6/10/2025, Monday

Dictionary - dict

Always values are stored in key : value pairs

Here, key is immutable and values are mutable.

We represent dict by dict() @@ {:}

```
pdata={'name':'Lekhs', 'ph_no':'8971818272', 'place':'Swift'}
pdata
o/p {'name':'Lekhs',
     'ph_no':'          ',
     'place':'          '}
```

* To extract keys of dict

```
pdata.keys()
o/p dict_keys(['name', 'ph_no', 'place'])
```

* values `pdata.values()`

```
o/p dict_values(['Lekhs', '          ', '          '])
```

* To update / replace value

```
pdata['ph_no']='6F61955F2'
```

* To insert new element

```
pdata['age']=23
```

* To delete a particular key & value @@ element

```
pdata.pop('ph_no')    o/p  6F61955F2
```

* To delete last element

```
pdata.popitem()
o/p ('age', 22)
```

* To delete the dict pdata.delete[]

⇒ sdata = {'sid': [1,2,3,4], 'name': ['a','b','c','d'],
'class': ['8th','10th','7th','10th']}

⇒ String – Sequence of characters
Immutable in structure
Update can be done only by using concatenation.

str = 'Machine Learning'
str o/p 'Machine Learng'

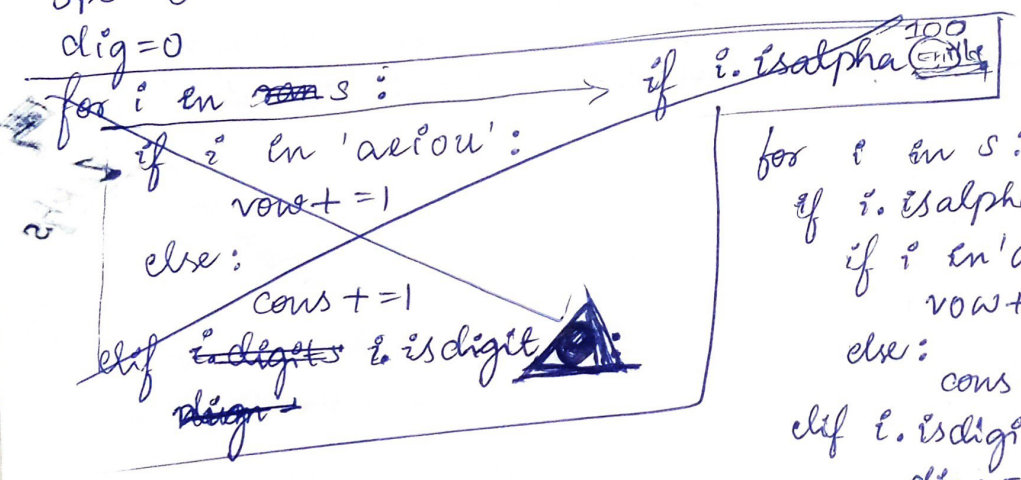
→ str[0] 'M'

→ str = str + 'python' o/p 'Machine Learnhingpython'

⇒ Count how many vowels, consonents, special characters and digits/no.s are there.

s = 'python@course$price$ is !2350'

vow = 0
cons = 0
spc = 0
dig = 0




```
for i in s:
    if i.isalpha():
        if i in 'aeiou':
            vow+=1
        else:
            cons+=1
    elif i.isdigit():
        dig+=1
    else:
        spc+=1
```

print(f"no. of vowels: {vow}\n no. of consonents: {cons}\n no. of sc: {spc}\n no. of digits: {dig}")

o/p no. of vowels: 7
12
4
4

⇒ Create a separate list for upper & lower case characters.

s = "PyTHon STUdEnts"

```
su=0
sl=0
for i in s:
    if in!=' ':
        if i.isupper():
            su+=1
        else:
            sl+=1
print(su,sl)
```

o/p

['P','T','O','S','U','E'] ['y','h','n','t','d','n','t','s']

1 Create a string made of 1st, middle and last character. Write a program to create a new string made of an input string's first, middle and last character

```
s = input("Enter a string: ")
new_string = (s[0] + s[len(s)//2] + s[-1])
print(new_string)
```

O/P Enter a string: priya → pia

Parana → paa

2. Create a string made of an ~~input~~ string's first, middle, ~~and last character~~. middle 3 characters of an input string.

```
s = input("Enter a string: ")
mid = len(s)//2
```

s[mid-1 : mid+2]

```
new_str = ~~s[mid-1] + s[mm~~
          ~~(s[mid-1] : s[mid+2])~~
print(new_str
```

O/P Enter a string: JaSonAy

Son

3. Append new string in the middle of a given string. Given 2 strings, s1 and s2. Write a program to create a new string s3 by appending s2 in the middle of s1.

```
s1 = "Ault"
s2 = "Kelly"
~~s3 =~~
mid = len(s1)//2
s3 = (s1[:mid] + s2 + s1[mid:])
print(s3)
```

O/P AuKellylt