**Lamda Functions**

Lambda functions, also known as anonymous functions, are a feature in Python .It contain number of parameter are passsed in single line.

**Example1:**

# Regular function to square a number

def square(x):

return x \*\* 2

# Equivalent lambda function

lambda\_square = lambda x: x \*\* 2

# Using both functions

print(square(5))

print(lambda\_square(5))

**Example2:**

# Regular function to add two numbers

def add(x, y):

return x + y

# Equivalent lambda function

lambda\_add = lambda x, y: x + y

# Using both functions

print(add(3, 4))

print(lambda\_add(3, 4))

**example3**

# Using lambda directly

result = (lambda x: x \*\* 2)(5)

print(result) # Output: 25

**Example:4(list out the even number)**

# List of numbers

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Lambda function to filter even numbers

filter\_even = lambda x: x % 2 == 0

# Use the lambda function to filter even numbers from the list

even\_numbers = list(filter(filter\_even, numbers))

# Print the result

print("Original numbers:", numbers)

print("Even numbers:", even\_numbers)

**Filter Function**

The filter() function in Python is a built-in function that is used to filter elements of an iterable (such as a list) based on a specified function or condition. It returns an iterator containing the elements for which the function returns True.

**Syntax:**

**filter(function, iterable)**

**function**: A function that tests whether each element of an iterable will be included in the result. If None, it simply returns the elements of the iterable that are true.

**iterable:** An iterable (e.g., a list, tuple, etc.) whose elements are tested by the function.

**The filter()** function applies the function to each element of the iterable and returns only the elements for which the function returns True.

**Example:1**

**Example:4(list out the even number)**

# List of numbers

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

# Lambda function to filter even numbers

filter\_even = lambda x: x % 2 == 0

# Use the lambda function to filter even numbers from the list

even\_numbers = list(filter(filter\_even, numbers))

# Print the result

print("Original numbers:", numbers)

print("Even numbers:", even\_numbers)

**Example:2**

# Define a function to filter words longer than 3 characters

def filter\_long\_words(word):

return len(word) > 3

# List of words

words = ["apple", "banana", "kiwi", "orange", "grape"]

# Use filter() to get words longer than 3 characters

filtered\_words = list(filter(filter\_long\_words, words))

print("Original words:", words)

print("Words longer than 3 characters:", filtered\_words)

Map()

The map() function in Python is a built-in function that applies a specified function to all the items in an iterable (e.g., a list, tuple, etc.) and returns an iterator that produces the results.

**Syntax:**

**map(function, iterable, ...)**

**Example:1**

# Define a function that squares a number

def square(x):

return x \*\* 2

# List of numbers

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

# Use map() to apply the square function to each element in the list

squared\_numbers = map(square, numbers)

# Convert the iterator to a list for printing

result\_list = list(squared\_numbers)

print("Original numbers:", numbers)

print("Squared numbers:", result\_list)

**Example:2**

# Define a function to double a number

def double(x):

return x \* 2

# List of numbers

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

# Use map() to apply the double function to each element in the list

doubled\_numbers = map(double, numbers)

# Convert the iterator to a list for printing

result\_list = list(doubled\_numbers)

print("Original numbers:", numbers)

print("Doubled numbers:", result\_list)

**Zip()**

The **zip()** function in Python is a built-in function that takes two or more iterables (e.g., lists, tuples) and aggregates their elements into tuples. It returns an iterator of tuples where the i-th tuple contains the i-th element from each of the input iterables.

**Example:1**

# Two lists

names = ["John", "Alice", "Bob"]

ages = [25, 30, 22]

# Use zip() to pair elements from the two lists

zipped\_data = zip(names, ages)

# Convert the iterator to a list for printing

result\_list = list(zipped\_data)

print("Names:", names)

print("Ages:", ages)

print("Zipped data:", result\_list)