

19.SpecialFunctions

June 1, 2020

0.1 Special Functions in Python

0.1.1 Anonymous Function or Inline function

Syntax:

```
fun_name = lambda arg1, arg2, .. : return_statement
```

```
[1]: def hello():  
      print("Hello world")
```

```
[2]: hello()
```

Hello world

```
[3]: (lambda : print("hello world"))()
```

hello world

```
[4]: h = lambda : print("Hello World")
```

```
[5]: h()
```

Hello World

```
[6]: h = lambda : (print("hi"), print("Bye"))
```

```
[8]: h()  
      # (None, None)  
      print('_'*30)
```

hi

Bye

```
[9]: def even(num):  
      if num % 2 == 0:  
          return "EVEN"
```

```
    return "ODD"
```

```
[10]: even(10)
```

```
[10]: 'EVEN'
```

```
[11]: even(5)
```

```
[11]: 'ODD'
```

```
[12]: e = lambda num: "EVEN" if num % 2 == 0 else "ODD"
```

```
[13]: print(e)
```

```
<function <lambda> at 0x0000025935AAF8B8>
```

```
[14]: e(10)
```

```
[14]: 'EVEN'
```

```
[15]: even(5)
```

```
[15]: 'ODD'
```

```
[17]: (lambda num: num**2)(5) # Anonymous
```

```
[17]: 25
```

Mapping Syntax

```
map_object = map( function, iterable)
```

```
[18]: data = [ var for var in range(1, 11)]  
      print(data)
```

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

```
[19]: def func(num):  
      return num**2  
      sq_data = map( func, data )  
      # func, data
```

```
[20]: print(sq_data)
```

```
<map object at 0x000002593798EBC8>
```

```
[21]: print(next(sq_data))  
      # 1 -> func -> 1
```

```
# state 1
```

1

```
[22]: print(next(sq_data))  
# 2 -> func -> 4
```

4

```
[23]: print(*sq_data)
```

9 16 25 36 49 64 81 100

```
[24]: next(sq_data)
```

```
↳ -----  
StopIteration                                Traceback (most recent call↳  
↳last)
```

```
<ipython-input-24-f964b1086465> in <module>  
----> 1 next(sq_data)
```

StopIteration:

Eager Execution

Lazy Execution

map is a single used object

map works on principle of lazy evaluation

```
[26]: from sys import getsizeof
```

```
[28]: m = map(lambda x:x**2, [ 1, 2, 3, 4])  
print(getsizeof(m)) # lazy evaluation
```

56

```
[29]: l = list(m)  
print(getsizeof(l)) # eager
```

120

```
[30]: print(*m)
```

```
[31]: print(l)
```

```
[1, 4, 9, 16]
```

```
[35]: l = [ 1, 2, 3, 4, 5]
m = map( lambda x: x**2 , l)
# map lambda func <- l (ref) <- 0 index
```

```
[36]: print(m)
```

```
<map object at 0x0000025937A03448>
```

```
[37]: l.pop(1); l.pop(3)
# [1, 3, 4, 5], [ 1, 3, 4]
# l = [1, 3, 4]
print("Now let's print square of elements.")

print(*m) # output ? l-> 0 -> 1 -> lambda -> 1, 1 -> 3 -> 9, 2 -> 4 -> 16, ?
```

```
Now let's print square of elements.
1 9 16
```

```
[38]: l = [ 1, 2, 3, 4, 5]
for e in l:
    print(e)
    l.pop(0)
```

```
1
3
5
```

```
[39]: def sq(a, b):
        return a**2 + b**2
```

```
[40]: sq(1, 2)
```

```
[40]: 5
```

```
[42]: s1 = [1, 2, 3]
s2 = [4, 5, 6]
```

```
[43]: l = list( map( sq, s1, s2 ) ) # list for eager because map is lazy
```

```
[44]: print(l)
```

[17, 29, 45]

```
[45]: text = """The coronavirus outbreak came to light on December 31, 2019
when China informed the World Health Organisation of a cluster of
cases of pneumonia of an unknown cause in Wuhan City in Hubei Province.
Subsequently the disease spread to more Provinces in China, and to the
rest of the world.The WHO has now declared
it a pandemic. The virus has been named SARS-CoV-2 and
the disease is now called COVID-19"""
```

```
[47]: words = text.split()
```

we have to transform data into lower case

```
[51]: new_words = map(lambda word: word.lower(), words)
```

filter choose values based boolean value return by functions

syntax:

```
filter_object = filter(func, sequence)
```

note - func should return True or False to filter sequence data

```
[53]: from random import randint
```

```
[54]: data = [ randint(1, 50) for var in range(20) ]
```

```
[55]: data
```

```
[55]: [47, 34, 20, 23, 11, 2, 27, 11, 32, 50, 7, 13, 22, 35, 38, 48, 27, 33, 48, 35]
```

```
[56]: even = list( filter( lambda num: True if num % 2 == 0 else False, data ) )
odd = list( filter( lambda num: True if num % 2 else False, data))
```

```
[59]: print("Data: ", data)
print("Even: ", even)
print("Odd : ", odd)
```

Data: [47, 34, 20, 23, 11, 2, 27, 11, 32, 50, 7, 13, 22, 35, 38, 48, 27, 33, 48, 35]

Even: [34, 20, 2, 32, 50, 22, 38, 48, 48]

Odd : [47, 23, 11, 27, 11, 7, 13, 35, 27, 33, 35]

```
[63]: chars = list(" ".join(words))
```

```
[74]: chr(122)
```

```
[74]: 'z'
```

```
[75]: seq = [' ', '\t', '\n', ] + [ chr(var) for var in range(65, 91) ] + [ chr(var)
↳ for var in range(97, 123)]
```

```
[78]: print(seq)
```

```
[' ', '\t', '\n', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z', 'a', 'b',
'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
's', 't', 'u', 'v', 'w', 'x', 'y', 'z']
```

```
[81]: special = list( filter( lambda char: char not in seq, chars))
```

```
[82]: print(special)
```

```
['3', '1', ',', '2', '0', '1', '9', '.', ',', '.', '.', '-', '-', '2', '-', '1',
'9']
```

OOPs

encapsulation

data hiding

inheritance

polymorphism

abstraction

message passing

object

class

constructor

destructor

Reduce

it returns single value and apply function repeatatively untill converge to result

Syntax:

```
from functools import reduce
```

```
value = reduce(func, seq, initial)
```

```
[97]: seq = [ 1, 2, 3]
      initial = 0
      def add(a, n):
          print("__"*20)
          print("Initial: ", a)
          print("Next    : ", n)
          print("__"*20)
          return a+n
```

```
[98]: from functools import reduce
```

```
[100]: value = reduce(add, seq, initial)
       print("Answer: ", value)
```

```
-----
Initial:  0
Next     :  1
-----
```

```
-----
Initial:  1
Next     :  2
-----
```

```
-----
Initial:  3
Next     :  3
-----
```

```
-----
Answer:  6
-----
```

```
[93]: print("Answer: ", value)
```

```
Answer:  15
```

```
[101]: ans = reduce(lambda x,y:x+y**2, [1, 2, 3, 4], 5)
```

```
[102]: print("Answer: ", ans)
```

```
Answer:  35
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```

[]:

[]: