

lambda

In [2]:

```
def funname():print("hello")print("hai")
funname()
```

```
File "<ipython-input-2-f9f4ec5bf4b2>", line 1
    def funname():print("hello")print("hai")
                                   ^
```

SyntaxError: invalid syntax

- lambda is a anonymous function
- this function doesn't contain any name
- this can be defined in a single line
 - syntax:
 - lambda args:expression

In [3]:

```
m=5
m*2
```

Out[3]:

10

In [4]:

```
def double(num):
    return num*2
```

In [5]:

```
double(5)
```

Out[5]:

10

In [8]:

```
double1=lambda a:a*2
```

In [9]:

```
double1(11)
```

Out[9]:

22

In [10]:

```
power=lambda a,n:a**n
```

In [11]:

```
power(10,2)
```

Out[11]:

100

map

- it is used to apply a single function across the iterable
- it returns the result
- no.of iterable==no.of arguments in the function
 - syntax:
 - map(function,*iterable)

filter

- it is used to apply a single function across the iterable
- it returns true or false
- it can be only applicable on a single iterator
- the function must return true or false
 - syntax:
 - filter(function,iterable)

In [12]:

```
l=[1,2,3]  
l=list(map(double,l))
```

In [13]:

```
l
```

Out[13]:

[2, 4, 6]

In [14]:

```
1
```

Out[14]:

```
[2, 4, 6]
```

In [17]:

```
l1=list(map(lambda x:x*2,l))
```

In [18]:

```
l1
```

Out[18]:

```
[8, 16, 24]
```

In [19]:

```
1
```

Out[19]:

```
[4, 8, 12]
```

In [22]:

```
def addlist(l,l1):  
    l2=[]  
    for i in range(len(l)):  
        l2.append(l[i]+l1[i])  
    return l2
```

In [23]:

```
addlist(l,l1)
```

Out[23]:

```
[12, 24, 36]
```

In [25]:

```
addlist([1,2,3],[1,2,3,4])
```

Out[25]:

```
[2, 4, 6]
```

In [29]:

```
l=[1,2,23]
l1=[1,25,36]
l2=list(map(lambda a,b:a+b,l,l1))
print(l2)
```

[2, 27, 59]

In [32]:

```
l3=list(map(lambda x,a:a**x,l1,l))
'''
l1=[1,25,36]-->x
l=[1,2,23]-->a
1st step:
x=1,a=1-->1**1-->1
2nd step:
x=25,a=2-->2**25-->33554432
'''
l3
```

Out[32]:

[1, 33554432, 10524515126174167358877236351104092889324551536161]

In [35]:

```
l=[1,2,3]
l1=[2,3,4]
l2=map(lambda a,x: a**x,l1,l)
'''
l=[1,2,3]-->x
l1=[2,3,4]-->a
1st:
x=1,a=2-->a**x-->2**1-->2
2nd
x=2,a=3-->a**x-->3**2-->9
3rd
x=3,a=4-->a**x-->4**3-->64
'''
print(list(l2))
```

[2, 9, 64]

In [41]:

```
smarks=[75,35,32,18,45,69,58,42,90,100,25,47,83]
psmarks=list(filter(lambda x:x>=35,smarks))
```

In [39]:

```
psmarks
```

Out[39]:

```
[75, 35, 45, 69, 58, 42, 90, 100, 47, 83]
```

In [42]:

```
fsmarks=list(filter(lambda x:x<35,smarks))
```

In [43]:

```
fsmarks
```

Out[43]:

```
[32, 18, 25]
```

In [46]:

```
# 66 1000 15 65  
l=list(map(lambda x:int(x),input().split()))  
sum(l)
```

```
66 1000 15 65
```

Out[46]:

```
1146
```

In [47]:

```
l=input().split()
```

```
66 1000 15 65
```

In [48]:

```
l
```

Out[48]:

```
['66', '1000', '15', '65']
```

In [49]:

```
sum(1)
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-49-b957b889506a> in <module>  
----> 1 sum(1)
```

TypeError: unsupported operand type(s) for +: 'int' and 'str'

In [50]:

```
l=list(map(lambda x:int(x),l))
```

In [51]:

```
l
```

Out[51]:

```
[66, 1000, 15, 65]
```

In [52]:

```
sum(l)
```

Out[52]:

```
1146
```

In [53]:

```
l=[1,2,3,4,5,6,7,8,9,10]  
l1=[i for i in range(1,11)]
```

In [54]:

```
l1
```

Out[54]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

In [57]:

```
l2=[i for i in range(1,100) if i%2==0]
```

In [58]:

```
l2
```

...

In [59]:

```
l3=[i for i in range(25,50) if i%3==0]
```

In [60]:

```
l3
```

Out[60]:

```
[27, 30, 33, 36, 39, 42, 45, 48]
```

In [61]:

```
l=[]  
for i in range(1,101):  
    if i%2!=0:  
        l.append(i)  
print(l)
```

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39,  
41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65, 67, 69, 71, 73, 75, 77,  
79, 81, 83, 85, 87, 89, 91, 93, 95, 97, 99]
```

In [62]:

```
l3=[x for x in range(1,101) if x%2!=0]  
l3
```

Out[62]:

```
[1,  
3,  
5,  
7,  
9,  
11,  
13,  
15,  
17,  
19,  
21,  
23,  
25,  
27,  
29,  
31,  
33,  
35,  
37,  
39,  
41,  
43,  
45,  
47,  
49,  
51,  
53,  
55,  
57,  
59,  
61,  
63,  
65,  
67,  
69,  
71,  
73,  
75,  
77,  
79,  
81,  
83,  
85,  
87,  
89,  
91,  
93,  
95,  
97,  
99]
```


In []: