



TOPSTechnologies

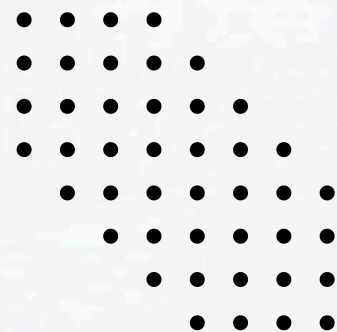
Advance Python

Presented for :

TOPs Technologies

Presented by :

Sumit B Yadav





Que 1

Functional programming is a programming paradigm in which we try to bind everything in a pure mathematical functions style. It is a declarative type of programming style. Its main focus is on "what to solve" in contrast to an imperative style where the main focus is "how to solve". It uses expressions instead of statements.

1. First-Class and Higher-Order Functions

In Python, functions are first-class citizens, which means they can be assigned to variables, passed as arguments to other functions, and returned from other functions.

2. Pure Functions

A pure function is a function that, given the same inputs, always returns the same output and has no side effects (e.g., modifying a global variable or changing the state of a program).

3. Immutability

In functional programming, data is immutable. Instead of modifying data, new data structures are created.

4. Higher-Order Functions and Lambdas

Higher-order functions can take other functions as arguments or return them as results. Lambda functions are small anonymous functions defined using the lambda keyword.

5. Functional Programming Tools

Python provides several built-in functions and modules to facilitate functional programming, such as `map()`, `filter()`, `reduce()`, and the `functools` module.

6. Recursion

Functional programming often uses recursion instead of loops for iteration. Recursion is a method where the solution to a problem depends on solutions to smaller instances of the same problem.

7. List Comprehensions

List comprehensions provide a concise way to create lists based on existing lists. They can often replace map and filter.

Que. 2

`map()` -

The `map()` function applies a given function to each item in an iterable (like a list) and returns an iterator of the results.

`filter()` -

The `filter()` function applies a given function to each item in an iterable and returns an iterator with the items for which the function returns `True`.

`reduce()` -

The `reduce()` function, which is available in the `functools` module, applies a given function to the items of an iterable, cumulatively, to reduce the iterable to a single value.

Combining `map()`, `filter()`, and `reduce()`

You can combine these functions to perform more complex data processing tasks.

Que. 3

Closures

A closure in Python is a function object that has access to variables in its lexical scope, even when the function is called outside that scope. Closures are a way to retain access to variables from the outer function even after the outer function has finished executing.

Decorators

Decorators are a powerful and useful tool in Python that allows you to modify the behavior of a function or method. They are functions that take another function as an argument, extend its behavior, and return a new function with the extended behavior.

Decorators with Arguments

Decorators can also accept arguments. To achieve this, you need to define a function that returns a decorator.

Using `functools.wraps`

When writing decorators, it's good practice to use `functools.wraps` to preserve the metadata of the original function, such as its name and docstring.