09/10/25

Dictionaries :

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

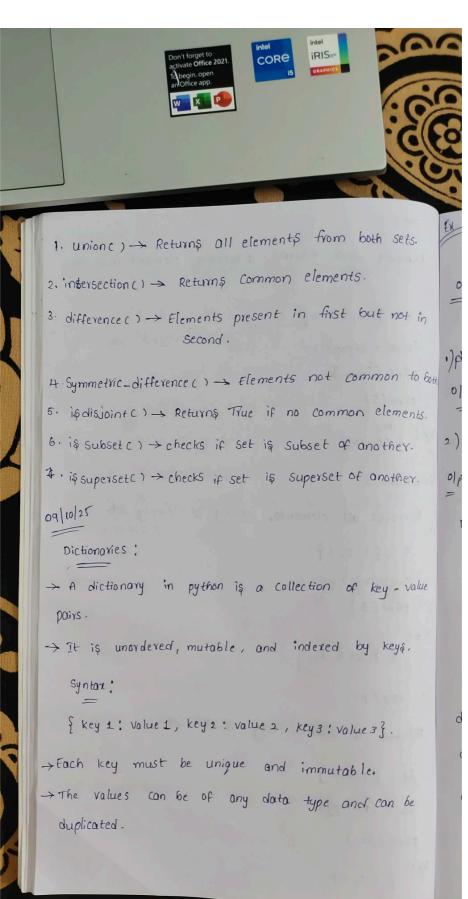
-> It is unordered, mutable, and indexed by keys.

Syntax:

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

- Fach key must be unique and immutable.

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





```
d1 = { 'name: 'vasu', 'DOB': '21 lo6/20 out, 'ph no': '9391777249'}
 print (d1).
0/0:
 { nome : Vasu, DOB : 21/06/2004, ph no : 9391777249}.
i) print (d1. keys())
olp: dict_keys (['name', 'Das', 'phno']).
2) d1. valuesc)
olp: dict-values [['vasu', '21 106 12004', '939 1777249']).
  Dictionary Methods:
  dict. keysc) -> 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 (7 -> Returns a Shallow copy.
```

```
String :
-> A string in python is a sequence of characters
 enclosed within single quotes (' '), double quotes (" ")
 or triple quotes (" " 08 " " u u u u).
-> A string is used to store text data.
 -> String is immutable.
 EX!
    Str1 = 'my name is vasantha!
     temp = Str1. split()
     print (temp)
     Str1 = 1'. join (temp)
     print (Sty 1).
 O[p:
     [ 'my, 'name', 'is, 'vasantha']
    myrame is vasantha.
   String operations:
  1. Concatenation!
 EN "Hello" + "World"
   Olp . Helloworld.
```

```
2. Repetition:
"#" * 3
01p:
3. length:
  len ("python")
 olp: 6.
4. Membership:
'y' in "python"
Olp; 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 inder of substring.
 Count (Sub) -> counts occurences.
```

```
1) Count upper case char and Lower case char.
     Str3 = pyTHon SubJECT
     1c = 0
     uc = 0
     for i in stra:
         if i. isupper():
             uc+=1
         else :
              1c+=1
    print (f Lower count = { LC } In upper count = { uc}")
 Olp:
  = 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 cool
-> 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]
```