**Lambda Functions**

* The lambda operator or lambda function is a way to create small anonymous functions, i.e. functions without a name. These functions are throw-away functions, i.e. they are just needed where they have been created.
* Lambda functions are mainly used in combination with the functions filter(), map() and reduce().
  + **Eg:**square=(lambda a:a\*\*2)
  + Square(2)
  + **Eg2:**
  + Add\_two\_num=(lambda a,b:a+b)
  + Add\_two\_num(10,20)
  + Lambda functions are autoreturned functions.
* **Map():**
  + The advantage of the lambda operator can be seen when it is used in combination with the map() function.
  + map() is a function with two arguments:
    - r=map(function,sequence)
    - example:
    - **Eg:**
    - L1=[1,2,3]
    - L2=list(map(square,l1))
    - **Eg2:**
    - L3=list(map(lambda i:i+10,l1))
* **Filter():**
  + Filter can be used to remove unwanted data.
  + It reuires if else.
  + **Syntax:**
    - Filter(function\_name,list)
    - **Eg:**
    - L4=list(filter(lambda x:x%2==0,l1))
* **Reduce():**
  + The function reduce(function,sequence) continously applies the function to the sequence.it returns a single value.
  + **Syntax:**
    - From functools import reduce
    - Reduce(function,list)
    - **Eg:**
    - List=[12,14,16,18]
    - From functools import reduce
    - Reduce(lambda x,y:x+y,list)