Take-Home (Day 4)

Let's begin with some hands-on practice exercises

1. A list of words is given. Convert the words into uppercase (use lambda and map functions)

lsbool = ['True','FALse','tRUe','tRue','False','faLse','True']
Output: ['TRUE', 'FALSE', 'TRUE', 'TRUE', 'FALSE', 'FALSE', 'TRUE']

In [2]:

```
# type your code here
lsbool = ['True','FALse','tRUe','tRue','False','faLse','True']
res_lsbool = map(lambda x:x.upper(),lsbool)
print(res_lsbool)
print(list(res_lsbool))
```

```
<map object at 0x0000020CF047EA08>
['TRUE', 'FALSE', 'TRUE', 'TRUE', 'FALSE', 'FALSE', 'TRUE']
```

2. Write a function to print the height of students in the class (here count of students is not fixed)

In [5]:

```
# type your code here
def height(*x):
    print(x)
height(10,20,30,40,50,60,70)
```

```
(10, 20, 30, 40, 50, 60, 70)
```

3. Create a dictionary containing four lambda functions square, cube, squareroot, multiply by 2.

E.g. dict = {'Square': function for squaring, 'Cube': function for cube, 'Squareroot': function for squareroot, 'Double': function for double} and so on

Pass the values (input from the user) to the functions in the dictionary respectively. Then add the outputs of each function and print it.

In [2]:
```
# type your code here
import math
x = {'Square':lambda m : m*m, 'Cube':lambda m : m**3 , 'Squareroot': lambda m :
def di():
    n = int(input("Enter a positive integer: ")),
    a = []
    for i in x.values():
        a.append(list(map(i,n)))
    return a
di()
```

```
Enter a positive integer: 24
```

Out[2]:
```
[[576], [13824], [4.898979485566356], [48]]
```

4. A list of words is given. Define a function to find the words from the list that have their second character in uppercase

ls = ['hello', 'Dear', 'hOw', 'ARe', 'You']

Output: ['hOw', 'ARe']

In [3]:
```
# type your code here
ls = ['hello', 'Dear', 'hOw', 'ARe', 'You']
print(list(filter(lambda x:x[1].isupper(),ls)))
```

```
['hOw', 'ARe']
```

5. Define a function to filter the marks from the tuple "tmarks" that are greater than 40 and print corresponding names from the "tnames" tuple

tnames = ('John','Sharon','Jack','Annie')

tmarks = (32,50,75,12)

In [4]:
```
# type your code here
tnames = ('John','Sharon','Jack','Annie')
tmarks = (32,50,75,12)

[(i,j) for i,j in zip(tnames,tmarks) if j>40 ]
```

Out[4]:
```
[('Sharon', 50), ('Jack', 75)]
```

6. A dictionary of names and their weights on earth is given. Find how much they will weigh on the moon. (use map and lambda functions)

Weight of people in kg WeightOnEarth = {'John':45, 'Shelly':65, 'Marry':35}

Gravitational force on the Moon GMoon = 1.622

Gravitational force on the Earth GEarth = 9.81

Output: Weight on Moon: {'John': 7.44, 'Shelly': 10.75, 'Marry': 5.79}

In [5]:

```
# type your code here
WeightOnEarth = {'John':45, 'Shelly':65, 'Marry':35}
GMoon = 1.622
GEarth = 9.81
WeightonMoon = list(map(lambda x: (x/GEarth)*GMoon , WeightOnEarth.values()))
print(WeightonMoon)
```

```
[7.440366972477065, 10.747196738022426, 5.786952089704384]
```

7. A list containing multiple lists is given. Convert each inner list into sets and find intersection of all the sets (use reduce function)

given_sets = [[1, 2, 3, 4, 8], [2, 3, 8, 5, 6], [8, 4, 5, 3, 7], [6, 9, 8, 3], [9, 12, 3, 7, 6, 8, 4, 6, 21, 1, 6]]
Output: {8, 3}

In [6]:

```
# type your code here
from functools import reduce
given_sets = [[1, 2, 3, 4, 8], [2, 3, 8, 5, 6], [8, 4, 5, 3, 7], [6, 9, 8, 3], [9
a = list(map(lambda x:set(x),given_sets))
reduce(lambda x,y: x.intersection(y),a)
```

Out[6]:

```
{3, 8}
```

8. Write a function to store the price of different products in the supermarket (number of products is not fixed)

In [7]:

```
# type your code here

def product():
    n = int(input('enter number of items: '))
    price_list = [[input('Product name : '), float(input('product Price '))] for
    return price_list

product()
```

```
enter number of items: 2
Product name : Onion
product Price 50
Product name : Carrot
product Price 30
```

Out[7]:

```
[['Onion', 50.0], ['Carrot', 30.0]]
```

9. Define a function to multiply two global variables, 16 and 12

In [9]:

```
# type your code here
x = 58
y = 32
def multiply():
    global x
    global y
    a = x*y
    return a
multiply()
```

Out[9]:

```
1856
```

10. Find the cumulative average of the list [9,8,7,6,5] using accumulate() and lambda function

Input list: [9,8,7,6,5]

In [11]:

```
# type your code here

from itertools  import accumulate
a =  [9,8,7,6,5]
print(list(accumulate(a,lambda x,y: (x+y))))
```

```
[9, 17, 24, 30, 35]
```