Day 9 – LAMBDA

Lambda

● A lambda function is a small anonymous function & can take any number of arguments, but only have one expression

● Python Lambda function is known as the anonymous function that is defined without a name. Python allows us to not declare the function in the standard manner, i.e., by using the def keyword. Rather, the anonymous functions are declared by using the lambda keyword

```
lambda arguments : expression
```

Add 10 to argument a, and return the result:

In [1]:

```
x = lambda m : m + 10
print(x(5))
```

```
15
```

● A bound variable is an argument to a lambda function In contrast, a free variable is not bound and may be referenced in the body of the expression. A free variable can be a constant or a variable defined in the enclosing scope of a function.

● Lambda functions can take any number of arguments

● The power of lambda is better shown when you use them as an anonymous function inside another function, when you have a function definition that takes one argument, and that argument will be multiplied with an unknown number.

In [2]:

```
def myfunc(n):
    return lambda a : a * n

mydoubler = myfunc(2)

print(mydoubler(11))
```

```
22
```

Use lambda functions when an anonymous function is required for a short period of time.

● The Python built-in filter() function accepts a function and a list as an argument. It provides an effective way to filter out all elements of the sequence. It returns the new sequence in which the function evaluates to True.

```
tuple(filter(lambda x:(x % 3 == 0),lst))
```

● The map() function in Python accepts a function and a list. It gives a new list which contains all modified items returned by the function for each item.

```
list(map(lambda x : x ** 2, lst))
```

Exercises

- **Create a lambda function that multiplies argument x with argument y**

In [3]:

```python
a = lambda x, y : x * y
print(a(12, 10))
```

```
120
```

- **Write a Python program to create Fibonacci series to n using Lambda**

In [5]:

```python
def fibonacci(count):
    listA = [0, 1]
    any(map(lambda _:listA.append(sum(listA[-2:])),
         range(2, count)))
    return listA[:count]
print(fibonacci(int(input())))
```

```
20
[0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584,
4181]
```

- **Write a Python program that multiply each number of given list with a given number**

In [7]:

```python
num = [1, 4, 9, 16, 25]
x = 5
print("Given List: ", num)
print("\nGiven number: ", x)
nums_updated = list(map(lambda y:y*x, num))
print("\nResult:", nums_updated)
```

```
Given List:  [1, 4, 9, 16, 25]

Given number:  5

Result: [5, 20, 45, 80, 125]
```

- **Write a Python program to find numbers divisible by 9 from a list of numbers**

In [8]:

```python
nums = [1, 4, 9, 16, 25, 36]
result = list(filter(lambda x: (x % 9 == 0), nums))
print("Numbers divisible by 9 are",result)
```

```
Numbers divisible by 9 are [9, 36]
```

- **Write a Python program to count the even numbers in a given list of integers**

In [10]:

```
nums = [1, 4, 9, 16, 25, 36]
result = list(filter(lambda x: (x % 2 == 0), nums))
print("Even numbers in the list are",result)
print("count is",len(result))
```

```
Even numbers in the list are [4, 16, 36]
count is 3
```

Completed Day 9's notes & exercises

THANK YOU!

Check out My Repository at https://github.com/AakankshaJarode/BestEnlist_Python_Internship.git (https://github.com/AakankshaJarode/BestEnlist_Python_Internship.git)

Chech out My LinkedIn Page at https://www.linkedin.com/in/aakanksha-jarode-1b0195179 (https://www.linkedin.com/in/aakanksha-jarode-1b0195179)