

## 30\_Nov

- **List comprehension:** [expression for item in iterable]
  - In list required [] square brackets
- **Set comprehension:** {expression for item in iterable}
  - In Set required {} curly braces
- **Dictionary comprehension:** {key: value for item in iterable}
  - In Dictionary required {key:value} curly braces with key : value
- List comprehensions is a way to write the code line in a single line
- When ever we use for loops and if conditions the time will be increase
- list comprehension will able to write **for loop and if condition** in a single line
- which reduce the time to get the output

```
In [1]: for i in range(1,11):
        print(i, end=' ')
```

```
1 2 3 4 5 6 7 8 9 10
```

```
In [2]: list1 = []
        for i in range(1,11):
            list1.append(i)

        list1
```

```
Out[2]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

- In above code we convert into a single line is called a list Comprehension

### Method-1: Only for loop

```
In [ ]: list1=[<code here>]

# Syntax
list1=[<output> <for loop>]
```

```
In [8]: lst1 = [i for i in range(1,11)]
        lst1
```

```
Out[8]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
In [ ]: # QUE:1- l1 = ['hyd', 'chennai', 'mumbai', 'pune']
        # ans= ['Hyd', 'Chennai', 'Mumbai', 'Pune']
```

```
In [11]: l1 = ['hyd','chennai','mumbai','pune']
#ans= ['Hyd', 'Chennai', 'Mumbai', 'Pune']
ans = []
for i in l1:
    ans.append(i.capitalize())
```

```
In [12]: lst = [i.capitalize() for i in l1]
lst
```

```
Out[12]: ['Hyd', 'Chennai', 'Mumbai', 'Pune']
```

```
In [ ]: #Que2:- l1 = ['hyd','chennai','mumbai','pune']
# ans1 = ['HYD', 'CHENNAI', 'MUMBAI', 'PUNE']
```

```
In [15]: l1 = ['hyd','chennai','mumbai','pune']
# ans1 = ['HYD', 'CHENNAI', 'MUMBAI', 'PUNE']

ans1=[]

for i in l1:
    ans1.append(i.upper())
```

```
In [16]: lst2 = [i.upper() for i in l1]
lst2
```

```
Out[16]: ['HYD', 'CHENNAI', 'MUMBAI', 'PUNE']
```

```
In [19]: #Que3:- l1 = ['hyd','chennai','mumbai','pune']
# ans = [0,10,20,30] multiple index with 10

l1 = ['hyd','chennai','mumbai','pune']
ans = []
for i in range(len(l1)): # using range solve
    ans.append(i*10)

ans
```

```
Out[19]: [0, 10, 20, 30]
```

```
In [21]: lst3 = [i*10 for i in range(len(l1))]
lst3
```

```
Out[21]: [0, 10, 20, 30]
```

```
In [27]: l1 = ['hyd','chennai','mumbai','pune']
ans = []
for i in l1: #using in solve
    ans.append(l1.index(i)*10)

ans
```

```
Out[27]: [0, 10, 20, 30]
```

```
In [28]: lst=[l1.index(i)*10 for i in l1]
lst
```

Out[28]: [0, 10, 20, 30]

```
In [29]: #Que3:- l1 = ['hyd', 'chennai', 'mumbai', 'pune']
# ans = [6] sum of indexs
summ = 0
l1 = ['hyd', 'chennai', 'mumbai', 'pune']
ans = []

for i in range(len(l1)):
    summ = summ+i
ans.append(sum)
ans
```

Out[29]: [6]

```
In [16]: lst1 = [summ = sum+i for i in l1]
lst1
```

Cell In[16], line 1  
 lst1 = [summ = sum+i for i in l1]  
 ^  
 SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?

- in list comprehension we can't give equal (=) to it will through the error

```
In [30]: lst4=[summ for i in range(len(l1))]
lst4
```

Out[30]: [6, 6, 6, 6]

```
In [1]: l1 = ['hyd', 'chennai', 'mumbai', 'pune']
summ = 0
ans = [l1.index(i) for i in l1]
sum(ans)
```

Out[1]: 6

## Method-2

- for loop with if condition

```
In [ ]: #syntax
lst = [<if output> <for loop> <if condition> ]
```

```
In [ ]: #Que6:- ['hyd', 'chen#ai', 'mu#bai', 'pune']
# ans=['chen#ai', 'mu#bai']
```

```
In [8]: l1 = ['hyd', 'chen#ai', 'mu#bai', 'pune']
# ans=['chen#ai', 'mu#bai']

for i in l1:
    if '#' in i:
        print(i)
```

chen#ai  
 mu#bai

```
In [11]: ans = [i for i in l1 if '#' in i]
ans
```

```
Out[11]: ['chen#ai', 'mu#bai']
```

```
In [ ]: #Que7:- ['hyd','chen#ai','mu#bai','pune']
# ans=['hyd', 'pune']
```

```
In [12]: l1 = ['hyd','chen#ai','mu#bai','pune']
# ans=['hyd', 'pune']

for i in l1:
    if "#" not in i:
        print(i)
```

```
hyd
pune
```

```
In [13]: lst = [i for i in l1 if "#" not in i]
lst
```

```
Out[13]: ['hyd', 'pune']
```

```
In [ ]: # Que:3- list1 = [100,27,20,37,38,47,52,87,94,239]
# using this we need to extract even and odd numbers
# even_list = []
# odd_list = []
```

```
In [19]: list1 = [100,27,20,37,38,47,52,87,94,239]
even_list = []
odd_list = []
for i in list1:
    if i%2==0:
        even_list.append(i)
    else:
        odd_list.append(i)

even_list,odd_list
```

```
Out[19]: ([100, 20, 38, 52, 94], [27, 37, 47, 87, 239])
```

```
In [22]: even_lst = [i for i in list1 if i%2==0]
even_lst
```

```
Out[22]: [100, 20, 38, 52, 94]
```

```
In [23]: odd_lst = [i for i in list1 if i%2!=0]
odd_lst
```

```
Out[23]: [27, 37, 47, 87, 239]
```

```
In [ ]: method-1: [<output> <for loop>]
method-2: [<output> <for loop> <if condition>]
method-3: [<if output> <if condition> <else> <else output> <for loop>]
```

```
In [28]: lst1 = [i if i%2==0 else i for i in list1]
lst1
```

Out[28]: [100, 27, 20, 37, 38, 47, 52, 87, 94, 239]

```
In [29]: lst1 = ['even' if i%2==0 else 'odd' for i in list1]
         lst1
```

Out[29]: ['even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd', 'even', 'odd']

```
In [30]: lst1 = [f'even{i}' if i%2==0 else f'odd:{i}' for i in list1]
         lst1
```

Out[30]: ['even100',  
'odd:27',  
'even20',  
'odd:37',  
'even38',  
'odd:47',  
'even52',  
'odd:87',  
'even94',  
'odd:239']

```
In [33]: l1 = [45,76,89,53,67,90,82,49,33,99]
         # 1. filter the students who pass marks>=50
         # 2. calculate grads of all students
         #      A: 85+, B: 70-84, C: 50-69 D:<50
         #      4 lists will come
         # 3. Add 5 grace marks to students who failed <50
         # 4. find the average marks of the class
```

Out[33]: [33, 45, 49, 53, 67, 76, 82, 89, 90, 99]

```
In [ ]: if l1 >= 50:
         if l1 >= 85:
             print("A")
         elif l1 >= 70:
             print("B")
         elif l1 >= 50:
             print("C")
         elif l1 <50:
             print("D")
```

```
In [35]: # 1. filter the students who pass marks>=50
         l1 = [45,76,89,53,67,90,82,49,33,99]

         [i for i in l1 if i>=50]
```

Out[35]: [76, 89, 53, 67, 90, 82, 99]

```
In [ ]: # 2. calculate grades of all students
         #      A: 85+, B: 70-84, C: 50-69 D:<50
         #      4 lists will come
         l1 = [45,76,89,53,67,90,82,49,33,99]
```

```
In [39]: l1 = [45,76,89,53,67,90,82,49,33,99]
         grade_A = []
         grade_B = []
         grade_C = []
         grade_D = []
```

```

for i in l1:
    if i >=85:
        grade_A.append(i)
    elif i >= 70:
        grade_B.append(i)
    elif i >=50:
        grade_C.append(i)
    else:
        grade_D.append(i)

print(grade_A)
print(grade_B)
print(grade_C)
print(grade_D)

```

```

[89, 90, 99]
[76, 82]
[53, 67]
[45, 49, 33]

```

#### Method-4: for if elif else

- There is no elif in list comprehension
- elif also we can write: else+if
- method-3: [ ]
- method-4: [ ]

```

In [ ]: - method-3: [<if output> <if condition> <else> <else output> <for loop>]

- method-4: [<if output> <if condition> <else> <elif out> <elif cond> <else> <el

```

```

In [17]: l1 = [45,76,89,53,67,90,82,49,33,99]
grade_A = []
grade_B = []
grade_C = []

for i in l1:
    if i >=85:
        grade_A.append(i)
    elif i >= 70:
        grade_B.append(i)
    else:
        grade_C.append(i)

l3 = [i if i >=85 else i if i >=70 else i for i in l1 ]
l3

```

```
Out[17]: [45, 76, 89, 53, 67, 90, 82, 49, 33, 99]
```

```

In [45]: l1 = [45,76,89,53,67,90,82,49,33,99]
[i if i >=85 else i if i >= 70 else i for i in l1]

```

```
Out[45]: [45, 76, 89, 53, 67, 90, 82, 49, 33, 99]
```

```
In [44]: l1 = [45,76,89,53,67,90,82,49,33,99]

[f'A:{i}' if i>85 else f'B:{i}' if i>=70 else f'C:{i}' for i in l1]
```

```
Out[44]: ['C:45',
          'B:76',
          'A:89',
          'C:53',
          'C:67',
          'A:90',
          'B:82',
          'C:49',
          'C:33',
          'A:99']
```

```
In [ ]: f'A:{i}' if i>85
        else f'B:{i}' if i >=70: else+if = elif
        else f'C:{i}' if i >50: else+if = elif
        else f'D:{i}'
```

```
In [ ]: - method-1: [<output> <for loop>]
        - method-2: [<if output> <for loop> <if condition>]
        - method-3: [<if output><if condition><else><else output> <for loop>]
        - method-4: [<if output><if condition><else><if output><if condition><else><else
```

```
In [46]: # 4. find the average marks of the class
```

```
l1 = [45,76,89,53,67,90,82,49,33,99]
summ =0
for i in l1:
    summ=summ+i
print(summ)
print('avg:',summ/len(l1))
```

```
683
avg: 68.3
```

```
In [47]: summ = 0
        [summ = summ+i for i in l1]
```

```
Cell In[47], line 2
    [summ = summ+i for i in l1]
    ^
SyntaxError: invalid syntax. Maybe you meant '==' or ':=' instead of '='?
```

### Walrus Operator

- in list comprehension = will not work
- If you want to write equal to (=), then will use Walrus operator :=

```
In [48]: summ = 0
        [summ := summ+i for i in l1]
```

```
Out[48]: [45, 121, 210, 263, 330, 420, 502, 551, 584, 683]
```

### List comprehension

- and

### **Lambda funtions**

**Very Very Important it will ask in interview 100%**

In [ ]: