- We can store the output values by using append operations
- we are taking empty list
- we are iterating through elements from a given list
- the result we are appending in a new list

In [4]:

```
square_list=[]
for i in range(10):
    square_list.append(i*i)
square_list
```

Out[4]:

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

- There are almost 3 lines are in the process
- The Three lines we can make into a single line
- This is called as LIST COMPREHENSION

$pattern - 1$

Only for loop

In [ ]:

```
# syntax
# [<output>  <for loop>]
```

In [5]:

```
square_list=[]
for i in range(10):
    square_list.append(i*i)
square_list

# Q1) What is the output ? i*i
# Q2) what is the output list name? square_list
# Q3) What is the for loop ?  for i in range(10)
# [<output>  <for loop>]
sq_list=[i*i for i in range(10)]
```

Out[5]:

```
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
```

In [ ]:

```
#Q2) l1=['hyd','mumbai','chennai','blr']
#  upper_list= ['Hyd','Mumbai','Chennai','Blr']

#Q3) l2=['Hyd','Mumbai','Chennai','Blr']
# lower_list=['hyd','mumbai','chennai','blr']
```

In [7]:

```
#Q2) l1=['hyd','mumbai','chennai','blr']
#  upper_list= ['Hyd','Mumbai','Chennai','Blr']
l1=['hyd','mumbai','chennai','blr']
upper_list=[]
for i in l1:
    upper_list.append(i.upper())
upper_list
```

Out[7]:
```
['HYD', 'MUMBAI', 'CHENNAI', 'BLR']
```

In [8]:
```
# Q1) What is the output ? i.upper()
# Q2) what is the output list name? upper_list
# Q3) What is the for loop ?  for i in l1
# [<output>  <for loop>]
l1=['hyd','mumbai','chennai','blr']
upper_list=[i.upper() for i in l1]
upper_list
```

Out[8]:
```
['HYD', 'MUMBAI', 'CHENNAI', 'BLR']
```

In [9]:
```
#Q3) l2=['Hyd','Mumbai','Chennai','Blr']
# lower_list=['hyd','mumbai','chennai','blr']
l2=['Hyd','Mumbai','Chennai','Blr']
lower_list=[i.lower() for i in l2]
lower_list
```

Out[9]:
```
['hyd', 'mumbai', 'chennai', 'blr']
```

In [10]:
```
# Q4)l2=['Hyd','Mumbai','Chennai','Blr']
# sum of the indexes of all elements
l2=['Hyd','Mumbai','Chennai','Blr']
for i in l2:
    print(l2.index(i))
```

```
0
1
2
3
```

In [12]:
```
l2=['Hyd','Mumbai','Chennai','Blr']
idx_list=[l2.index(i) for i in l2]
sum(idx_list)
```

Out[12]:
```
6
```

In [ ]:
```
  1 l1=['Hyd', 'Mumbai', 'Chennai', 'Blr']
      2 Sum_List=[i for i in range(len(l1))]
----> 3 sum(Sum_List)
      4 Sum_List

TypeError: 'list' object is not callable
```

- making empty list will not works in List Comprehenshion
- intializations will not works in List Comprehenshion
- assignment opertaions also not works

In [13]:
```
# Fail
l2=['Hyd','Mumbai','Chennai','Blr']
for i in l2:
    val=l2.index(i)
    print(val)

idx_list=[val=l2.index(i) for i in l2]
```

```
  Cell In[13], line 7
    idx_list=[val=l2.index(i) for i in l2]
              ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

$pattern - 2$

for loop - if condition

In [ ]:

```
# syntax

#[<output>  <for loop>  <if condition>]
```

In [14]:

```
# Q5)l1=['Hyd','Mumbai','Chennai','blr']
#  ans=['Mumbai','Chennai']
#  we want lements which are len of element >4
l1=['Hyd','Mumbai','Chennai','blr']
for i in l1:
    if len(i)>4:
        print(i)
```

```
Mumbai
Chennai
```

In [15]:

```
# Q1) what is the output:  i
# Q2) what is the if condition: if len(i)>4
# q3) what is the for llop : for i in l1
#[<output>  <for loop>  <if condition>]
l1=['Hyd','Mumbai','Chennai','blr']
op=[i for i in l1 if len(i)>4]
op
```

Out[15]:

```
['Mumbai', 'Chennai']
```

In [ ]:

```
# Q3)l1=['Hyd','Mum#bai','Chen#nai','blr']
#  ans=['Mum#bai'',Chen#nai']
#  we want lements which are having '#'

# Q5)l1=['Hyd','Mumbai','chennai','blr']
#  ans= ['Hyd','Mumbai']
#  we want lements which are having first letter capital

# Q6)l1=['Hyd','Mum#bai','Chen#nai','blr']
#  ans_#=['Mum#bai'',Chen#nai']    for loop and if cond
#  ans_without_#=['Hyd'','blr']
#  we want lements which are having '#'  for loop if cond

# Q9) str1='virat.kohli@rcb.com, Rohit.sharma@mi.co, KL.Rahul@lucknow.com'
# Firstname=[]  second name=[]  cname =[]
# append first name should be in first name list
#        second name shoul be in second name list
#        thirs name will be in thirs name list
```

In [17]:

```
# Q6)l1=['Hyd','Mum#bai','Chen#nai','blr']
#  ans=['Mum#bai'',Chen#nai']
#  we want lements which are having '#'
l1=['Hyd','Mum#bai','Chen#nai','blr']
op=[]
for i in l1:
    if '#' in i:
        op.append(i)
op
```

Out[17]:
```
['Mum#bai', 'Chen#nai']
```

In [20]:
```
op_ash=[i for i in l1  if '#' in i]
op_ash
```

Out[20]:
```
['Mum#bai', 'Chen#nai']
```

In [21]:
```
op_not_in_ash=[i for i in l1  if '#' not in i]
op_not_in_ash
```

Out[21]:
```
['Hyd', 'blr']
```

In [28]:
```
l1=['Hyd','Mum#bai','Chen#nai','blr']
op_title=[i for i in l1  if i.istitle()]
op_title
```

Out[28]:
```
['Hyd']
```

In [52]:
```
# Q9) str1='virat.kohli@rcb.com, Rohit.sharma@mi.co, KL.Rahul@lucknow.com'
# Firstname=[]  second name=[]  cname =[]
# append first name should be in first name list
#        second name shoul be in second name list
#        thirs name will be in thirs name list

# Firstname
str1='virat.kohli@rcb.com, Rohit.sharma@mi.co, KL.Rahul@lucknow.com'
list1=str1.split(',')
list1[0][:list1[0].index('.')] # Virat
list1[1][:list1[1].index('.')] # Rohit
list1[2][:list1[2].index('.')] # KL
# list1[i][:<first_dot_index>]
#list1[i][:list1[i].index('.')]

f_name=[]
for i in list1:
    f_name.append(i[:i.index('.')])

[i[:i.index('.')] for i in list1]
```

Out[52]:
```
['virat', ' Rohit', ' KL']
```

In [34]:
```
list1[0],list1[1],list1[2]
```

Out[34]:
```
('virat.kohli@rcb.com', ' Rohit.sharma@mi.co', ' KL.Rahul@lucknow.com')
```

In [35]:
```
list1[0].index('.')
```

Out[35]:
```
5
```

In [46]:

```
str1='virat.kohli@rcb.com, Rohit.sharma@mi.co, KL.Rahul@lucknow.com'
list1=str1.split(',')
list1[0].index('.')
list1[2][:list1[2].index('.')]
```

Out[46]:

```
' KL'
```

In [53]:

```
str1='virat.kohli@rcb.com, Rohit.sharma@mi.co, KL.Rahul@lucknow.com'
list1=str1.split(',')
list1
```

Out[53]:

```
['virat.kohli@rcb.com', ' Rohit.sharma@mi.co', ' KL.Rahul@lucknow.com']
```

In [55]:

```
list1[0][:5]
# what is 5: index of first .
```

Out[55]:

```
'virat'
```

In [60]:

```
name1=list1[0]
ix=list1[0].index('.')
name1[:list1[0].index('.')]

list1[2][:list1[2].index('.')]

list1[i][:list1[i].index('.')]
```

Out[60]:

```
' KL'
```

In [61]:

```
for i in range(len(list1)):
    print(list1[i][:list1[i].index('.')])
```

```
virat
 Rohit
 KL
```

In [62]:

```
for i in list1:
    print(i[:i.index('.')])
```

```
virat
 Rohit
 KL
```

$Pattern - 3$

if-else

In [ ]:

```
[<output> <forloop>]   :  for loop
[<output> <forloop> <if condition>]: for-if
[<if_output> <if> <else> <else_output> <for>] : if-else
```

even-odd

In [2]:

```
even_list,odd_list=[],[]
for i in range(1,20):
    if i%2==0:
        even_list.append(f"even:{i}")
    else:
        odd_list.append(f"odd:{i}")
```

In [4]:

```
even_list,odd_list
```

Out[4]:

```
(['even:2',
  'even:4',
  'even:6',
  'even:8',
  'even:10',
  'even:12',
  'even:14',
  'even:16',
  'even:18'],
 ['odd:1',
  'odd:3',
  'odd:5',
  'odd:7',
  'odd:9',
  'odd:11',
  'odd:13',
  'odd:15',
  'odd:17',
  'odd:19'])
```

In [6]:

```
op_list=[]
for i in range(1,20):
    if i%2==0:
        op_list.append(f"even:{i}")
    else:
        op_list.append(f"odd:{i}")
op_list

#[<if_op> <if_con> <else> <else_op> <for_loop>]
[f"even:{i}" if i%2==0 else f"odd:{i}" for i in range(1,20)]
```

Out[6]:

```
['odd:1',
 'even:2',
 'odd:3',
 'even:4',
 'odd:5',
 'even:6',
 'odd:7',
 'even:8',
 'odd:9',
 'even:10',
 'odd:11',
 'even:12',
 'odd:13',
 'even:14',
 'odd:15',
 'even:16',
 'odd:17',
 'even:18',
 'odd:19']
```

In [ ]:

```

```