**Lambda Expression Syntax:**

**lambda** **arguments**: **expression**

**Key Characteristics:**

* Lambda expressions are anonymous functions (functions without names)
* They can take any number of arguments but can only have one expression
* The expression is evaluated and returned automatically
* They are also called "lambda functions" or "anonymous functions"

**Working Mechanism:**

1. **Basic Structure:**

double = lambda x: x \* 2

**2. Common Use Cases:**

**a) With map():**

numbers = [1, 2, 3, 4]

squared = list(map(lambda x: x\*\*2, numbers))

# Result: [1, 4, 9, 16]

**b) With filter():**

numbers = [1, 2, 3, 4, 5, 6]

evens = list(filter(lambda x: x % 2 == 0, numbers))

# Result: [2, 4, 6]

**c) With sorted():**

pairs = [(1, 'one'), (3, 'three'), (2, 'two')]

sorted\_pairs = sorted(pairs, key=lambda x: x[1])

# Result: [(1, 'one'), (3, 'three'), (2, 'two')]

**Advantages:**

* Concise syntax for simple operations
* Useful for short operations that won't be reused
* Excellent for functional programming operations
* Ideal for single-use functions passed as arguments

**Limitations:**

* Can only contain a single expression
* Cannot include multiple lines of code
* Limited functionality compared to regular functions
* Should be used sparingly for readability