

Anonymous function or Lambdas

=====

A function without a name is called 'anonymous function'. So far, the functions we wrote were defined using the keyword 'def'. But anonymous functions are not defined using 'def'. They are defined using the keyword 'lambda' and hence they are also called 'Lambda functions'.

```
def square(x):  
    return x*x
```

```
square(5)
```

o/p: 25

The same function can be written as

```
lambda x : x * x
```

1. Python program that create a lambda function that return a square value of a given number

```
a = lambda x: x*x  
value = a(5)  
print("Square of 5 is:", value)  
o/p:
```

Square of 5 is: 25

Lambda function contain only one expression and they return the result implicitly. Hence we should not write any return statement in lambda functions.

2. Python lambda function calculating sum of two numbers.

```
b = lambda x,y : x+y  
value1 = b(10,20)  
print("The sum is:", value1)  
o/p:
```

The sum is: 30

lambdas with filter() function :

The filter() is useful to filter out the elements of sequence depending on the result of a function. We should supply a sequence to filter function as:

filter(function, sequence)

1. A Python program using filter() to filter out even numbers from a list

```
def is_even(n):  
    if n%2 == 0:  
        return True  
    else:  
        return False
```

```
lst = [1,2,3,4,5,6]  
lst1 = list(filter(is_even, lst))  
print(lst1)
```

2. lambda function that returns even numbers

```
lst = [1,2,3,4,5,6]  
lst1 = list(filter(lambda x: (x%2 == 0), lst))  
print(lst1)
```

lambdas with map() function :

The map() function is similar to filter() function, but it acts on each element of the sequence and perhaps changes the elements. The format of map() function is:

map(function, sequence)

The 'function' performs a specified operation on all the elements of the sequence and the modified elements are returned which can be stored in another sequence.

1. Python map function to find square of each element of sequence

```
def squares(x):
```

```
    return x*x
```

```
lst = [1,2,3,4,5,6]
```

```
lst1 = list(map(squares, lst))
```

```
print(lst1)
```

```
o/p: [1, 4, 9, 16, 25, 36]
```

```
lst = [1,2,3,4,5,6]
```

```
lst1 = list(map(lambda x: x*x, lst))
```

```
print(lst1)
```

```
o/p: [1, 4, 9, 16, 25, 36]
```

```
a = lambda x: x*x
```

```
lst = [1,2,3,4,5,6]
```

```
for i in lst:
```

```
    print(a(i))
```

```
o/p:
```

```
1
```

```
4
```

```
9
```

```
16
```

```
25
```

```
36
```