

Lambda Functions in Python

In Python, a lambda function is a small anonymous function without a name. It is defined using the lambda keyword and has the following syntax:

lambda arguments: expression

Lambda functions are often used in situations where a small function is required for a short period of time. They are commonly used as arguments to higher-order functions, such as map, filter, and reduce.

Here is an example of how to use a lambda function:

```
# Function to double the input
def double(x):
    return x * 2
```

```
# Lambda function to double the input
lambda x: x * 2
```

The above lambda function has the same functionality as the double function defined earlier. However, the lambda function is anonymous, as it does not have a name.

Lambda functions can have multiple arguments, just like regular functions. Here is an example of a lambda function with multiple arguments:

```
# Function to calculate the product of two numbers
def multiply(x, y):
    return x * y
```

```
# Lambda function to calculate the product of two numbers
lambda x, y: x * y
```

Lambda functions can also include multiple statements, but they are limited to a single expression. For example:

```
# Lambda function to calculate the product of two numbers,
# with additional print statement
lambda x, y: print(f'{x} * {y} = {x * y}')
```

In the above example, the lambda function includes a print statement, but it is still limited to a single expression.

Lambda functions are often used in conjunction with higher-order functions, such as map, filter, and reduce which we will look into later.