**Lambda Operator**

1. The lambda operator or lambda function is a way to create small anonymous functions, i.e. functions without a name.
2. they are just needed where they have been created.
3. Lambda functions are mainly used in combination with the functions filter(), map() and reduce().
4. he general syntax of a lambda function is quite simple:  
   lambda argument\_list: expression
5. >>> f = lambda x, y : x + y

>>> f(1,1)

2

### The map() Function

1. *map()* applies the function *func* to all the elements of the sequence *seq*. It returns a new list with the elements changed by *func*
2. r = map(func, seq)
3. The first argument *func* is the name of a function and the second a sequence (e.g. a list) *seq*.
4. Ex:

>>> Celsius = [39.2, 36.5, 37.3, 37.8]

>>> Fahrenheit = map(lambda x: (float(9)/5)\*x + 32, Celsius)

>>> print Fahrenheit

[102.56, 97.700000000000003, 99.140000000000001, 100.03999999999999]

>>> C = map(lambda x: (float(5)/9)\*(x-32), Fahrenheit)

>>> print C

[39.200000000000003, 36.5, 37.300000000000004, 37.799999999999997]

5. map() can be applied to more than one list. The lists have to have the same length.

6. >>> a = [1,2,3,4]

>>> b = [17,12,11,10]

>>> c = [-1,-4,5,9]

>>> map(lambda x,y:x+y, a,b)

[18, 14, 14, 14]

>>> map(lambda x,y,z:x+y+z, a,b,c)

[17, 10, 19, 23]

>>> map(lambda x,y,z:x+y-z, a,b,c)

[19, 18, 9, 5]

### Filtering