URBAN EDGE

Chapter 8 - Functions & Recursions

A function is a group of statements performing a specific task.

When a program gets bigger in size and its complexity grows, it gets difficult for a programmer to keep track on which piece of code is doing what!

A function can be reused by the programmer in a given program any number of

Example and Syntax of a function
The syntax of a function looks as follows:

```
def func1():
    print("Hello")
```

This function can be called any number of times, anywhere in the program.

Function Call
Whenever we want to call a function, we put the name of the function followed by parenthesis as follows:

```
func1()
```

→ This is called function call

Function definition
The part containing the exact set of instructions which are executed during the function call.

URBAN EDGE

Quick Quiz : Write a program to greet a user with "Good day" using functions.

Types of functions in Python
There are two types of functions in Python:
1. Built in functions → Already present in Python
2. User defined functions → Defined by the user

Examples of built in function includes len(), print(), range() etc.

The func1() function we defined is an example of user defined function

Functions with arguments
A function can accept some values it can work with. We can put these values in the parenthesis
A function can also return values as shown below:

```
def greet(name):
    gr = "Hello " + name
    return gr

a = greet("Harry")
```

→ "Harry" is passed to greet in name

→ a will now contain "Hello Harry"

Default Parameter Value
We can have a value as default argument in a function.
If we specify name = 'stranger' in the line containing def, this value is used when no argument is passed.

For example:

```
def greet(name = "Stranger"):
        # function body
```

greet() → Name will be "Stranger" in function body (default)
greet("Harry") → Name will be "Harry" in function body (passed)

Recursion

Recursion is a function which calls itself
It is used to directly use a mathematical formula as a function. For example:

factorial(n) = n × factorial(n-1)

This function can be defined as follows:

```
def factorial(n):
    if i==0 or i==1:  → Base condition which doesn't call the function any further
        return 1
    else:
        return n * factorial(n-1)  → Function calling itself
```

This works as follows:

Factorial(4)
↓ ↳ [Function called]
4 × Factorial(3)
4 × [3 ↓ × factorial(2)]
4 × 3 × [2 ↓ × factorial(1)]
4 × 3 × 2 × [1] [Function returned]

The programmer need to be extremely careful while working with recursion to ensure that the function doesnt Infinitely keep calling itself.
Recursion is sometimes the most direct way to code an algorithm