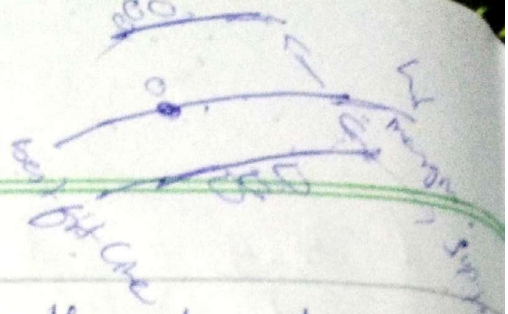

4, ⇒ remove ( ) ?

it is also the use for dlete the element from the sets, as well as discard work is same

ex:-

s = {10, 20, 30}

s.remove (20) → through value

print (s) output

{10, 30}

3 Types of A
1) Weak
2) Generative
3) Super AI.

5, discard ( )

it is similar same as remove.

s = {10, 20, 30}

s.discard(20)

print (s)

{10, 30}

6 Clear( ): it is used for the empty the data

ex:-

s = {10, 20, 30}

s.clear( ) print (s)

it will clear the set and answer will

set( )

⑦ update ( )

s = {10, 20, 30}

L = [30, 40, 60] → list

s.update ( ) → set

print (s)

{10, 20, 30, 40, 60}

Note:- it will not repeat the 30.

=> functions of sets :-

1. set( )
2. add( )
3. pop( )
4. remove( )
5. clear( )
6. discard( )
7. update.

3 => pop( )

s = {10, 20, 30}

s.pop( )

print(s)

so it will randomly delete any element

ex:- 30 delete.

output

{10, 20}.

Note: pop function also return the value so if we print it like.

print(s.pop( ))

it will give out put what it delete.

1 => set( ) ?

L = [10, 20, 30] this is a list and assume that we have to convert into the set.

then we will use set( ) function for converting data of list to set.

sol

L = [10, 20, 30]

s = set(L)

print(s)

{10, 20, 30}

and same we can convert tupple to sets.

2 add( ) is is used for add the new element inside the sets

s = {10, 20, 30}

s.add(40)

print(s).

output

{10, 20, 30, 40}

-> note set can provide the answer randomly also like 10 or 20 may be come first

=> Dictionary Methods:

myDict.key() # returns all keys in collection forms

myDict.values() # returns all values.

=> Sets:- it is the unordered data type, and it is unindex. means it will not get using indexing numbers. Note that inside the set data can be dlete but we are not allow to change it. it use curley brackets {}.

set can be represent set() function. or {} -> it also use in dictionary but in dictionary we use the key value pairs. but inside the sets it not exist. it hold the uniqye data.

S= {10, 20, 30}. for print values we have to iterate it by using loop.

```
for a in S:
    print(a):
```

10
20
30

⇒ Nasted Dict: asan keh b jika key ji value khai dict thae shadeenda ayo.

```
student = {
    "name" : "shazil",
    "score" : {
        "chem" : 98,
        "phy" : 97
        "math" : 95
    }
}
```

it is called nasted dict

print (student) -> it will print all data.
but for printing only score of students

```
print ( student ["score"] )
```

and if we want to see just chem marks then

~~print (student ["score"])~~

```
print ( student ["score"] ["chem"] )
```

-> but key can be tupple, can be number, can be decimal.

Note:- Dict is mutable:-

-> Note: There is no idex concept in Dict.

=> for accessing value:-

```
print (info["name"])
```

-> for add value:-

```
info["name"] = "shrad" (// it will overwrit prev val
print (info)
```

-> for adding totally new key value pair.

```
info["surname"] = "khapra"
print (info)
```

so that's why it is called mutable.

=> empty Dict:

```
null-dict = { }
print (null-dict)
```

for add value in null

-> `null-dict["name"] = "nood"`

```
print (null-dict)
```

=> Dictionary:- Dictionaries are used to store data value in key:value pairs. They are unordered, mutable (changeable) and dont allow duplicate keys it store value in word <-> meaning

ex:- word is like a key.
Meaning is like a value.
so if we want search any value the we will search word.
so if we have key value type information then we will add/use Dictionary:- in python.

ex:-

```
dict = {
"name" : "shradha",
"cgpa" : 9.6
"marks" : [98, 97, 95],

}
print.(dict)
```

→ note:- in values all the other data types are allow

- ex:- info={

```
"subject" : ["python", "c", "java"],
"topics" : ("dict", "set"),
"age" : 35.
}
```

print (info)

Note:- keys can not be List. is mutable