Lambda functions

03 July 2024 13:00

What is a lambda function?

A lambda function is a small, anonymous function that can be defined inline within a shorthand way to create a function without declaring a separate named function.

Syntax

The syntax for a lambda function is:

```
lambda arguments: expression
```

Where:

- arguments is a comma-separated list of variables that will be passed to the function.
- expression is the code that will be executed when the function is called.

Example

Let's say we want to create a function that takes a single argument x and returns its s with a lambda function:

```
square = lambda x: x ** 2
```

Here, x is the argument, and x ** 2 is the expression that will be executed when the fu

Using the lambda function

Now that we have our lambda function, we can use it like any other function:

```
result = square(4)
print(result)  # Output: 16
```

In this example, we pass the value 4 to the square lambda function, and it returns the is 16.

Benefits of lambda functions

Lambda functions are useful when:

- You need a small, one-time-use function.
- You want to pass a function as an argument to another function.
- You want to create a function on the fly without declaring a separate named functio

Common use cases

Lambda functions are often used in:

larger expression. It's a

square. We can do this

unction is called.

result of `4 ** 2`, which

n.

- Data processing and filtering (e.g., with filter(), map(), and reduce() functions).
- Event handling and callbacks.
- Creating small, one-time-use functions for specific tasks.