INTERMEDIATE PYTHON

LESSON 5 |**Lambda, map,filter & redue**| 26-12-18

Video tutorials **:**

Lambda: ***https://www.youtube.com/watch?v=iMOIGsNHQjM***

Map, Filter & Reduce: ***https://www.youtube.com/watch?v=8roRLynIYMM***

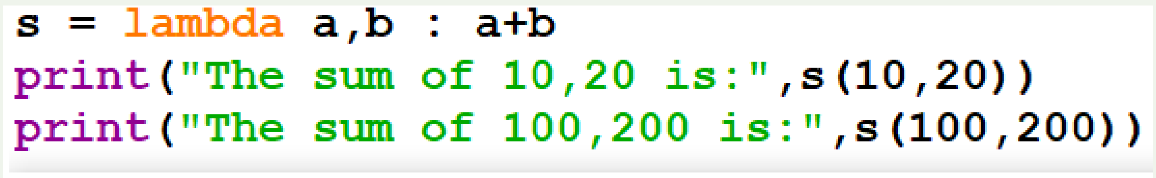
Lambda function

lambda operator or lambda function is used for creating small, one-time and anonymous function objects in Python.

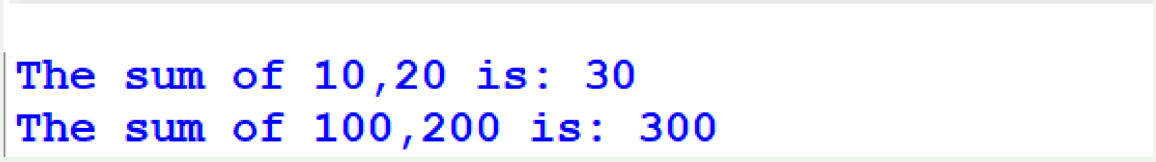
```
lambda arguments : expression
```

Lambda operator can have any number of arguments, but it can have only one expression. It cannot contain any statements and it returns a function object which can be assigned to any variable.

Example 1: Lambda function to find sum of two given numbers:

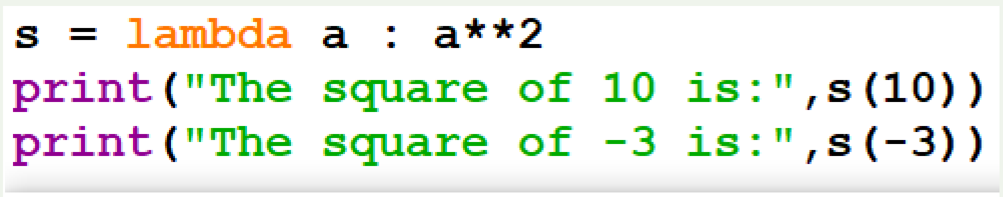


```python
s = lambda a,b : a+b
print("The sum of 10,20 is:",s(10,20))
print("The sum of 100,200 is:",s(100,200))
```

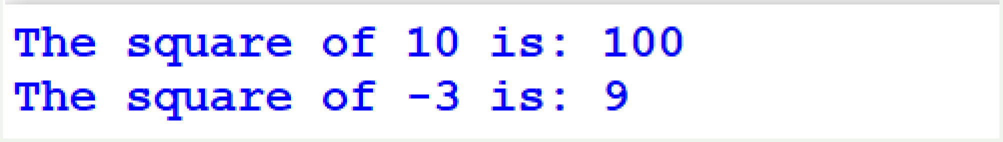


```
The sum of 10,20 is: 30
The sum of 100,200 is: 300
```

Example 2: Lambda function to find the square of a given number:

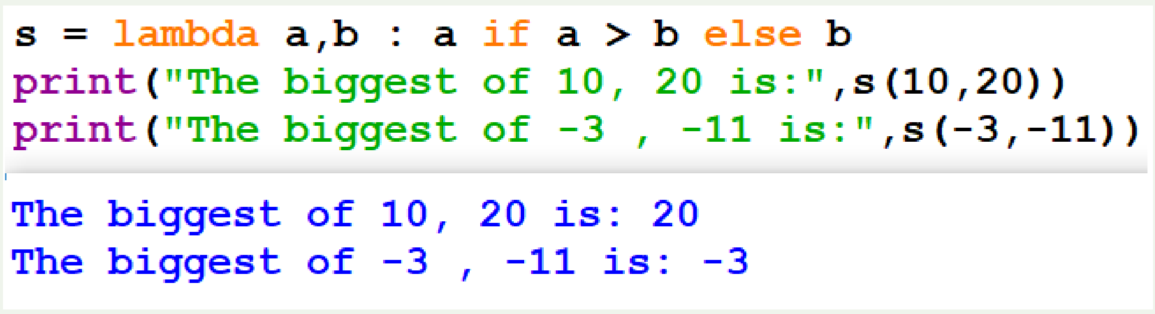


```python
s = lambda a : a**2
print("The square of 10 is:",s(10))
print("The square of -3 is:",s(-3))
```



```
The square of 10 is: 100
The square of -3 is: 9
```

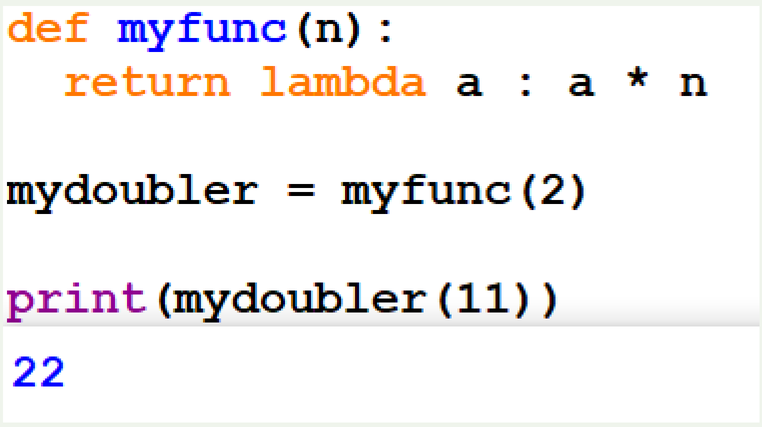
Example 3: Lambda function to find biggest of given values



```
s = lambda a,b : a if a > b else b
print("The biggest of 10, 20 is:",s(10,20))
print("The biggest of -3 , -11 is:",s(-3,-11))
```

```
The biggest of 10, 20 is: 20
The biggest of -3 , -11 is: -3
```

Example 4: Lambda function inside another function



```
def myfunc(n):
  return lambda a : a * n

mydoubler = myfunc(2)

print(mydoubler(11))
```

```
22
```

Note 1: Lambda Function internally returns expression value and we are not required to write return statement explicitly.

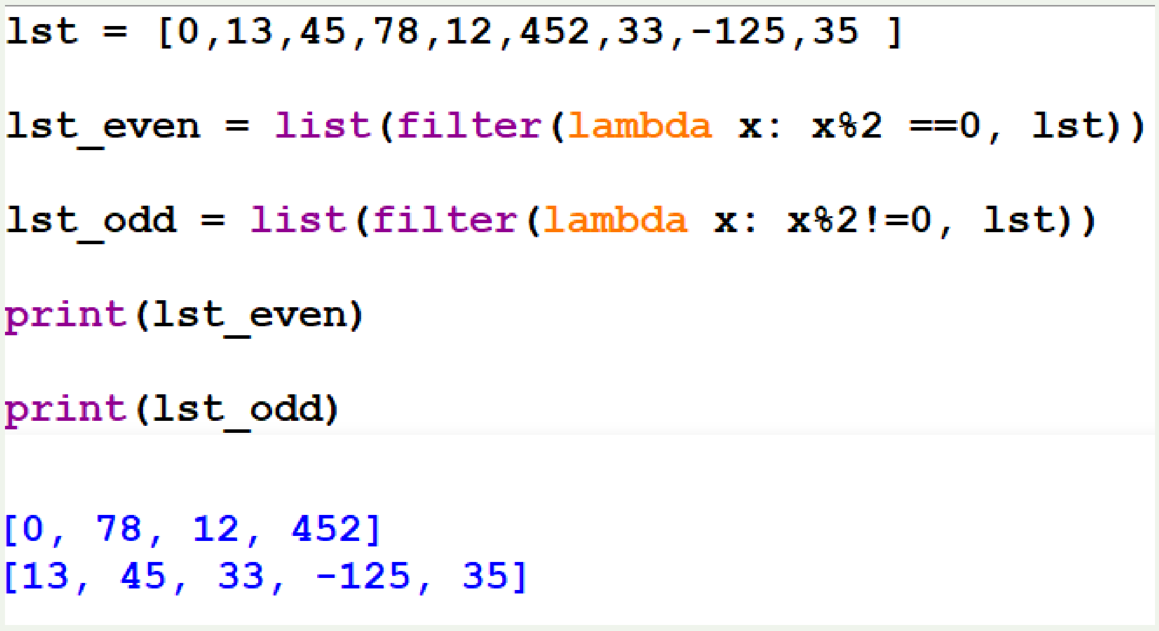
Note 2: Sometimes we can pass function as argument to another function. In such cases lambda functions are best choice. We can use lambda functions very commonly with filter(), map() and reduce()functions, These functions expect function as argument.

Filter () function

We can use filter() function to filter values from the given sequence based on some condition.

fiter (function, sequence)

Where function argument is responsible to perform conditional check . Sequences can be list, tuple or string.



```
lst = [0,13,45,78,12,452,33,-125,35 ]

lst_even = list(filter(lambda x: x%2 ==0, lst))

lst_odd = list(filter(lambda x: x%2!=0, lst))

print(lst_even)

print(lst_odd)
```

```
[0, 78, 12, 452]
[13, 45, 33, -125, 35]
```

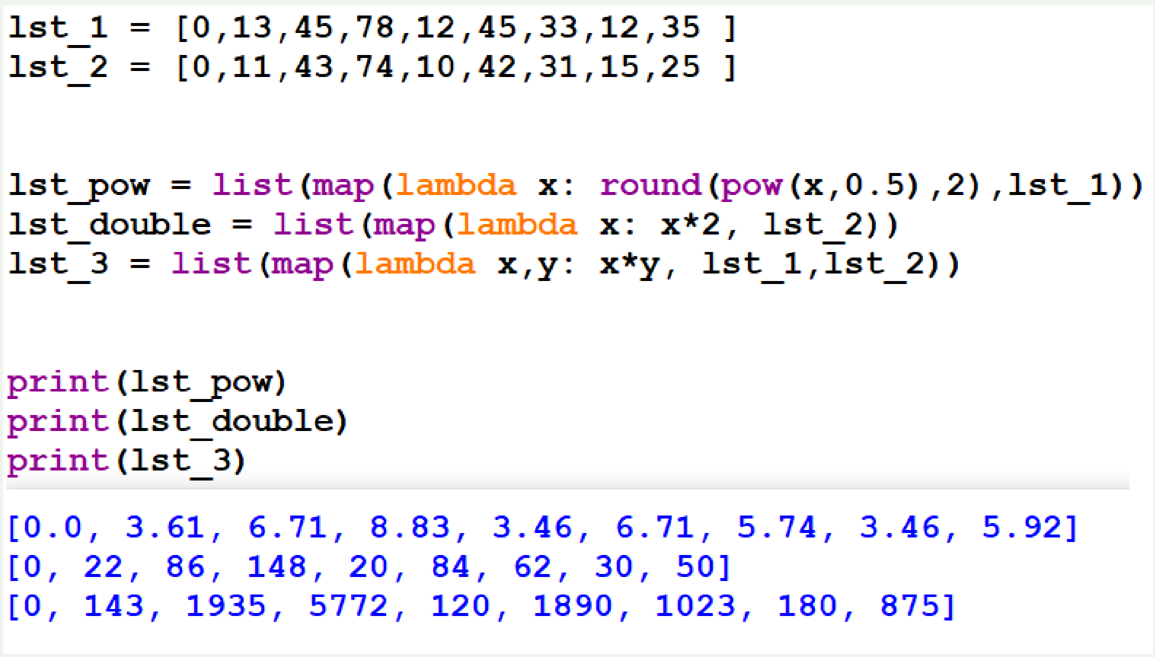
map () function

For every element present in the given sequence , apply some functionality and generate new element with the required modification. For this requirement we should go for map() function.

map (function, sequence)

The function can be applied on each element and generates new sequence. We can apply map () function on multiple lists also. But make sure all list have same length,

map (lambda x,y: x*y, lst_1,lst_2)



```
lst_1 = [0,13,45,78,12,45,33,12,35 ]
lst_2 = [0,11,43,74,10,42,31,15,25 ]


lst_pow = list(map(lambda x: round(pow(x,0.5),2),lst_1))
lst_double = list(map(lambda x: x*2, lst_2))
lst_3 = list(map(lambda x,y: x*y, lst_1,lst_2))


print(lst_pow)
print(lst_double)
print(lst_3)
```

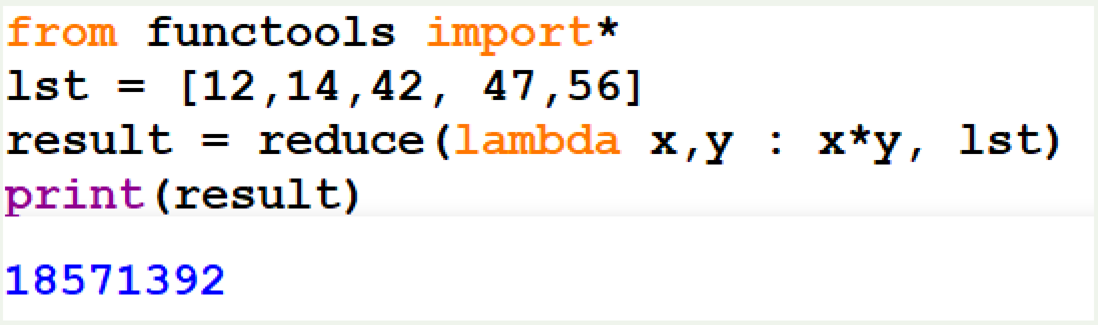
```
[0.0, 3.61, 6.71, 8.83, 3.46, 6.71, 5.74, 3.46, 5.92]
[0, 22, 86, 148, 20, 84, 62, 30, 50]
[0, 143, 1935, 5772, 120, 1890, 1023, 180, 875]
```

reduce() function

reduce () function reduces a sequences of elements into a single elemet by applying the specified function.

reduce (function, sequence)

Reduce () function present in functools module and hence we should write import statement.



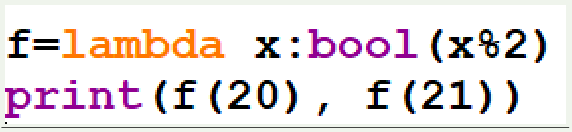
```
from functools import*
lst = [12,14,42, 47,56]
result = reduce(lambda x,y : x*y, lst)
print(result)

18571392
```

Conclusion
That's all for today. We learned quite a bit about Python3 lambda expression with syntax, how to create Lambda expressions in python. Then we saw what qualifies as an expression, and also learned how to provide default arguments in Python Lambda expression. Finally, we tried out three functions, filter(), map(), and reduce(), to make use of lambdas. A lambda, as compared to a normal function, offers its own benefits.

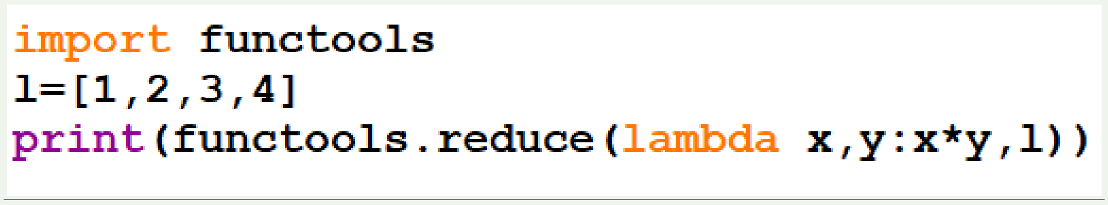
Exercices

1. **What is the output of the code shown below?**



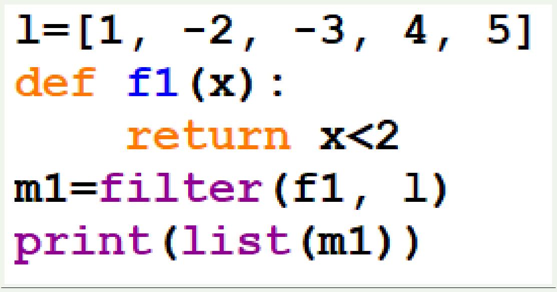
```python
f=lambda x:bool(x%2)
print(f(20), f(21))
```

2. **What is the output of the code shown below?**



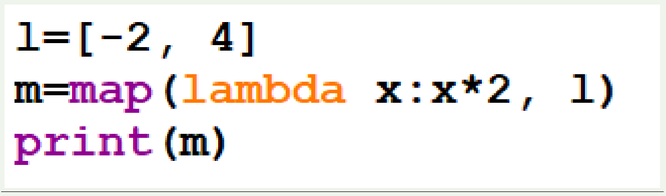
```python
import functools
l=[1,2,3,4]
print(functools.reduce(lambda x,y:x*y,l))
```

3. **What is the output of the code shown below?**



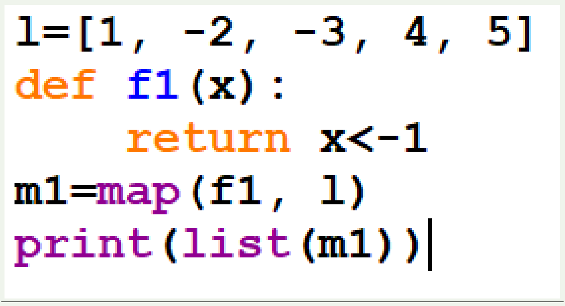
```python
l=[1, -2, -3, 4, 5]
def f1(x):
    return x<2
m1=filter(f1, l)
print(list(m1))
```

4. **What is the output of the code shown below?**



```python
l=[-2, 4]
m=map(lambda x:x*2, l)
print(m)
```

5. **What is the output of the code shown below?**



```
l=[1, -2, -3, 4, 5]
def f1(x):
    return x<-1
m1=map(f1, l)
print(list(m1))
```