1. Sorting Lists

In [6]:
```
# sorted function returns new list
li = [9, 1, 2, 3, 6, 7, 5, 4, 8]

s_li = sorted(li)
sr_li = sorted(li, reverse=True)

print('Sorted Variables:\t', s_li)
print('Reversely Sorted Variables:\t', sr_li)
print('Original Variables:\t', li)

# Sort the list without creating new variable
li.sort()
print('Original Variables:\t', li)
```

```
Sorted Variables:        [1, 2, 3, 4, 5, 6, 7, 8, 9]
Reversely Sorted Variables:      [9, 8, 7, 6, 5, 4, 3, 2, 1]
Original Variables:      [9, 1, 2, 3, 6, 7, 5, 4, 8]
Original Variables:      [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

2. Sorting the Other Objects

We can use sorted function to other objects.

In [8]:
```
tup = (9, 1, 2, 3, 6, 7, 5, 4, 8)
s_tup = sorted(tup)
print('Tuple\t', s_tup)

di = {'name' : 'Corey', 'job' : 'programming', 'age' : 22, 'os' : 'Linux'}
s_di = sorted(di)
print('Dict\t', s_di)
```

```
Tuple    [1, 2, 3, 4, 5, 6, 7, 8, 9]
Dict     ['age', 'job', 'name', 'os']
```

3. Sort the objects in different criteria

In [9]:
```
li = [-6, -5, -4, 1, 2, 3]
s_li = sorted(li, key=abs)
print(s_li)
```

```
[1, 2, 3, -4, -5, -6]
```

In [15]:
```
class Employee():
    def __init__(self, name, age, salary):
        self.name = name
        self.age = age
        self.salary = salary

    def __repr__(self):
        return f'({self.name}, {self.age}, ${self.salary})'

e1 = Employee('Carl', 37, 70000)
e2 = Employee('Sarah', 29, 80000)
e3 = Employee('John', 43, 90000)


employees = [e1, e2, e3]
s_employees = sorted(employees, key=lambda x:x.salary, reverse=True)
s_employees
```

Out[15]:
```
[(John, 43, $90000), (Sarah, 29, $80000), (Carl, 37, $70000)]
```