**Python Anonymous/Lambda Function**

In Python, anonymous function is a [function](https://www.programiz.com/python-programming/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. Hence, anonymous functions are also called lambda functions.

Syntax of Lambda Function in python

**lambda arguments: expression**

Lambda functions can have any number of arguments but only one expression. The expression is evaluated and returned. Lambda functions can be used wherever function objects are required.

# Program to show the use of lambda functions

double = lambda x: x \* 2

print(double(5))

In the above program, lambda x: x \* 2 is the lambda function. Here x is the argument and x \* 2 is the expression that gets evaluated and returned.

This function has no name. It returns a function object which is assigned to the identifier double. We can now call it as a normal function. The statement

double = lambda x: x \* 2

is nearly the same as

def double(x):

return x \* 2

## Why Use Lambda inside Functions?

The power of lambda is better shown when you use them as an anonymous function inside another function.

Say you have a function definition that takes one argument, and that argument will be multiplied with an unknown number:

def myfunc(n):

return lambda a : a \* n

mydoubler = myfunc(2)

tripler = myfunc(3)

print(mydoubler(11))

print(tripler(11))

### Use of Lambda Function in python

We use lambda functions when we require a nameless function for a short period of time.

In Python, we generally use it as an argument to a higher-order function (a function that takes in other functions as [arguments](https://www.programiz.com/python-programming/function-argument)). Lambda functions are used along with built-in functions like filter(), map() etc.

# Lambda Filter

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

new\_list = list((lambda x: (x%2 != 0 ), li ))

print(new\_list)

#---------------------------------------------------------

# Lambda Map

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

new\_list = list(map(lambda x: x\*2, li ))

print(new\_list)

#---------------------------------------------------------

# Lambda Reduce

from functools import reduce

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

sum = reduce((lambda x,y : x+y), li)

print(sum)