

🎯 Topic: Lambda Functions, Map, Filter, Reduce in Python

✓ 1. What is a Lambda Function?

A **lambda** function is a **small, anonymous function** — defined without a name using the **lambda** keyword.

▀ Syntax:

```
lambda arguments: expression
```

✓ Example 1: Add two numbers

```
add = lambda a, b: a + b
print(add(5, 3)) # Output: 8
```

✓ Example 2: Square of a number

```
square = lambda x: x * x
print(square(4)) # Output: 16
```

💡 Use Lambda when:

- You want a quick one-line function
 - You don't want to reuse it many times
-

✓ 2. What is `map()`?

`map()` applies a **function to each item** in an iterable (like a list) and returns a **new list**.

☒ Syntax:

```
map(function, iterable)
```

☒ Example: Convert all strings to uppercase

```
fruits = ["apple", "banana", "mango"]
result = list(map(lambda fruit: fruit.upper(), fruits))
print(result) # ['APPLE', 'BANANA', 'MANGO']
```

☒ Example: Square all numbers

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

✓ 3. What is `filter()`?

`filter()` selects items from a list that **meet a condition** (returns True).

☒ Syntax:

```
filter(function, iterable)
```

✓ Example: Filter even numbers

```
nums = [1, 2, 3, 4, 5, 6]
even = list(filter(lambda x: x % 2 == 0, nums))
print(even) # [2, 4, 6]
```

✓ Example: Filter names with length > 5

```
names = ["Ram", "Nandini", "Gowtham", "Nila"]
long_names = list(filter(lambda name: len(name) > 5, names))
print(long_names) # ['Nandini', 'Gowtham']
```

✓ 4. What is `reduce()`?

`reduce()` reduces a list to a single value by applying a function cumulatively (comes from `functools` module).

▢ Syntax:

```
from functools import reduce
reduce(function, iterable)
```

✓ Example: Sum of all numbers

```
from functools import reduce

nums = [1, 2, 3, 4]
total = reduce(lambda a, b: a + b, nums)
print(total) # Output: 10
```

Example: Find the maximum value

```
nums = [10, 5, 22, 7]
maximum = reduce(lambda a, b: a if a > b else b, nums)
print(maximum) # Output: 22
```



Summary Table

| Concept | Purpose | Function Signature |
|-----------------|--|-------------------------------|
| lambda | One-line anonymous function | lambda x: x * 2 |
| map() | Transform each item in a list | map(function, iterable) |
| filter() () | Keep only items that meet condition | filter(function, iterable) |
| reduce() () | Reduce list to single value | reduce(function, iterable) |



Real-Life Example for Teaching:

Shopping Cart – Filter expensive items and sum them

```
from functools import reduce

prices = [250, 900, 1200, 400, 1500]
```

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

```
# Step 1: Keep only items above ₹1000
expensive = list(filter(lambda x: x > 1000, prices))

# Step 2: Calculate total
total = reduce(lambda a, b: a + b, expensive)

print("Expensive items total:", total)
```

💻 Output:

```
Expensive items total: 2700
```



BONUS: When to use what?

| Use When You Want To... | Use |
|-----------------------------------|-----------------------------|
| Create a short 1-line function | <code>lambda</code> |
| Apply a function to all items | <code>map()</code> |
| Select only matching items | <code>filter ()</code> |
| Combine values into one | <code>reduce ()</code> |

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

About the Author

Gowtham SB is a **Data Engineering expert, educator, and content creator** with a passion for **big data technologies, as well as cloud and Gen AI**. With years of experience in the field, he has worked extensively with **cloud platforms, distributed systems, and data pipelines**, helping professionals and aspiring engineers master the art of data engineering.

Beyond his technical expertise, Gowtham is a **renowned mentor and speaker**, sharing his insights through engaging content on **YouTube and LinkedIn**. He has built one of the **largest Tamil Data Engineering communities**, guiding thousands of learners to excel in their careers.

Through his deep industry knowledge and hands-on approach, Gowtham continues to **bridge the gap between learning and real-world implementation**, empowering individuals to build **scalable, high-performance data solutions**.

Socials

 **YouTube** - <https://www.youtube.com/@dataengineeringvideos>

 **Instagram** - <https://instagram.com/dataengineeringtamil>

 **Instagram** - <https://instagram.com/thedatatech.in>

 **Connect for 1:1** - <https://topmate.io/dataengineering/>

 **LinkedIn** - <https://www.linkedin.com/in/sbgowtham/>

 **Website** - <https://codewithgowtham.blogspot.com>

 **GitHub** - <http://github.com/Gowthamdataengineer>

 **WhatsApp** - <https://lnkd.in/g5JrHw8q>

 **Email** - atozknowledge.com@gmail.com

 **All My Socials** - <https://lnkd.in/gf8k3aCH>

Gowtham SB

www.linkedin.com/in/sbgowtham/

Instagram - @dataengineeringtamil

linkedin.com/in/sbgowtham/