In [2]:
```
def cube(x):
    return x*x*x
print(cube(2))
```

```
8
```

In [4]:
```
l = [1,4,6,4,8]
new_items = []
for items in l:
    new_items.append(cube(items))
print(new_items)
```

```
[1, 64, 216, 64, 512]
```

MAP

In [12]:
```
def cube(x):
    return x*x*x

l = [1,5,9,-4,6]
new_l = list(map(cube, l))
print(new_l)
```

```
[1, 125, 729, -64, 216]
```

In [13]:
```
def add(x,y):
    return x + y
l1 = [1,5,-2,8]
l2 = [4,-1,6,9]

l = list(map(add,l1,l2))
print(l)
```

```
[5, 4, 4, 17]
```

MAP WITH LAMBDA

In [15]:
```
numbers = (2,4,6,8,7)
result = list(map(lambda x:x**3, numbers))
print(result)
```

```
[8, 64, 216, 512, 343]
```

In [19]:
```
str = ('c','c++','java','python')
str_u = list(map(lambda x:x.upper(), str))
print(str_u)
```

```
['C', 'C++', 'JAVA', 'PYTHON']
```

In [21]:
```
l1 = [12,56,89,-6,42]
l2 = [23,45,-34,28,19]
l = list(map(lambda x,y:x+y, l1,l2))
print(l)
```

```
[35, 101, 55, 22, 61]
```

FILTER

In [23]:
```
def filter_function(a):
    return a>4

num = [12,-3,0,9,2,7,1,6]
new_a = list(filter(filter_function, num))
print(new_a)
```

```
[12, 9, 7, 6]
```

In [24]:
```
def filter_even(a):
    return a%2==0
num = [10,13,11,7,-4,88,42,91,-64]
even = list(filter(filter_even, num))
print(even)
```

```
[10, -4, 88, 42, -64]
```

FILTER WITH LAMBDA

In [25]:
```
num = [10,13,11,7,-4,88,42,91,-64]
even = list(filter(lambda x:x%2 == 0, num))
odd = list(filter(lambda x:x%2!=0, num))
print(even)
print(odd)
```

```
[10, -4, 88, 42, -64]
[13, 11, 7, 91]
```

In [27]:
```
names = ["Alice", "Bob", "Ankit", "Arun", "Ravi"]
result = list(filter(lambda name: name.startswith('A'), names))
print(result)
```

```
['Alice', 'Ankit', 'Arun']
```

REDUCE

In [29]:
```
from functools import reduce
numbers = [1,4,6,8,2]

result = reduce(lambda x,y:x+y, numbers)
print(result)
```

```
21
```

In [30]:
```
from functools import reduce
numbers = [3, 5, 2, 8, 1]

result = reduce(lambda a, b: a if a > b else b, numbers)
print(result)
```

```
8
```

In [31]:
```
from functools import reduce
nums = [2, 3, 4]

result = reduce(lambda x, y: x * y, nums)
print(result)
```

```
24
```

In [ ]: