In Python, **map**, **filter**, and **reduce** are higher-order functions that allow you to process iterable data (such as lists, tuples, or sets) efficiently and concisely.

**1️ map() – Transforming Data**

The **map()** function applies a given function to all items in an iterable and returns a new iterable (map object).

**Syntax:**

map(function, iterable)

**Example:**

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

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

print(squared\_numbers) # Output: [1, 4, 9, 16, 25]

💡 **Use case**: When you need to apply the same operation to every element in a list.

**2️⃣ filter() – Filtering Data**

The **filter()** function filters elements from an iterable based on a condition.

**Syntax:**

filter(function, iterable)

**Example:**

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

even\_numbers = list(filter(lambda x: x % 2 == 0, numbers))

print(even\_numbers) # Output: [2, 4, 6]

💡 **Use case**: When you need to remove elements that don't meet a condition.

**3️⃣ reduce() – Aggregating Data**

The **reduce()** function (from the functools module) applies a function cumulatively to items in an iterable, reducing them to a single value.

**Syntax:**

from functools import reduce

reduce(function, iterable)

**Example:**

from functools import reduce

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

sum\_numbers = reduce(lambda x, y: x + y, numbers)

print(sum\_numbers) # Output: 15 (1+2+3+4+5)

💡 **Use case**: When you need to combine all elements into a single result (like summing or multiplying values).

**🔥 Key Differences**

| **Function** | **Purpose** | **Input** | **Output** |
| --- | --- | --- | --- |
| map() | Transform each element | Function + Iterable | New iterable (transformed elements) |
| filter() | Select elements based on a condition | Function (returns True/False) + Iterable | New iterable (filtered elements) |
| reduce() | Aggregate elements into a single value | Function (with two arguments) + Iterable | Single result |

**🎯 Example Using All Three**

from functools import reduce

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

# Step 1: Square all numbers

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

# Step 2: Keep only even squares

filtered = list(filter(lambda x: x % 2 == 0, squared))

# Step 3: Sum the remaining numbers

sum\_result = reduce(lambda x, y: x + y, filtered)

print(sum\_result) # Output: 56 (4 + 16 + 36)

**🚀 When to Use?**

* **Use map()** when you want to apply a function to every item.
* **Use filter()** when you want to extract elements based on a condition.
* **Use reduce()** when you want to combine all elements into a single value.