Dt: 11/12/25
Day - 10

what is a function?

A function is a block of reusable code that performs a specific task.
You write it once & use it many times.

why functions use?

- Avoid repeating code
- make programs cleaner & shorter
- make debugging easy
- improve code organization

Syntax of functions:

```
def function_name():
    # code
    return value
```

Types of functions:

1. Built in functions
   Provided by Python [print(), len(), input()... etc]
2. User defined functions
   created by the programmer using def
3. Lambda functions
   Anonymous single line function created with lambda

1. Built in functions:-

These functions come Pre-installed with Python

Example:-

```
PRint ("hello")
len ("python")
type (10)
input ("enter a value")
```

You don't have to define them - just use them.

2. user defined functions:-

created by the programmer using def

Syntax:-

```
def function_name(P1, P2 .... Pn):
    code block
    return value
```

→ methods to create user-defined functions:

1. without input & without return value
2. with input & without return value
3. without input & with return value
4. with input & with return value.

1. without input & without return value.

the function does not take any parameters & does not return anything

Syntax:-

```
def fun_name():
    code block
call function
fun_name():
```

Example:-

```
def even_odd():
    num = int(input("enter a value"))
    if (num%2==0):
        print(num, "is even")
    else:
        print(num, "is odd")
```

output:-

```
even_odd()
enter a value: 25
25 is odd.
```

2. Without input and without return value

The function takes Parameters but does not return valu

Syntax:

```
def fun_name (P1, P2, ... Pn):
    code block
```

Call function:

```
fun_name (P1, P2, ..., Pn):
```

Example:

```
def even_odd_1(n):
    if(n%2==0):
        print (n, "is even")
    else:
        print (n, "is odd")
```

output:

```
even_odd_1(50)
50 is even
```

3. without input and with return value.

The function does not take parameters, but returns a value.

Syntax:

```
def fun_name():
    code block
    return value
```

call function:

```
var = fun_name():
```

Example:

```
def even_odd_2():
    num= int(input("enter a value"))
    if (num%2==0):
        return num," even"
    else:
        return num, "odd"
```

output:

```
even_odd_2()
enter a value 25
```

4. with input & with return value.

the function takes parameters & returns a value.

Syntax:

```
def fun_name (P1, P2, ---- Pn):
    code block
    return value.
```

call function:

```
var = fun_name (P1, P2, ---- Pn)
```

Example:

```
def even_odd_3 (n):
    if (n%2 == 0):
        return n, "even"
    else:
        return n, "odd"
```

output:

```
even_odd_3 (25)
(25, 'odd')
```

write a program to print prime numbers between 1 to 10

```
~~def sum_of~~
    for num in range (1, 11):
        if num > 1:
            is_prime = True
            for i in range (2, num):
                if num % i == 0:
                    is_prime = False
                    break
            if is_Prime:
                print (num)
```

output:-

2 3 5 7