Python

Full stack Skills Bootcamp

Introducing Python Lambdas

- What are Functions?
  - Lambda functions are small, anonymous functions defined using the lambda keyword. They are designed for situations where a simple function is needed for a short duration.
- Characteristics:
  - Number of Arguments: They can take any number of arguments (including none).
  - Single Expression: They can only contain a single expression, which makes them concise and easy to use for simple operations.

Defining a Basic Lambda

- Creating a Lambda Function:

```python
square = lambda x: x ** 2
print("Square of 5:", square(5))  # Output: Square of 5: 25
```

- In this example, we define a lambda function that calculates the square of a number x.
- The function is assigned to the variable “square”, which can then be called like a regular function.
- Output: When we call square(5), it computes 5^2 and returns 25.

```python
def a(x, y):
    return x + y

b = lambda x, y: x + y
```

Lambda with map( )

```
numbers = [1, 2, 3, 4]
squares = list(map(lambda x: x ** 2, numbers))
print("Squares of numbers:", squares)  # Output: Squares of numbers: [1, 4, 9, 16]
```

- The map() function applies the provided lambda function to each element in the list numbers.
- Here, the lambda function takes each number x and returns its square.
- The result of map() is an iterable, which is converted to a list using the list() function.
- Output: The list of squares, [1, 4, 9, 16], is produced by mapping the square function over the original list.

Lambda with filter( )

```python
evens = list(filter(lambda x: x % 2 == 0, numbers))
print("Even numbers:", evens)  # Output: Even numbers: [2, 4]
```

- The filter() constructs an iterator from elements of the iterable numbers for which the lambda function returns true.

- In this case, the lambda checks if each number x is even (i.e.., x % 2 == 0).
- The filtered result is converted to a list.
- Output: The even numbers extracted from the list are [2, 4], demonstrating how filtering works with lambda functions.

Lambda with sorted( )

```python
tuple_list = [(4, 'pineapple'), (2, 'banana'), (3, 'cherry')]
sorted_list = sorted(tuple_list, key=lambda x: x[1])
print("Sorted list by second element:", sorted_list)  # Output:
```

- The sorted() function sorts the list of tuples based on the second element of each tuple.
- The lambda function is used as the sorting key, taking each tuple x and returning x[1], which is a fruit name.
- Output: The sorted list, [(2, 'banana'), (3, 'cherry'), (4, 'pineapple')], reflects the alphabetical order of the second elements.

Conclusion

- Key Points:

  - Conciseness: Lambda functions allow you to write shorter code for simple functions, reducing boilerplate.
  - Higher-Order Functions: They are commonly used with functions like map(), filter(), and sorted(), enabling functional programming paradigms.
  - Use with Care: While lambda functions are powerful, they can reduce code readability if overused or if the expression is too complex. For maintainability, consider using named functions for more complicated logic.