Map, Filter and Reduce in Python | Advanced Python

Introduction:

In this blog we will talk about some really useful functions which will easily do some processes in your code for you. Functions are as follows:

- Map
- Filter
- Reduce

Note: All these three functions return an object so we have to type cast it to list with the help of list() function.

Map:

Map function takes **two arguments**. It takes a **function name and a list** and it automatically applies that function on all the list items and returns an object. We don't have to make a for loop and pass those list items through it one by one. Also with the help of this function, code becomes extremely small. Here is an example:

Without map function:

```
def square(n):
    return n**2

h1 = [1,2,4,5,7]
```

```
sq =[]
for item in h1:
    sq.append(item**2)
print(sq)
```

With map function:

```
def square(n):
    return n**2

h1 = [1,2,4,5,7]
sq = list(map(square, h1))
print(sq)
```

You can also pass list through **multiple functions** with the help of map function.

Filter:

It does what it is named, it filters. It filters a list for you according to the function you gave it. It takes two arguments, **function name and list name**. Function should **return true or false**. It takes the list and for whatever list item the function **returns true**, it filters all those items and shows you. Here is an example:

```
def greater_than_2(n):
    if n>2:
        return True
    else:
        return False

h1 = [1,2,3,4,5,6,7,-2,-5]
greater_th_2 = list(filter(greater_than_2, h1))
print(greater_th_2)
```

Reduce:

We can't directly use reduce like map or filter function. We have to **import it first**. To import it write:

```
from functools import reduce
```

It also takes a function and a list and applies that function on each list item but the difference is, other two functions separately applied the given function on every list item but this function applies it like a chain on every list item. Here is an example for you to understand better:

```
from functools import reduce

def string_sum(a, b):
    return a+b

newstring = reduce(string_sum, ["You", " are", " on", " codewithharry"])
print(newstring)
```

Output:

```
You are on codewithharry
```

It could be used like this too:

```
from functools import reduce

def sum_num(a, b):
    return a+b

li = reduce(sum_num, [1,2,3,4])
print(li)
```

Output:

```
10
```

Like this it links the next list element with the previous output.

#tut6a.py file as described in the video

```
#Map function

# map(function_to_apply, list of inputs)

def square(n):
    return n**2

h1 = [1,2,4,5,7]
# sq =[]
# for item in h1:
#     sq.append(item**2)

sq = list(map(square, h1))
print(sq)
```

#tut6b.py file as described in the video

```
def greater_than_2(n):
    if n>2:
        return True
    else:
        return False

h1 = [1,2,3,4,5,6,7,-2,-5]
greater_th_2 = list(filter(greater_than_2, h1))
print(greater_th_2)
```

#tut6c.py file as described in the video

```
from functools import reduce
```

Copy

```
def sum_num(a, b):
    return a+b


li1 = reduce(sum_num, [1,2,3,4])
print(li1)
```

Previous

Next

CodeWithHarry

Copyright © 2022 CodeWithHarry.com