

```
# -*- coding: utf-8 -*-
"""
```

Python Modules

Summary By: Varpe K.M.

Reference: <https://www.javatpoint.com/python-lambda-functions>

```
"""
```

```
"""
```

Python Lambda Functions

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 lambda keyword.

However, Lambda functions can accept any number of arguments, but they can return only one value in the form of expression.

The anonymous function contains a small piece of code.

It simulates inline functions of C and C++,

but it is not exactly an inline function.

The syntax to define an Anonymous function is given below.

```
lambda arguments : expression
```

```
"""
```

#Example 1

```
x = lambda a:a+10 # a is an argument and a+10 is an expression which got evaluated and returned
print("sum = ",x(20))
```

```
"""
```

Why use lambda functions?

The main role of the lambda function is better described in the scenarios when we use them anonymously inside another function.

In python, the lambda function can be used as an argument to the higher order functions as arguments.

Lambda functions are also used in the scenario where we need a Consider the following example.

Example 1

```
"""
```

#the function table(n) prints the table of n

```
def table(n):
```

```
    return lambda a:a*n; # a will contain the iteration variable i and a multiple of n is returned
```

```
n = int(input("Enter the number?"))
```

```
b = table(n) #the entered number is passed into the function table. b will contain a lambda function
```

```
for i in range(1,11):
```

```
    print(n,"X",i,"=",b(i)); #the Lambda function b is called with the iteration variable i,
```

```
"""
```

Output:

Enter the number?10

10 X 1 = 10

10 X 2 = 20

10 X 3 = 30

10 X 4 = 40

10 X 5 = 50

10 X 6 = 60

10 X 7 = 70

10 X 8 = 80

10 X 9 = 90

10 X 10 = 100

```
"""
```

```
def make_adder(n):
```

```
    return lambda x: x + n

plus_3 = make_adder(3)
print(plus_3)
plus_5 = make_adder(5)
print(plus_5)
plus_3(4)
plus_5(4)
```