- Lambda Functions is also a kind of functions representation
- more efficient way

In [3]:
```
def summ(num):
    return(num+10)
summ(10)
#function name is summ the variable name is num, return output is num+10
```

Out[3]: 20

In [ ]:
```
<function_name>=lambda <variable_name>:<return output>
```

In [6]:
```
summ =lambda num:num+10
summ(20)
```

Out[6]: 30

In [7]:
```
def cube(num):
    return(num*num*num)
cube(10)
```

Out[7]: 1000

In [8]:
```
cube= lambda num : num*num*num
cube(10)
```

Out[8]: 1000

- Two arguments

In [11]:
```
def add(n1,n2):
    return(n1+n2)
add(100,200)
```

Out[11]: 300

In [ ]:
```
<function_name>=lambda <variable_name1>, <variable_name1>:<return output>
```

In [13]:
```
add=lambda n1,n2 :n1+n2
add(100,200)
```

Out[13]: 300

In [16]:
```
def avg(n1,n2,n3):
    return((n1+n2+n3)/3)
avg(100,200,300)
```

Out[16]: 200.0

In [18]:
```
avg=lambda n1,n2,n3 :(n1+n2+n3)/3
avg(100,200,300)
```

Out[18]: 200.0

if-else

In [21]:
```
def Max(n1,n2):

    if n1>n2:
        return(n1)
    else:
        return(n2)
Max(100,200)
```

Out[21]: 200

In [ ]:
```
lambda <v1,v2,v3,v4.....>:<output>

if - else we need to write in a single

[<if_output><if_condition>else <else_output>]

<function_name>=lambda <var1><var2>:<if_out><if_con> else <else_output>
```

In [22]:
```
Max =lambda n1,n2: n1 if n1>n2 else n2
Max(100,200)
```

Out[22]: 200

In [23]:
```
list1=['hyd','mumbai','chennai']
#list2=['Hyd','Mumbai','Chennai']
################################################
list2=[]
for i in list1:
    list2.append(i.capitalize())
################################################
list2=[i.capitalize() for i in list1]
list2
################################################
list1=['hyd','mumbai','chennai']
list(map(lambda i:i.capitalize(),list1))
```

Out[23]: ['Hyd', 'Mumbai', 'Chennai']

map

In [ ]:
```
map(lambda <varible>:<output>,<iterator>))
```

In [ ]:
```
for lists lambda function is going to be like this

lambda <variable>:<output>,<iterator>
```

In [26]:
```
list1=['hyd','mumbai','chennai']
list2=list(map(lambda i:i.capitalize(),list1)) #list map
```

In [29]:
```
l1=[1,2,3,4]
# l2=[1,4,9,16]
l2=list(map(lambda i:i*i,l1))
l2
```

Out[29]: [1, 4, 9, 16]

In [30]:
```
l1=['hydera#bad','Mum#bai','Chenna#i','Blr','pune']
#l2=['hydera#bad','Mum#bai','Chenna#i']
l2=[]
for i in l1:
    if '#' in i:
        l2.append(i)
l2
l2=[i for i in l1 if '#' in i]
```

Out[30]: ['hydera#bad', 'Mum#bai', 'Chenna#i']

In [31]:
```
l2=[i for i in l1 if '#' in i]
l2
```

Out[31]: ['hydera#bad', 'Mum#bai', 'Chenna#i']

filter

In [ ]:
```
map means map the input values to output
                  i                   i*i
                  i                   i.capitalize()
                  i                   '#' in i (T/F)
```

In [ ]:
```
map will give just true or false, or if there is direct formula or direct
while filter will give the desired output
```

In [ ]:
```
l2=[i for i in l1 if '#' in i]
```