7) Functions and Methods.

- **Defining and calling functions in Python.**

  What is Function?

  Functions is block of code we can use it again and again.

Types of Function :

1. Built in Function (print(), input())
2. User Define

Functions are fundamental building blocks in Python that allow you to organize and reuse code. Here's a comprehensive explanation of how to define and call functions in Python.

Basic Syntax

```
def function_name(parameters):
    """docstring (optional)"""
  # function body
return value # optional
```

Calling Functions

greet("nayan") # Output: Hello, nayan!

1. Simple Function

In Python, you define a function using the def keyword:
python

```
def naam(name):
    print("Your Name:", name)
naam("Rahul")
```

2. Calling a Function

```
# Function define
def great():
    print("Hello, welcome to Python!")
# Function call
great()
```

3. Function With Parameter And Return

```
def add(n1,n2):
    return n1+n2
n1=int(input("Enter Number 1 : "))
n2=int(input("Enter Number 2 : "))
n=add(n1,n2)
print(n)
```

4. Parameters And Arguments

```
def add(n1,n2):
    print("Addition : ",n1+n2)

n1=int(input("Enter Number 1 : "))
n2=int(input("Enter Number 2 : "))
add(n1,n2)
```

5. Function Without Parameter With Return Type.

```
def fun1():
    n1 = int(input("Enter Number 1 "))
    n2 = int(input("Enter Number 2 "))
    return n1+n2
fun1()   # return ho chuka hai but console screen me dikhe ga nahi because they are not print
```

- **Function arguments (positional, keyword, default).**

1. **Positional Arguments.**

   Positional arguments are passed to the function in order, matching the order of the parameters.

   The order is very important because python assign values to parameters based on their position.

2. **Keyword Arguments**

   Keyword arguments are passed by explicitly specifying the parameter name along with its value.

   The order of keyword arguments does not matter because Python uses the parameter names to assign values.

3. **Default Arguments**

   Default arguments allow you to assign default values to parameters in the function definition.

   Parameters that assume a default value if no argument is provided. Characteristics:

If an argument is **not provided** during the function call, the default value is used.

- **Scope Of Variable In Python.**

1. **Local Scope**

   A variable declared inside a function is local to that function.

   It cannot be accessed outside the function.

2. **Global Scope**

   A variable declared outside any function is global.

   it can be accessed anywhere in the program, including inside functions (read-only by default).

3. **Enclosing Scope (Non-local)**

   For nested functions, variables from the outer function are accessible in the inner function (but not modifiable unless declared as nonlocal)

4. **Built-in Scope**

   Python's built-in names (like print, len, str, etc.) have the widest scope and are available everywhere.

- **Built-in methods for strings, lists, etc.**

Python provides many useful built-in methods for working with strings, lists, dictionaries, tuples, and sets. Here's a comprehensive overview:

1. **String Method**

   Python strings are immutable sequences of Unicode characters with these key method categories:

   Case Manipulation Methods

   - upper()/lower(): Convert case
   - title(): Title case conversion
   - capitalize(): Capitalize first character
   - swapcase(): Swap cases

Search and Validation Methods

- find()/index(): Locate substrings
- startswith()/endswith(): Check prefixes/suffixes
- isalpha()/isdigit()/isalnum(): Character type checks

Transformation Methods

- replace(): Substring replacement
- split()/rsplit(): String splitting
- join(): Sequence concatenation
- format(): String formatting

2. List Method

Lists are mutable sequences with these fundamental operations:

Modification Methods

- append(): Add single element
- extend(): Add multiple elements
- insert(): Insert at position
- remove()/pop(): Element removal

Organizational Methods

- sort(): In-place sorting
- reverse(): Order reversal
- copy(): Shallow copying

Information Methods

- index(): Position finding
- count(): Occurrence counting

3. **Directory Methods**

   Dictionaries are mutable mappings with these core operations:

   Access Methods

   - get(): Safe key access
   - keys()/values()/items(): View objects
   - setdefault(): Get with default insertion

Modification Methods

   - update(): Multiple key updates
   - pop()/popitem(): Item removal
   - clear(): Complete emptying