17-02-2025

Set Operations:

1.create 5 sets

set1={1,"hello",2,3,20}

print(set1)

print(type(set1))

set1.add(45)

print(set1)

set1.remove('hello')

print(set1)

output:

{1, 2, 3, 'hello', 20}

<class 'set'>

{1, 2, 3, 'hello', 20, 45}

{1, 2, 3, 20, 45}

#set2

set2={10,20,30,40}

print(type(set1))

set2.add(50)

set2.remove(10)

print(set2)

output:

<class 'set'>

{40, 50, 20, 30}

#set3

set3={"hello","world","python","is","a"}

print(type(set3))

set3.add("programming language")

set3.remove("world")

print(set3)<class 'set'>

{40, 50, 20, 30}

Output:

<class 'set'>

{'hello', 'a', 'is', 'programming language', 'python'}

#set4

set4={1,True,2,0,False,30}

print(type(set4))

set4.add(10)

set4.remove(0)

print(set4)

output:

<class 'set'>

{1, 2, 10, 30}

#set5

set5={1,2,3,True,False}

print(set5)

print(type(set5))

output:

{False, 1, 2, 3}

<class 'set'>

2.Perform of union, intersection and difference among them.

set1={10,20,30,40}

set2={20,30,50,60}

print(set1.union(set2))

print(set1.intersection(set2))

print(set1.difference(set2))

output:

{40, 10, 50, 20, 60, 30}

{20, 30}

{40, 10}

3.write a python program to print even numbers

num=4

if num%2==0:

  print("the given number is even number")

output:

the given number is even number

4.write a program to print odd numbers

num=4

if num%2!=0:

  print("the given number is odd number")

output:

the given number is odd number

5. #write a program to print whether the given number is even or odd

num=4

if num%2==0:

  print("the given number is even number")

else:

  print("the given number is odd number")

output:

the given number is even number

6.write a program to print only the capital letters in a list.

list1=["Hello World","Hey"]

capital\_letters = []

for word in list1:

    for char in word:

        if char.isupper():

            capital\_letters.append(char)

print(capital\_letters)

print("".join(capital\_letters))

output:

['H', 'W', 'H']

HWH

7.Take 2 lists and print the common elements in them.

list\_1=[10,20,30,40]

list\_2=[30,40,50,60]

common\_elements=[]

for i in list\_1:

  if i in list\_2:

    common\_elements.append(i)

print(common\_elements)

output:

[30, 40] or

list\_1=[10,20,30,40]

list\_2=[30,40,50,60]

common\_elements=[]

for i in list\_1:

  for j in list\_2:

    if i==j:

      common\_elements.append(i)

print(common\_elements)

8.take 2 lists and print the unique elements in them.

list\_1=[10,20,30,40]

list\_2=[30,40,50,60]

unique\_elements=[]

for i in list\_1:

  if i not in list\_2:

    unique\_elements.append(i)

print(unique\_elements)

output:

[10,20]

9.take 2 tuples and print the common elements in them.

tuple\_1=("hello","world","python","is","a","programming","language")

tuple\_2=("python","is","a","interpretered","language")

common\_elements=[]

for i in tuple\_1:

  if i in tuple\_2:

    common\_elements.append(i)

print(tuple(common\_elements))

output:

('python', 'is', 'a', 'language')

10.take 2 tuples and print the unique elements in them.

tuple\_1=("hello","world","python","is","a","programming","language")

tuple\_2=("python","is","a","interpretered","language")

unique\_elements=[]

for i in tuple\_1:

  if i not in tuple\_2:

    unique\_elements.append(i)

print(tuple(unique\_elements))

output: ('hello', 'world', 'programming')

11.take 2 dictionaries and print the common elements in them.

dict\_1={1:"hello",2:"world",3:"python",4:"is",5:"a",6:"programming",7:"language"}

dict\_2={1:"python",2:"is",3:"a",4:"interpretered",5:"language"}

common\_elements=set()

for i in dict\_1.values():

  if i in dict\_2.values():

    common\_elements.add(i)

print(common\_elements)

output:

{'language', 'a', 'is', 'python'}

12.take 2 dictionaries and print the unique elements in them.

dict\_1={1:"hello",2:"world",3:"python",4:"is",5:"a",6:"programming",7:"language"}

dict\_2={1:"python",2:"is",3:"a",4:"interpretered",5:"language"}

unique\_elements=set()

for i in dict\_1.values():

  if i not in dict\_2.values():

    unique\_elements.add(i)

print(unique\_elements)

output:

{'world', 'hello', 'programming'}

13.create a tuple

1. Covert this tuple to list
2. Revert back list to tuple

tuple1=(1,2,3,4,5)  
tuple2=list(tuple1)  
tuple2.insert(5,6)  
print(tuple(tuple2))

14.delete the tuple

tuple2.clear()  
print(tuple(tuple2))

15. difference between list, tuple, set and dictionary

|  |  |  |  |
| --- | --- | --- | --- |
| List | Tuple | Set | Dictionary |
| 1. A list is an ordered, mutable collection of elements. | 1. A tuple is an ordered, immutable collection of elements. | 1. A set is an unordered collection of unique elements. | 1. A dictionary is an ordered collection of key-value pairs. |
| 2.Duplicate data entry is allowed in a List. | 2. Duplicate data entry is allowed in a Tuple. | 2. All elements are unique in a Set. | 2. Keys are unique, but two different keys can have the same value. |
| 3.The list can be represented by [ ]. | 3.The tuple can be represented by (). | 3.The set can be represented by {}. | 3.The dictionary can be represented by {}. |
| 4.A list can be created using the list() function. | 4.A tuple can be created using the tuple() function. | 4.A set can be created using the set() function. | 4.A dictionary can be created using the dictionary() function. |
| 5.List items are indexed and we can access them by referring to the index number and indexing starts from ‘0’.  Ex: list1 = [1, 2, 3,4] print(list1[2])  o/p: 3 | 5.Tuple items are also has indexing that starts from ‘0’.  Ex: tuple1=(1,2,4,6)  Print(tuple1[0])  o/p: 1 | 5. Set does not have an index based mechanism. | 5. Dictionary has a Key based indexing i.e. keys identify the value.  Ex: dict1 = {"one": 1, "two": 2, "three": 3}  print(dict1['two'])  o/p: 2 |
| 6. In list, new items can be added using the append() method. | 6. Being immutable, new data cannot be added to it. | 6. The add() method adds an element to a set. | 6. update() method updates specific key-value pair. |
| 7. The pop() method allows deleting an element. | 7. Being immutable, no data can be popped/deleted. | 7. Elements can be removed by using remove() method. | 7. The pop(key) removes specified key along with its value. |
| 8. sort() method sorts the elements. | 8. Immutable, so sorting method is not applicable. | 8. Unordered, so sorting is not advised. | 8. Keys are sorted by using the sorted() method. |
| 9. It is used for general-purpose collection. | 9. Fixed collection of items. | 9. Unique items, mathematical operations. | 9. Key-value mappings purpose |
| Ex: list1 = [1 , 2, 'abc', 3, 'def’]  Print(list1)  o/p: [1,2,’abc’,3,’def’] | Ex: tuple1 = (1, 2, 'abc', 3, 4)  print(tuple1)  o/p:  (1, 2, 'abc', 3, 4) | Ex: set1 = {1, 2, 3, 'abc', 6}  print(set1)  o/p:  {'abc', 1, 2, 3, 6} | Ex: dict1 = {"key1": "value1", "key2": "value2"}  print(dict1)  o/p:{‘key1': 'value1', ‘key2': 'value2'} |