

Ques. What is Lambda/Anonymous Function?

- A lambda function is a small anonymous function.
- In Python, an anonymous function is a function that is defined without a name.
- While normal functions are defined using the **def** keyword in Python, anonymous functions are defined using the **lambda** keyword.
- **Notice** that the anonymous function does not have a return keyword. This is because the anonymous function will automatically return the result of the expression in the function once it is executed.

```
def add(a,b):  
    print(a+b)  
add(5,10)  
  
# Using Lambda function  
x = lambda a: a + 10  
print(x(5))
```

Output:- 15

- You can use lambda function in **filter()**

```
# filter() function is used to filter a given iterable (list like object) using  
another function that defines the filtering logic.  
# Syntax:- filter(object, iterable)  
# The object here should be a lambda function which returns a boolean value.  
mylist = [2,3,4,5,6,7,8,9,10]  
list_new = list(filter(lambda x : (x%2==0), mylist))  
print(list_new)
```

Output:- [2, 4, 6, 8, 10]

- We can use lambda function in **map()**

```
# map() function applies a given function to all the itmes in a list and returns  
the result. Similar to filter(), simply pass the lambda function and the list (or  
any iterable, like tuple) as arguments.
```

```
mylist = [2,3,4,5,6,7,8,9,10]  
list_new = list(map(lambda x : x%2, mylist))  
print(list_new)
```

Output:- [0, 1, 0, 1, 0, 1, 0, 1, 0]

- You can use lambda function in **reduce()** as well

reduce() function performs a repetitive operation over the pairs of the elements in the list. Pass the lambda function and the list as arguments to the reduce() function. For using the reduce() function, you need to import reduce from functools library.

```
from functools import reduce
list1 = [1,2,3,4,5,6,7,8,9]
sum = reduce((lambda x,y: x+y), list1)
print(sum)
```

Output:- 45 //i.e 1+2, 1+2+3 , 1+2+3+4 and so on.

How to use lambda function to manipulate a Dataframe
 # You can also manipulate the columns of the dataframe using the lambda function. It's a great candidate to use inside the apply method of a dataframe. I will be trying to add a new row in the dataframe in this section as example.

```
import pandas as pd
df = pd.DataFrame([[1,2,3],[4,5,6]],columns = ['First','Second','Third'])
df['Forth']= df.apply(lambda row: row['First']*row['Second']* row['Third'],
axis=1)
df
```

Output:-

	First	Second	Third	Forth
0	1	2	3	6
1	4	5	6	120