TOPSTechnologies

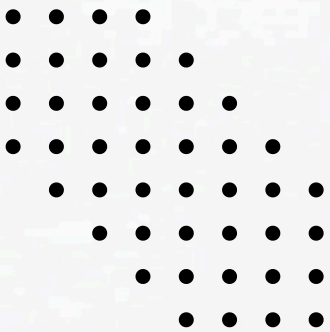
Functions And Methods

Presented for :

TOPs Technologies

Presented by :

Sumit B Yadav



Que 1

Defining Functions

In Python, a function is defined using the def keyword, followed by the function name, parentheses (), and a colon :. The function body is indented and contains the code that will execute when the function is called.

Calling Functions

A function is called by writing its name followed by parentheses (). If the function requires arguments, you pass them inside the parentheses.

Example -

```
# Defining a function that prints a greeting
def greet():
    print("Hello, World!")

# Calling the function
greet()
```

Que. 2

function arguments can be passed in various ways, including positional arguments, keyword arguments, and default arguments. Each type of argument has different rules for how it is passed and how it is handled in the function.

1. Positional Arguments

Positional arguments are the most basic type of arguments. They are passed to a function in the order in which they are defined in the function's parameter list.

2. Keyword Arguments

Keyword arguments are passed to the function by explicitly specifying the parameter name in the function call. The order of the arguments does not matter when using keyword arguments, as long as the correct parameter names are used.

3. Default Arguments

Default arguments allow a function to have optional parameters. If an argument is not provided when calling the function, the default value is used. Default arguments must be specified after non-default arguments in the function definition.

4. Combining Positional, Keyword, and Default Arguments

You can combine positional arguments, keyword arguments, and default arguments in the same function. However, there are rules:

- Positional arguments should come first.
- Default arguments should come after positional arguments.
- Keyword arguments can be used in any order after positional and default arguments.

Que. 3

There are mainly four types of scopes for variables in Python:

1. Local Scope
2. Enclosing (or Nonlocal) Scope
3. Global Scope
4. Built-in Scope

1. Local Scope

A variable defined inside a function is said to have a local scope. It is accessible only within that function.

2. Enclosing (or Nonlocal) Scope

This is also known as the lexical or nonlocal scope. It's the scope of variables in the outer function for nested functions. Variables in the enclosing scope can be accessed using the nonlocal keyword in the inner function.

3. Global Scope

A variable defined at the top level of a script or module has a global scope. It is accessible from any part of the code, including within functions (though you need to declare it as global inside functions if you want to modify it).

4. Built-in Scope

Python has a set of built-in functions and names that are always available. These are in the built-in scope.

Scope Resolution: LEGB Rule

Python uses the LEGB rule to resolve the scope of variables:

1. Local: Variables defined within the function.
2. Enclosing: Variables in the local scope of any enclosing functions.
3. Global: Variables defined at the top level of a script or module.
4. Built-in: Names preassigned in Python.

Que. 4

Python provides a variety of built-in methods for strings, lists, and other data types. These methods help perform common operations easily and efficiently. Below is a list of some commonly used methods for strings and lists.

String Methods

Here are some commonly used string methods in Python:

1. str.upper(): Converts all characters in a string to uppercase.
2. str.lower(): Converts all characters in a string to lowercase.

python

3. str.capitalize(): Capitalizes the first character of the string.
4. str.title(): Capitalizes the first character of each word.
5. str.strip(): Removes leading and trailing whitespace.

List Methods

Here are some commonly used list methods in Python:

1. list.append(x): Adds an item to the end of the list.

2. list.extend(iterable): Extends the list by appending elements from the iterable.

3. list.insert(i, x): Inserts an item at a given position.

4. list.remove(x): Removes the first occurrence of an item.