Lambda in Python

What is Lambda?

In Python, lambda is used to create anonymous (nameless) functions, also called lambda functions.

These are small, one-line functions defined without using the def keyword.

Syntax:

lambda arguments: expression

Example 1: Simple lambda function

```
square = lambda x: x * x
print(square(5))   # Output: 25
```

Example 2: Lambda with multiple arguments

```
add = lambda a, b: a + b
print(add(3, 4))   # Output: 7
```

Example 3: Using lambda with map(), filter(), reduce()

```
nums = [1, 2, 3, 4]
squared = list(map(lambda x: x**2, nums))
print(squared)  # [1, 4, 9, 16]


even_nums = list(filter(lambda x: x % 2 == 0, nums))
print(even_nums)  # [2, 4]


from functools import reduce
product = reduce(lambda x, y: x * y, nums)
print(product)  # 24
```

When to use lambda:

- When you need a quick, simple function for short-term use
- Useful with functions like map(), filter(), sorted(), etc.
- When the function is so simple that naming it is unnecessary

Limitations of lambda:

- Can only contain one expression (no loops, multiple statements)
- Less readable for complex logic (better to use def)

Summary:

lambda functions are lightweight, anonymous functions used when you need a quick function without formally defining it.