
x = Lambda a, b, c : a+b+c
print(f" Sum is : {x(3,3,4)} ")
```

* Arguments in functions :-

i.) *args (Non- Keyword Arguments) :

args allows us to pass multiple arguments to a function. It collects them into a tuple.

```
def add_numbers(*args):
    total = sum(args)
    print("Sum is:", total)

add_numbers(5, 10, 15)
```

ii.) **kwargs (Keyword Arguments) :-

**kwargs allow us to pass multiple keyword arguments (name-value pairs).

- It Collects them to a dictionary.

```
def print_details(**kwargs):
    for key, value in kwargs.items():
        print(f"{key}: {value}")

print_details(name = "Amaan", age = 22)
```

3. iii) Using *args and **kwargs together:

- We can use both in the same function but args must come before *kwargs.

```
def display_info(*args, **kwargs):
    print("Positional arguments:", args)
    print("Keyword arguments:", kwargs)

display_info(1, 2, 3, name = "Saurabh", age=25)
```

iv.) Required arguments :-

- These are arguments that must be passed to a function otherwise, Python throws an error, There is no default value for these arguments.

```
def greet(name):
    print("Hello", name)
greet("Amaan")
```

v.) Default arguments : (optional)

- These arguments have a default value assigned. If no value is provided, Python uses the default.

```
def greet(name = "Guest"):
    print(f"Hello {name}!")
greet()
```

3.)

vi.) Return keyword:

The Return keyword sends a value back from a function; if returns is not used

Questions:-

1.) Find the maximum number using *args :-

```
def find_max(*args):
    returns max(args)

print(find_max(2, 100, 5, 10))
```

output - 100.

2.) Calculate product of numbers using *args

```
def Product(*args):
    result = 1
```

```
    for i in args:
        result *= i
    return result

print(product(2,3,4))
```

output- 24

3.) Generate a full name using **kwargs:-

```
def full_name(**kwargs):
    return f"{kwargs.get('first_name','unknown')} {kwargs.get('last_name','Mehta')}"
```

4.) Count vowels in a Given String using return:

```
def Count_Vowel(string):
    Vowels = 'aeiou AEIOU'
    return sum(1 for char in string in vowels)
```

5.) Create a Shopping Cart using args and *kwargs

```
def Shopping_Cart(*items, **prices):
    total = sum(prices[items] for item in
                items if item in prices)
    return total

print(Shopping_cart("apple", "Mango", apple=30,
                    *Mango = 20)
```