Comprehension

- is helpful to reduce the code

Types of comprehensions

- list comprehension
- set comprehension
- dict comprehension

In [2]:
```
l = [1,2,3,4,3,4,2,3,2,2]  # #[1,2,3,4]
list(set(l))
```

Out[2]:
```
[1, 2, 3, 4]
```

In [8]:
```
# print all the even numbers from the given list
l = [1,4,2,3,4,3,4,2,3,2,2,6] #[2,4]
l1 = []
for i in l: #1,4,2,3,4,3,4...
    if i%2==0: #4,2,4,4,2,2,2,6
        if i not in l1: #4,2,4
            l1.append(i)  #[4,2,6]
l1.sort()
l1
```

Out[8]:
```
[2, 4, 6]
```

In [9]:
```
# Display all the unique elements from the given list without using
#set and list as a keywords
l = [1,4,2,3,4,3,4,2,3,2,2,6] #[2,4]
l1 = []
for i in l: #1,4,2,3,4,3,4...
    if i not in l1: #4,2,4
        l1.append(i)  #[4,2,6]
l1.sort()
l1
```

Out[9]:
```
[1, 2, 3, 4, 6]
```

```
#### List Comprehension  []
syntax:

- if cotains only if condition===> [output loop condition]
- if contains both if and else ==> [output(if) condition loop]
```

In [13]:
```
l = [1,4,2,3,4,3,4,2,3,2,2,6]  # [4,2,4,4,2,2,2,6]
l1=[]
for i in l:
    if i%2==0:
        print(i,end = " ")
        l1.append(i)
print(l1)
```

```
4 2 4 4 2 2 2 6 [4, 2, 4, 4, 2, 2, 2, 6]
```

In [43]:
```
s = [i for i in l if i%2==0] #[2,4,6]
list(set(s))

```

Out[43]:
```
[2, 4, 6]
```

In [20]:
```
n = list(set([i for i in l if i%2==0]))
n.sort()
n
```

Out[20]:
```
[2, 4, 6]
```

In [44]:
```
l = [1,2,3,4,"a","b","c","b","c"] #["a","b","c"]
l1 = []
for i in l:
    if str(i).isalpha(): # 1   #'1'
        l1.append(i)
    else:
        l1.append("False")

print("the output is:",l1)
```

```
the output is: ['False', 'False', 'False', 'False', 'a', 'b', 'c', 'b', 'c']
```

In [45]:
```
s = [True if str(i).isalpha() else "False" for i in l]
print("the output is:",s)
```

```
the output is: ['False', 'False', 'False', 'False', True, True, True, True, Tru
e]
```

In [24]:
```
l1 = [1,2,3]
l2 = ["a","b","10",20]
l1+l2

# ["1","2","3","a","b","10"]
# [1, 2, 3, 'a', 'b',10]
```

Out[24]:
```
[1, 2, 3, 'a', 'b', '10', 20]
```

```
#### Set Comprehension
syntax:

```

```
- if cotains only if condition===> {output loop condition}
- if contains both if and else ==> {output(if) condition loop}
```

In [39]:
```
l = [1,4,2,3,4,3,4,2,3,2,2,6]  # {2,4,6}
{i for i in l if i%2==0}
```

Out[39]: {2, 4, 6}

In [47]:
```
print(l)
{i if str(i).isalpha() else "False" for i in l}
```

```
[1, 2, 3, 4, 'a', 'b', 'c', 'b', 'c']
```

Out[47]: {'False', 'a', 'b', 'c'}

```
#### Dictionary Comprehension
syntax:

- if cotains only if condition===> {output(key:value) loop condition}
- if contains both if and else ==> {output(if) condition loop}
```

In [54]:
```
l = [1,4,2,3,4,3,4,2,3,2,2,6] # {1:1,2:4,3:3,4:3,6:1},
# {1: 1, 4: 16, 2: 4, 3: 9, 6: 36}
l.sort()
d ={}
d1 = {}
for i in l:  # 1
    d[i] = l.count(i)  #d.add(i)
    d1[i] = i**2
print(d1)
print(d)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 6: 36}
{1: 1, 2: 4, 3: 3, 4: 3, 6: 1}
```

In [59]:
```
#print(l)
print({i:l.count(i) for i in l})
print({i:i**2 for i in l})

```

```
{1: 1, 2: 4, 3: 3, 4: 3, 6: 1}
{1: 1, 2: 4, 3: 9, 4: 16, 6: 36}
```

In [70]:
```
s = {}  # key:val
s["Apssdc"]=3**3
s
```

Out[70]: {'Apssdc': 27}

In [67]:
```
s = set({})
s.add("Apssdc")
s
```

Out[67]:
```
{'Apssdc'}
```

In [69]:
```
l = []
l.append(20)
l
```

Out[69]:
```
[20]
```

File Handling

- **File extension .txt** .Csv
- all the file data in file handling concept is string format

Steps involved in file handling

- open a file
- do the neccessary operations
- close a file

Methods and modes

- read # "r"
- write # "w","a"

In [75]:
```
# read method
f = open("file1.txt","r")
print(f.read())
f.close()
```

```
data1
data2
data3
data4
```

In [76]:
```
f = open("file2.txt","r")
print(f.read())
f.close()
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-76-b30e7d467852> in <module>
----> 1 f = open("file2.txt","r")
      2 print(f.read())
      3 f.close()

FileNotFoundError: [Errno 2] No such file or directory: 'file2.txt'
```

In [77]:
```
# write method
f = open("file1.txt","w")
f.write("data5")
f.close()
```

In [79]:
```
# append mode
f = open("file1.txt","a")
f.write("\n"+"data5")
f.close()
```

In [80]:
```
f = open("file2.txt","w")
f.write("This is a new file")
f.close()
```

In [ ]:
```

```