30/09/2025

Removing all occurrence of item 20

Using while loop

```
list1=[5,20,15,20,25,50,20]
while 20 in list1:
    list1.remove(20)
list1
```

Using for loop

```
list1=[5,20,15,20,25,50,20]
for i in list1:
    if i==20:
        list1.remove(20)
list1
```

o/p:

[5, 15, 25, 50]

Removing duplicates from list

```
l1=[1,1,2,2,2,3,4,5,5,6,7,7,8,8,8]
l1=list(set(l1))
l1
```

or

```
l1=[1,1,2,2,2,3,4,5,5,6,7,7,8,8,8]
uni=[]
for i in l1:
    if i not in uni:
```

```
    uni.append(i)
print(uni)
```

o/p:

[1, 2, 3, 4, 5, 6, 7, 8]

Tuple:

A Tuple in Python is an ordered collection of elements, similar to a list, but immutable meaning its elements cannot be changed after creation.

Tuples are used to store fixed collections of data that should not be modified.

Properties of Tuple

- **Ordered** – Maintains insertion order
- **Immutable** – Cannot be changed (no add/remove/update)
- **Allows Duplicates** – Yes
- **Heterogeneous** – Can store different data types
- **Indexing** – Supported
- **Slicing** – Supported

There are two ways to represented by

1. tuple()
2. ()

Creating a Tuple

```
t1=(34,'python',12.9,True,3+9j,12.9,'python')
t1
```

o/p:

(34, 'python', 12.9, True, (3+9j), 12.9, 'python')

Iterating through a tuple

```
for i in enumerate(t1):
    print(i)
```

o/p:

(0, 34)

(1, 'python')

(2, 12.9)

(3, True)

(4, (3+9j))

(5, 12.9)

(6, 'python')

Operations like append, insert, pop, clear, and sort cannot be performed on a tuple.

Sort a tuple of tuples by 2nd item

```
tuple1=(('a',23),('b',37),('c',11),('d',29))

t1=tuple(list(sorted(tuple1,key=lambda x:x[1])))

t1
```

o/p:

(('c', 11), ('a', 23), ('d', 29), ('b', 37))

Printing positions and values

```
for i in tuple1:
    print(i)
```

o/p:

('a', 23)

('b', 37)

('c', 11)

('d', 29)

Set:

A Set in Python is an unordered collection of unique elements.

It is used to store non-duplicate data and supports mathematical set operations such as union, intersection, and difference.

Properties of Set:

- Unordered → No index or slicing.
- Mutable → Can add or remove elements.
- No duplicates → Automatically removes repeated values.
- Heterogeneous → Can store mixed data types.

There are two ways to represent set:

1. set{}
2. {}

Creating and displaying a set

```
s1={12,'latha',11.7,True,'bcd',12}

s1
```

o/p:

{11.7, 12, True, 'bcd', 'latha'}

Adding elements

```
s1.add(4)

s1
```

o/p:

{11.7, 12, 4, True, 'bcd', 'latha'}

Deleting elements

```
s1.pop()
```

o/p:

True

Removing elements:

s1.remove(4)

o/p:

{11.7, 12, 'bcd', 'latha'}

Note: remove() gives an error if the element doesn't exist.

By using discard(), can avoid errors:

s1.discard(4)

Mathematical set operations

set1={1,2,3,4,5}

set2={4,5,6,7,8}

print(set1.union(set2))

print(set1.intersection(set2))

print(set1.difference(set2))

print(set2.difference(set1))

print(set1.symmetric_difference(set2))

o/p:

{1, 2, 3, 4, 5, 6, 7, 8}

{4, 5}

{1, 2, 3}

{8, 6, 7}

{1, 2, 3, 6, 7, 8}