Data Science & AI
&
NIC - Param

Python-For Data Science

Dictionary

Lecture No.- 01

By- Pankaj Sharma Sir

Recap of Previous Lecture

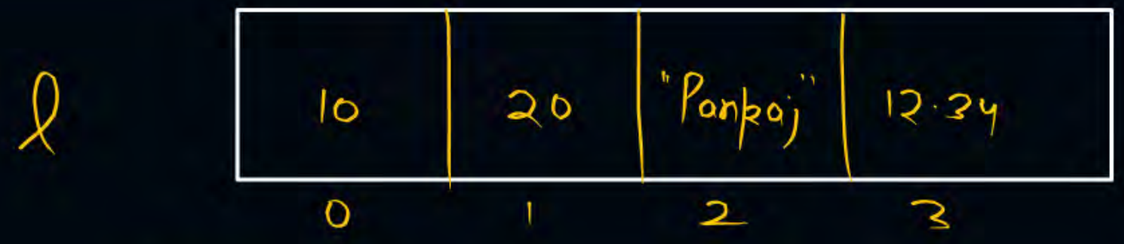
Topic Sets and Tuples

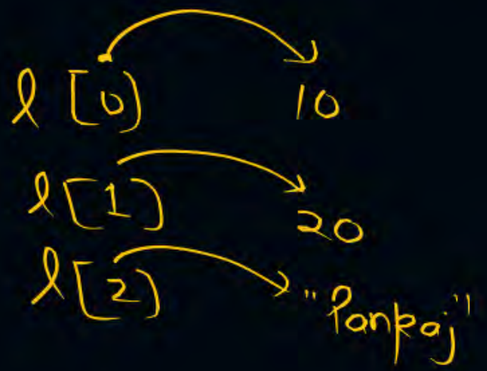
Topics to be Covered

Topic Dictionary

Topic : Dictionary

List





Problem solving

"Pankaj sharma is Pankaj sharma"

key : value

{ "Pankaj" : 2
"sharma" : 2
"is" : 1 }

unique

Key : value pairs

Roll : name

{ 1 : 'Amit' , 2 : 'Amit' , 3 : 'Anmol' }

1. # Empty dict

```
d = {}
```

2. d = { 1 : 'Amit' , 2 : 'Anita' }

3. d = { "Pankaj" : 3 , "sharma" : 2 , "is" : 1 }

d = { }

d[1] ⇒ KeyError

d[1] = 'Amit'

(key , value)

Coding ⇒

d = {1 : 'Amit', 2 : 'Pankaj'}

'Arun'

d[1] = 'Arun'

Problem 0

s = "Pankaj sir is Pankaj Sharma sir"

freq

wordskilist = s.split() ['Pankaj', 'sir', 'is', 'Pankaj', 'sharma', 'sir']

d = {}

for word in wordskilist :

d[word] = d[word] + 1

① word = 'Pankaj'

d['Pankaj']

= d['Pankaj'] + 1

KeyError

Problem 0

s = "Pankaj sir is Pankaj Sharma sir"

freq

wordskilist = s.split() ['Pankaj', 'sir', 'is', 'Pankaj', 'Sharma', 'sir']

d = {}

for word in wordskilist :

    if word in d :

        d[word] = d[word] + 1

    else

        d[word] = 1

a[i] = a[i]+1

⇒ a[i]++

No ++ operator

+3 ✓

+(+3) ✓

a = 2

a++ ⇒ Error

a = 3

++a ⇒ Error

```
s = input()
```

① Print freq. of each character

② freq. of vowels

```
a = 3
print(+++a)
```

```
s =     "Pankaj Sharma is good facult"
l =     list(s)
d = {}
for ele in l:
    if ele in 'aieouAEIOU':
        d[ele] = d.get(ele, 0) + 1
```

split()
list

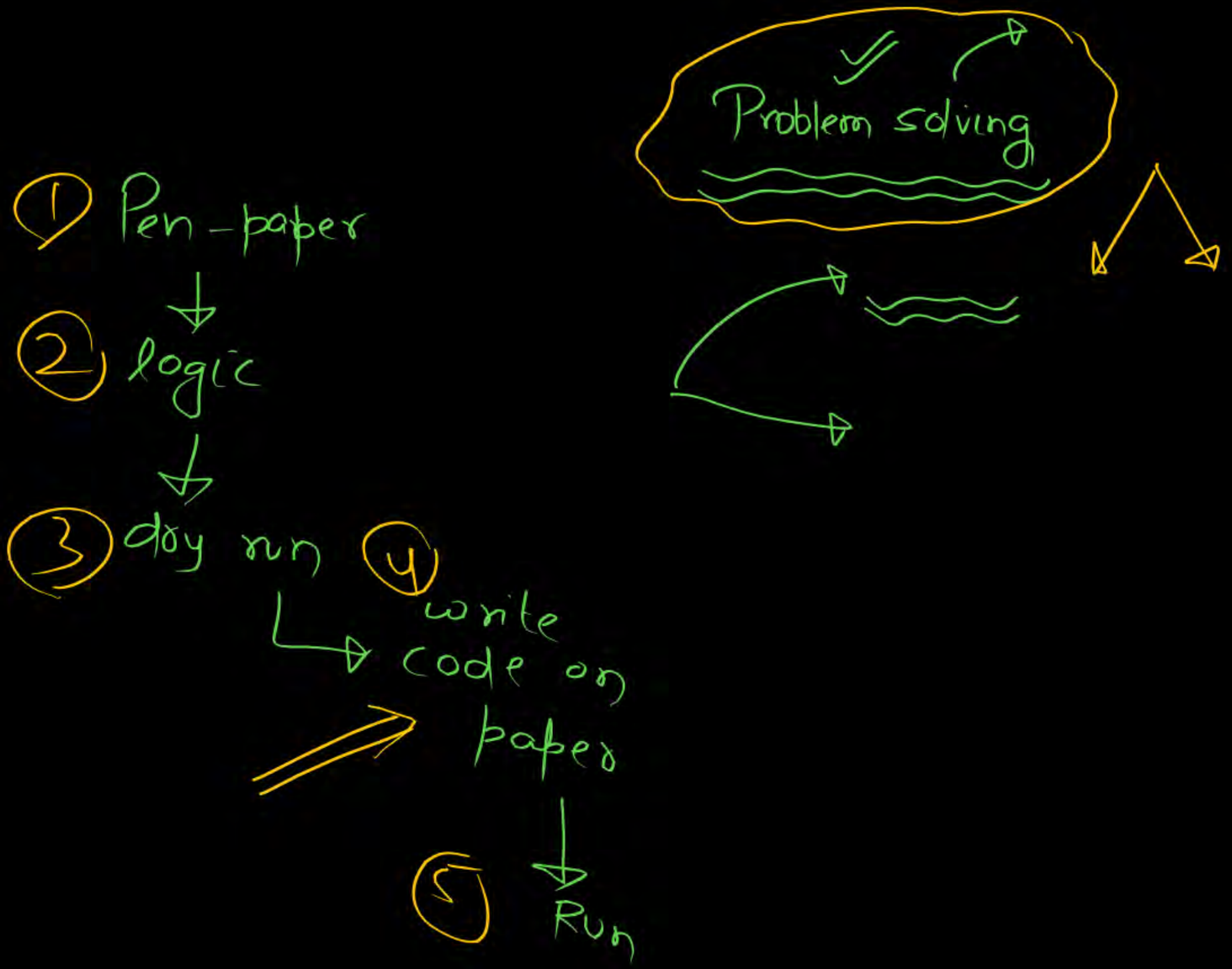
Given two sets

Q

S1
S2

} whether S2 is a subset of S1

```
for ele in S2:
    if ele not in S1:
        print("No")
        break
else:
    print("Yes")
```


Problem solving
1 Pen-paper
2 logic
3 dry run
4 write code on paper
5 Run

15th → 8th day

10th

12-14 → ds

→ OOPS

Complaint

Review

Algorithm

Chapter-1

→ complexity

→ Analysis

Day 15 dictionary

In [1]:
```
d={}
print(type(d))
```

```
<class 'dict'>
```

In [2]:
```
d={1:'amit',2:'anita',3:'anam'}
type(d)
```

Out[2]:
```
dict
```

In [3]:
```
d
```

Out[3]:
```
{1: 'amit', 2: 'anita', 3: 'anam'}
```

In [4]:
```
print(d)
```

```
{1: 'amit', 2: 'anita', 3: 'anam'}
```

In [8]:
```
(1,"pankaj")#pair of values
(2,"neeraj")#pair of values
(3,"pankaj")#pair of values
l=[(1,"pankaj"),(2,"neeraj"),(3,"pankaj")]#list of pairs
d=dict(l)#it will reate a dictionary
```

In [9]:
```
d
```

Out[9]:
```
{1: 'pankaj', 2: 'neeraj', 3: 'pankaj'}
```

In [10]:
```
#creating a dictionary by providing a list of keys
d=dict.fromkeys(["pankaj",12,34.5]) #i am providing 3 keys
#where are values for these keys ===>None
d
```

Out[10]:
```
{'pankaj': None, 12: None, 34.5: None}
```

In [11]:
```
d=dict.fromkeys(["pankaj",12,34.5]  , 0  ) #i am providing 3 keys
#we are providing default value 0 as value for each key
d
```

Out[11]:
```
{'pankaj': 0, 12: 0, 34.5: 0}
```

In [12]:
```
d={1:2,12.3:4,"pankaj":[1,2,3,4,"neeraj"],"neeraj":{1:34}}
d
```

Out[12]:
```
{1: 2, 12.3: 4, 'pankaj': [1, 2, 3, 4, 'neeraj'], 'neeraj': {1: 34}}
```

In [20]:
```
l=[(3,"pankaj"),(1,4),(1,"pankaj"),(1,"neeraj")]
```

In [21]:
```
d=dict(l)
```

In [22]:
```
d
```

Out[22]:
```
{3: 'pankaj', 1: 'neeraj'}
```

In [23]:
```
#accesing elements
d={1:2,12.3:4,"pankaj":[1,2,3,4,"neeraj"],"neeraj":{1:34}}
```

In [24]:
```
d[1] #retrieve value corresponding to key 1
```

Out[24]:
```
2
```

In [25]:
```
d[12.3]
```

Out[25]:
```
4
```

In [26]:
```
d["pankaj"]
```

Out[26]:
```
[1, 2, 3, 4, 'neeraj']
```

In [27]:
```
d[100]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[27], line 1
----> 1 d[100]

KeyError: 100
```

In [28]:
```
#another way to access elements
d
```

Out[28]:
```
{1: 2, 12.3: 4, 'pankaj': [1, 2, 3, 4, 'neeraj'], 'neeraj': {1: 34}}
```

In [29]:
```
d.get(1)
```

Out[29]:
```
2
```

In [30]:
```
d.get("pankaj")
```

Out[30]:
```
[1, 2, 3, 4, 'neeraj']
```

In [31]:
```
d.get(100)#no error but it returns None
```

In [32]:
```
print(d.get(100))
```

```
None
```

In [33]:
```
print(d.get(100,0))#if 100 as a key is present it will return the value for 100
#if 100 as a key is not present then it will return 0
```

```
0
```

In [34]:
```
d
```

Out[34]:
```
{1: 2, 12.3: 4, 'pankaj': [1, 2, 3, 4, 'neeraj'], 'neeraj': {1: 34}}
```

In [36]:
```
type(d.keys())
```

Out[36]:
```
dict_keys
```

In [37]:
```
d.values()
```

Out[37]:
```
dict_values([2, 4, [1, 2, 3, 4, 'neeraj'], {1: 34}])
```

In [38]:
```
d={1:'pankaj',2:'pankaj',3:'amit'}
d.values()
```

Out[38]:
```
dict_values(['pankaj', 'pankaj', 'amit'])
```

In [39]:
```
d.items()
```

Out[39]:
```
dict_items([(1, 'pankaj'), (2, 'pankaj'), (3, 'amit')])
```

In [40]:
```
#how to iterate on dict
for x in d: #x ==>iterateing on keys
    print(x)
```

```
1
2
3
```

In [41]:
```
for x in d: #x ==>iterateing on keys
    print(d[x])  #what is d[x] ===>value
```

```
pankaj
pankaj
amit
```

In [42]:
```
for x in d: #x ==>iterateing on keys
    print(x,d[x]) #key value
```

```
1 pankaj
2 pankaj
3 amit
```

In [43]:
```
a={}
a[1]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[43], line 2
      1 a={}
----> 2 a[1]

KeyError: 1
```

In [44]:
```
a[1]='amit'
```

In [45]:
```
a
```

Out[45]:
```
{1: 'amit'}
```

In [46]:
```
a[2]='pankaj'
```

In [47]:
```
a
```

Out[47]:
```
{1: 'amit', 2: 'pankaj'}
```

In [48]:
```
a[2]='neeraj'
```

In [49]:
```
a
```

Out[49]:
```
{1: 'amit', 2: 'neeraj'}
```

In [50]:
```
a={1:2,2:'pankaj',3:'neeraj',4:34.5}
b={1:'arun',3:89,5:12}
a.update(b)#for common key pair from b come to the a
```

In [51]:
```
b
```

Out[51]:
```
{1: 'arun', 3: 89, 5: 12}
```

In [52]:
```
a
```

Out[52]:
```
{1: 'arun', 2: 'pankaj', 3: 89, 4: 34.5, 5: 12}
```

In [53]:
```
a={1:2,2:'pankaj',3:'neeraj',4:34.5}
b={1:'arun',3:89,5:12}
b.update(a)#for common pair ==>pair in a comes to b
#(1:2),(3:'neeraj') comes to b
#(5:12) already in b
```

In [54]:
```
b
```

Out[54]:
```
{1: 2, 3: 'neeraj', 5: 12, 2: 'pankaj', 4: 34.5}
```

In [56]:
```
s=input("enter a string")
wordskilist=s.split()
d={}
for word in wordskilist :
    if word in d:
        d[word] =d[word]+1
    else:
        d[word]=1
print(d)
```

```
enter a stringpankaj sharma is pankaj sharma
{'pankaj': 2, 'sharma': 2, 'is': 1}
```

In [58]:
```
s=input("enter a string")
wordskilist=s.split()
d={}
for word in wordskilist :
    d[word]=d.get(word,0) + 1
print(d)
```

```
enter a stringpankaj sharma is pankaj sharma
{'pankaj': 2, 'sharma': 2, 'is': 1}
```

In [59]:
```
#duplicates keys not allowed
#heterogen objext are allowed for both keys and values
#it is not mandatory that all keys are of same type
#it is not mandatory that all values are of same type
#insertion order is not preserved
#mutable
#dynamic ==>add,remove
#indexing and slicing concept is not there in case of dict
```

In [60]:
```
d={1:2,3:4}
print(1 in d)
```

```
True
```

In [61]:
```
d={1:'amit',2:'pankaj',3:12.34}
del d[1]
```

In [62]:
```
d
```

Out[62]:
```
{2: 'pankaj', 3: 12.34}
```

In [63]:
```
del d[100]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[63], line 1
----> 1 del d[100]

KeyError: 100
```

In [64]:
```
a={1:'amir',2:'amit',3:'pankaj'}
del a
```

In [65]:
```
a
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[65], line 1
----> 1 a

NameError: name 'a' is not defined
```

In [66]:
```
a={1:'amir',2:'amit',3:'pankaj'}
a.clear()
```

In [67]:
```
a
```

Out[67]:
```
{}
```

In [68]:
```
#len()
#get()
#clear()
#dict()
#pop() ===>specific element ko remove
a={1:'amir',2:'amit',3:'pankaj'}
```

```
a.pop(1) #remove the corresponding pair associated with key as 1
#also return value 'amir'
```

Out[68]:
```
'amir'
```

In [69]:
```
a={1:'amir',2:'amit',3:'pankaj'}
a.popitem()  #randomly kisi ko bhi remove krega
```

Out[69]:
```
(3, 'pankaj')
```

In [70]:
```
# y.update(x)
#all the items(key,value) pairs of dict x will be added to dict y
d=3
print(+++++d)
```

```
3
```

In [71]:
```
d=3
print(d++)
```

```
  Cell In[71], line 2
    print(d++)
              ^
SyntaxError: invalid syntax
```

In [ ]:
```

```

THANK - YOU