In [1]:

```
first_list = [2, 4, 5]
print(first_list**2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-1-1f41c4b43dbf> in <module>()
      1 first_list = [2, 4, 5]
----> 2 print(first_list**2)

TypeError: unsupported operand type(s) for ** or pow(): 'list' and 'int'
```

In [2]:

```
first_list = [2, 4, 5]
```

In [3]:

```
print(map(lambda x: x**2, first_list))
```

```
<map object at 0x000001C529EA86A0>
```

In [4]:

```
first_list = [2,4,5]
print(list(map(lambda x: x**2, first_list)))
```

```
[4, 16, 25]
```

In [5]:

```
def squareit(n):
    return n**2
print(list(map(squareit, first_list)))
```

```
[4, 16, 25]
```

In [6]:

```
sums_list = [3,5,9,7]
sums_list2 = (4,5,6,7)
print(list(map(lambda x,y : x+y, sums_list,sums_list2)))
```

```
[7, 10, 15, 14]
```

In [7]:

```
list_of_names = ['nikola', 'james', 'albert']
list_of_names2 = ['tesla','watt','einstein']
proper = lambda x, y: x[0].upper()+x[1:] +' '+ y[0].upper()+y[1:]
print(list(map(proper, list_of_names,list_of_names2)))
```

```
['Nikola Tesla', 'James Watt', 'Albert Einstein']
```

In [8]:

```
#Filter
divby3 = lambda x:  x % 3 == 0
my_list = [3,4,5,6,7,8,9]
div = filter(divby3, my_list)
print(list(div))
```

```
[3, 6, 9]
```

In [11]:

```
#Reduce
from functools import reduce
q  = reduce(lambda x, y: x+y, range(1,4))
print(q)
```

```
6
```

In [12]:

```
list_of_nums = [22,45,32,20,87,94,30]
print(reduce(lambda x,y: x if x>y else y,list_of_nums))
```

```
94
```

In [17]:

```
list_of_nums = [4,3,2,1]
print(reduce(lambda x,y: x if x>y else y,list_of_nums))
```

```
4
```

In [20]:

```
list_of_nums = [5,4,3,2,1]
q  = reduce(lambda x, y: x+y, list_of_nums)
print(q)
```

```
15
```