Comprehensions

```
 - It is a process of creating new sequence from existed sequence

 - 3 Types of Comprehensions
       * list comprehension
       * dictionary comprehension
       * set comprehension
```

Advantages

- Less code
- Less complexity

# syntax for list comprehension

```
    [output_variable loop condition]
```

```
# i/p: [23,56,78,90,44,77,99]
# o/p : [23,77,99]

li = [23,56,78,90,44,77,99]
new = []
for i in li:
    if(i%2==1):
        new.append(i)
print(new)
```

```
[23, 77, 99]
```

```
k = [i for i in li if(i%2==1)]
print(k)
```

```
[23, 77, 99]
```

```
# print natural numbers upto 50
j = [i for i in range(50)]
print(j)
```

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19,
20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,
37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]
```

```
h = [i**2 for i in range(1,101)]
print(h)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256,
289, 324, 361, 400, 441, 484, 529, 576, 625, 676, 729, 784, 841, 900,
961, 1024, 1089, 1156, 1225, 1296, 1369, 1444, 1521, 1600, 1681, 1764,
1849, 1936, 2025, 2116, 2209, 2304, 2401, 2500, 2601, 2704, 2809,
2916, 3025, 3136, 3249, 3364, 3481, 3600, 3721, 3844, 3969, 4096,
4225, 4356, 4489, 4624, 4761, 4900, 5041, 5184, 5329, 5476, 5625,
5776, 5929, 6084, 6241, 6400, 6561, 6724, 6889, 7056, 7225, 7396,
7569, 7744, 7921, 8100, 8281, 8464, 8649, 8836, 9025, 9216, 9409,
9604, 9801, 10000]
```

```python
# sum of natural numbers
total = sum([i for i in range(1,101)])
print("sum: ",total)
```

```
sum:  5050
```

```python
# print numbers which are divisible by 3 and 5
num = [i for i in range(1,51) if(i%3==0 and i%5==0)]
print(num)
```

```
[15, 30, 45]
```

Task -01

1.Print vowels in a string using list comprehension

i/p: "list comprehension"

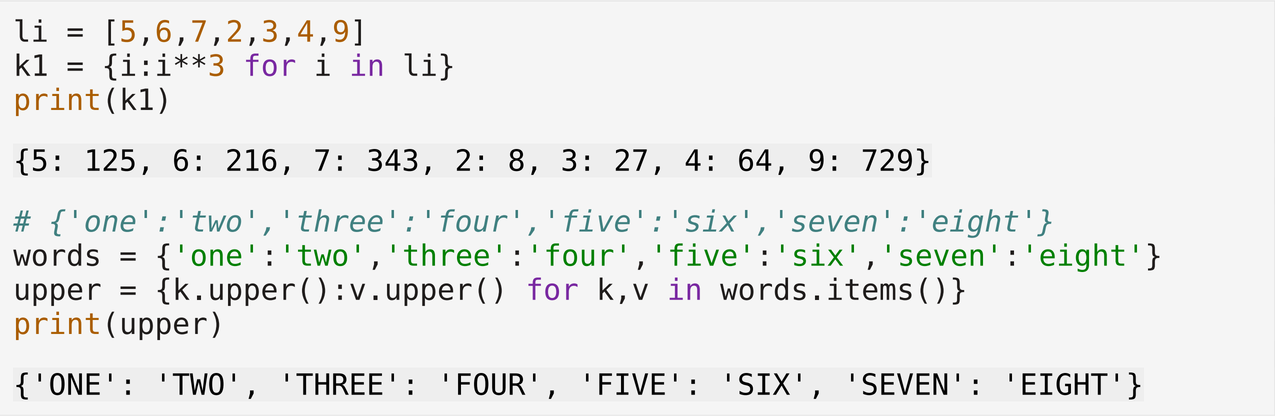
o/p: i o e

syntax for dictionary comprehension

[output_variable loop condition]

```python
k = {i:i**2 for i in range(1,6)}
print(k)
```

```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25}
```



```python
li = [5,6,7,2,3,4,9]
k1 = {i:i**3 for i in li}
print(k1)
```

```
{5: 125, 6: 216, 7: 343, 2: 8, 3: 27, 4: 64, 9: 729}
```

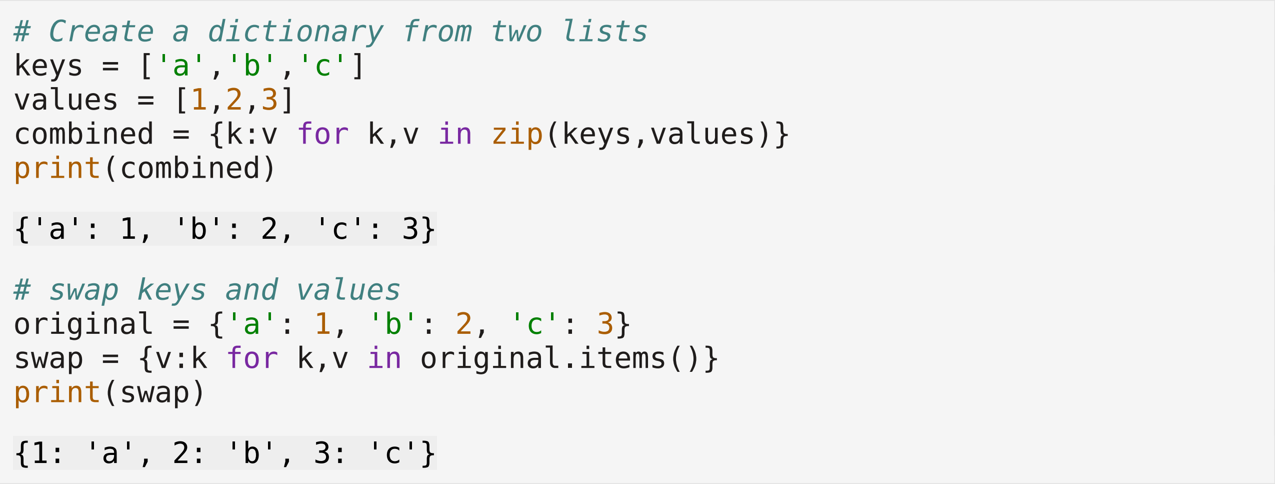
```python
# {'one':'two','three':'four','five':'six','seven':'eight'}
words = {'one':'two','three':'four','five':'six','seven':'eight'}
upper = {k.upper():v.upper() for k,v in words.items()}
print(upper)
```

```
{'ONE': 'TWO', 'THREE': 'FOUR', 'FIVE': 'SIX', 'SEVEN': 'EIGHT'}
```

TASK -02

I/P: {'ONE':'two','THREE':'four','FIVE':'six','SEVEN':'eight'}

O/P: {'one':'TWO','three':'FOUR','five':'SIX','seven':'EIGHT'}



```python
# Create a dictionary from two lists
keys = ['a','b','c']
values = [1,2,3]
combined = {k:v for k,v in zip(keys,values)}
print(combined)
```

```
{'a': 1, 'b': 2, 'c': 3}
```

```python
# swap keys and values
original = {'a': 1, 'b': 2, 'c': 3}
swap = {v:k for k,v in original.items()}
print(swap)
```

```
{1: 'a', 2: 'b', 3: 'c'}
```

task - 03

character frequency in a string

i/p: "dictionary"

o/p: {'d':1,'i':2,'c':1.............}

set comprehension

syntax :- {output_variable loop condition}

```
# print even numbers from 1 to 30 using set comprehension
even = {i for i in range(1,31) if(i%2==0)}
print(even)
```

```
{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30}
```

```
# print the values which are divisible by 3 or 5 using set
comprehension
n = {i for i in range(1,50) if(i%3==0 or i%5==0)}
print(n)
```

```
{3, 5, 6, 9, 10, 12, 15, 18, 20, 21, 24, 25, 27, 30, 33, 35, 36, 39,
40, 42, 45, 48}
```

```
# print common letters between two strings using set comprehension
str1 = "python"
str2 = "notebook"
common = {ch for ch in str1 if ch in str2}
print(common)
```

```
{'t', 'n', 'o'}
```

```
# print first letters of words
words = {'apple','banana','grape','mango','cherry'}
f_c={i[0] for i in words}
print(f_c)
```

```
{'m', 'g', 'b', 'a', 'c'}
```

Task - 04

i/p: words =
{'apple','banana','grape','mango','cherry'}

o/p: {'A','B','G','M','C'}