



Anonymous Lambda function in Python:

In Python, anonymous functions are created using the `lambda` keyword. They are also known as lambda functions or lambda expressions. Lambda functions are used when you need a small, one-line function without defining it using the `def` keyword.

Here's the basic syntax of a lambda function:

python

Copy code

```
lambda arguments: expression
```

Here's an example of how you can use a lambda function to create a simple square function:

python

Copy code

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

You can then use this `square` lambda function to calculate the square of a number:

python

Copy code

```
print(square(5)) # Output: 25
```

Lambda functions can take any number of arguments, but the expression must always result in a single value. They are often used in conjunction with functions like `map()`, `filter()`, and `reduce()`.

For example, using `map()` with a lambda function to double each element in a list:



python

Copy code

```
numbers = [1, 2, 3, 4, 5]
doubled = list(map(lambda x: x * 2, numbers))
print(doubled) # Output: [2, 4, 6, 8, 10]
```

While lambda functions are useful for simple operations, it's generally recommended to use regular functions for more complex tasks as they are easier to read and debug.

Code:

```
a = [[1, 14], [5, 6], [8, 23]]
a.sort(key=lambda x: x[1])
print(a)
```

Output:

```
C:\Users\test\PycharmProjects\project_1\.venv\Scripts\python.exe C:\Users\test\PycharmProjects\project_1\Project_1.py
[[5, 6], [1, 14], [8, 23]]
```