DAY-20

OCTOBER-06

DICTIONARY:

- Always values are stored in key:value pairs
- Here key is immutable and values are mutable
- We represent dictionary by dict() or {:}

Example:

pdata = {'name':'priyanka','ph no': '7349178176','place':'davanagere','email':'priyankabg99@gmail.com'}

pdata

o/p:

{'name': 'priyanka',

'ph no': '7349178176',

'place': 'davanagere',

'email': 'priyankabg99@gmail.com'}

- To print only keys - pdata.keys()
- To print only values – pdata.values()
- To change the values –
  pdata['ph no'] = '9807654321'
  pdata
  o/p:
  {'name': 'priyanka',
  'ph no': '9807654321',
  'place': 'davanagere',
  'email': 'priyankabg99@gmail.com'}
- To add new key value pair to the dictionary
  pdata['age'] = 26
  pdata
  o/p:
  {'name': 'priyanka',

'ph no': '9807654321',
'place': 'davanagere',
'email': 'priyankabg99@gmail.com',
'age': 26}

- To delete any value
  pdata.pop('email')
  To delete the last element or last key value pair go with pop item
  pdata.popitem()
  we can use delete function also.

Usage of dictionary:

- We use this in pandas library mostly in data cleaning, our data will mostly be 2 dimensional we cant directly create data instead we create a dictionary and change it into table in pandas.
- Dictionary can accept multiple values for the one key value.
  sdatas = {'sid':[1,2,3,4],'name':['a','b','c','d'],'class':['8th','10th','7th','10th']}
  sdatas
  o/p:
  {'sid': [1, 2, 3, 4],
  'name': ['a', 'b', 'c', 'd'],
  'class': ['8th', '10th', '7th', '10th']}

  sdatas = {'sid':(1,2,3,4),'name':('a','b','c','d'),'class':('8th','10th','7th','10th')}
  o/p:
  {'sid': (1, 2, 3, 4),
  'name': ('a', 'b', 'c', 'd'),
  'class': ('8th', '10th', '7th', '10th')}

String:

- String can hold a word or a sentence, hence considered as a sequence data type.
- String is immutable in structure.
- To add something to the string we should go with concatenation.

1. s = 'python@course$price&is!2350'

Count how many vowels,consonants,special characters and numbers are present in s

Code:

```
s = 'python@course$price&is!2350'
v = 0
c = 0
sp = 0
d = 0
for i in s:
    if i.isalpha():
        if i in 'aeiou':
            v += 1
        else:
            c += 1
    elif i.isdigit():
        d += 1
    else:
        sp += 1
print(f"Number of vowels:{v}\nNumber of consonants:{c}\nNumber of digits:{d}\nNumber of special characters:{sp}")
```

o/p:

Number of vowels:7

Number of consonants:12

Number of digits:4

Number of special characters:4

2. s1 = 'PyThOn StUdEnts"

Create seperate list for uppercase and lowercase characters.

Code:

```
s1 = 'PyThOn StUdEnts'
ucase = []
lcase = []
for i in s1:
    if i!=' ':
        if i.isupper():
         ucase.append(i)
        else:
         lcase.append(i)
print(f"Uppercase list:{ucase}\nLowercase list:{lcase}")
```

o/p:

Uppercase list:['P', 'T', 'O', 'S', 'U', 'E']

Lowercase list:['y', 'h', 'n', 't', 'd', 'n', 't', 's']

3. Create a string made of the first,middle and last character.

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

Code:

```
s = input("Enter a string: ")
p = len(s)//2
f = s[0]
```

```
l = s[-1]

m = s[p]

fs = f+m+l

print(fs)
```

o/p:

```
Enter a string:  Indhu

Idu
```

4. Create a string made of the middle three characters

Write a program to create a new string made of the middle three characters of an input string.

ex.JaSonAy-->Son

JhonDipPeta -->Dip

Code:

```
s = input("Enter a String:")

mid = int(len(s)/2)

new_string = s[mid-1:mid+2]

print(f"New String made of middle characters is: {new_string}")
```

o/p:

```
Enter a String: Indhu

New String made of middle characters is: ndh
```

5. Append new string in the middle of a given string

Given two 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"
```

expected-->AuKellylt

Code:

```
s1 = "Ault"
s2 = "Kelly"
mid = int(len(s)/2)
x = s1[:mid:]
x = x + s2
x = x + s1[mid:]
print("After appending new string in middle:", x)
```

o/p:

```
After appending new string in middle: AuKellylt
```