

09/10/25

Dictionaries :

→ A dictionary in python is a collection of key - value pairs.

→ It is unordered, mutable, and indexed by keys.

Syntax :

{ key 1 : value 1, key 2 : value 2, key 3 : value 3 }.

→ Each key must be unique and immutable.

→ The values can be of any data type and can be duplicated.

1. `union()` → Returns all elements from both sets.
2. `intersection()` → Returns common elements.
3. `difference()` → Elements present in first but not in second.
4. `Symmetric-difference()` → Elements not common to both.
5. `isdisjoint()` → Returns True if no common elements.
6. `issubset()` → checks if set is subset of another.
7. `issuperset()` → checks if set is superset of another.

09/10/25

Dictionaries :

→ A dictionary in python is a collection of key-value pairs.

→ It is unordered, mutable, and indexed by keys.

Syntax :

`{ key 1 : value 1 , key 2 : value 2 , key 3 : value 3 }`.

→ Each key must be unique and immutable.

→ The values can be of any data type and can be duplicated.

Ex `d1 = { 'name': 'vasu', 'DOB': '21/06/2004', 'ph no': '939177249' }`
`print (d1).`

o/p:

`{ 'name': 'vasu', 'DOB': '21/06/2004', 'ph no': '939177249' }`

1) `print (d1.keys())`

o/p: `dict_keys(['name', 'DOB', 'ph no'])`

2) `d1.values()`

o/p: `dict_values(['vasu', '21/06/2004', '939177249'])`

Dictionary Methods :

`dict.keys()` → Return all keys.

`dict.values()` → Return all values.

`dict.items()` → Return key value pairs.

`dict.get(key)` → Return value of key.

`dict.update(Other-dict)` → Add or updates items.

`dict.pop(key)` → Remove key value pairs.

`dict.clear()` → clear dictionary.

`dict.copy()` → Returns a shallow copy.

String :

2. Repetition:

"Hi" * 3

O/p: HiHiHi

3. Length:

len("python")

O/p: 6.

4. Membership:

'y' in "python"

O/p: True.

String Methods:

upper() → Converts to uppercase.

lower() → Converts to lowercase.

title() → Converts to title case.

strip() → Removes Spaces.

replace(Old, new) → Replaces substring.

split() → Splits into list.

join() → Joins list into string.

find(sub) → Finds index of substring.

count(sub) → Counts occurrences.

1) Count upper case char and lower case char.

```
str3 = 'python SUBJECT'
```

```
lc = 0
```

```
uc = 0
```

```
for i in str3:
```

```
    if i.isupper():
```

```
        uc += 1
```

```
    else:
```

```
        lc += 1
```

```
print(f"Lower Count = {lc} \n upper count = {uc}")
```

O/p :

=

lower count = 6

upper count = 8

List Comprehension :

→ List Comprehension is a short and elegant way to create lists in python - using a single line of code

→ List Comprehension allows you to create a new list by applying an expression to each item in an iterable.

Syntax :

new_list = [expression for item in iterable if condition]