List comprehension in python provides a concise way to create lists by applying an expression to each item in an iterable.
It combine conditional statements in a single line of code, making it both efficient & readable.

Syntax :-

[expression for item in iterable if condition]

Ex :- numbers = [1, 2, 3, 4, 5, 6, 7]

print even numbers.

even_numbers = [x for x in numbers if x % 2 == 0]
#output -> [2, 4, 6]

Squared of the number.

squared_numbers = [n**2 for n in numbers]

#output -> [1, 4, 9, 16, 25, 36, 49]

abcd

dict[i] = 1

apsara

dic[i]
a : a
key

Date: __.__.____

Set :- A set is an unordered collection of unique elements in Python.

Sets are defined using braces {} or by using the set() function.

~~Set~~ Element in a set are immutable, but set itself is mutable. (i.e., you can add or remove elements).

Set support mathematical set operations such as union, intersection and difference.

~~my~~

Set funtions :-

① add() : Adds an element to the set.

```
~~my_set~~ = my_set = set([1, 2, 3])
my_set.add(4)
print(my_set)          # {1, 2, 3, 4}
```

② remove() : Remove element from the set; raise KeyError if the element is not present.

```
my_set.remove(3)
print(my_set)          # {1, 2, 4}
```

3) discard(): Remove an element if it exists; doesn't raise an error if the element is not present.

```
my_set.discard(2)
print(my_set)          # output :- {1, 4}
```

4) pop(): Remove & returns an arbitary element from the set.

```
element = my_set.pop()
print(element)          # output :- 1
print(my_set)           # output :-
```

(5) union(): Returns the union of 2 sets.

```
set_a = {1,2,3}
set_b = {3,4,5}

print(set_a.union(set_b))

# output :- {1,2,3,4,5}
```

apsara

Date: __.__.____

6. intersection() :- Returns the intersection (common element) of 2 sets.

```
print(set_a.intersection(set_b))
                                    # output : {3}
```

7. difference() :- Return the difference (element in first set but not in 2nd set).

```
print(set_a.difference(set_b))
                                    # output : {1,2}
```

8. clear() : Remove all elements from the set.

```
set_a.clear()
print(set_a)                        # output: set()
```