26/09/2025

Data Structure:

Python data structures are ways of organizing and storing data so that they can be accessed and modified efficiently.

- list: []
- tuple:()
- set:{}
- dict:{key:value}

Python provides both built-in data structures and allows us to implement user-defined data structures.

Taking multiple inputs as strings

```
value=input("enter some values").split()
```

o/p: enter some values tanu 22 15.5

```
value
```

o/p: ['tanu', '22', '15.5']

- .split() → splits it into a list of words/numbers separated by spaces.

Taking multiple numeric inputs

```
num=list(map(int,input("enter a value:").split()))
```

o/p:

```
enter a value: 2 3 4 5 6
```

```
num
```

o/p: [2, 3, 4, 5, 6]

- input().split() → takes multiple inputs separated by space.
- map(int, ...) → converts each item from string to integer.
- list() → stores them into a list.

List [ ]

A **List** in Python is a **heterogeneous data structure** that can store **different types of data** (string, integer, float, boolean, complex numbers, etc.).

Properties of List:

- Ordered → maintains insertion order
- Mutable → can be modified (add, remove, change elements)
- Allows duplicates
- Can store heterogeneous elements

There are two ways to represent:

- list()
- [ ]

Creating a list with mixed data types

```
details=["tanu","15-05-2003",10.8,True,9686,4+7j]

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, True, 9686, (4+7j)]
```

Adding an element using append()

```
details.append("Data analyst")

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, True, 9686, (4+7j), 'Data analyst']
```

- append() adds a new element at the end of the list.

Modifying an element by index

```
details[3]=False

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, False, 9686, (4+7j), 'Data analyst']
```

- Indexing starts from 0.
- Changes the 4th element (True) to False.

Removing the last element using pop()

```
details.pop()

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, False, 9686, (4+7j)]
```

- pop() removes the last element from the list by default.

Appending a number

```
details.append(5.8)

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, False, 9686, (4+7j), 5.8]
```

Iterating through the list

```
for i in details:

    print(i)
```

o/p:

```
tanu

15-05-2003

10.8

False

9686
```

```
(4+7j)

5.8
```

Using enumerate() for index and value

```
for i in enumerate(details):

   print(i)
```

o/p:

```
(0, 'tanu')

(1, '15-05-2003')

(2, 10.8)

(3, False)

(4, 9686)

(5, (4+7j))

(6, 5.8)
```

- enumerate() returns both index and value as a tuple.

Appending a nested list

```
details.append([1,2,3])

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, False, 9686, (4+7j), 5.8, [1, 2, 3]]
```

- Adds another list as a single element inside the main list.

Removing the nested list

```
details.pop()
```

o/p:

```
[1,2,3]
```

Creating a new list

```
l2=[1,2,3]

l2
```

o/p:

```
[1, 2, 3]
```

Concatenating two lists

```
details=details+l2

details
```

o/p:

```
['tanu', '15-05-2003', 10.8, False, 9686, (4+7j), 5.8, 1, 2, 3]
```

- The + operator joins two lists into one.
- Elements of l2 are added to the end of details.

Updating a value in the list

```
details[1]="9786575643"

details
```

o/p:

```
['tanu', '9786575643', 10.8, False, 9686, (4+7j), 5.8, 1, 2, 3]
```

- Changes the second element (index 1) to a new value — "9786575643".

Inserting an element at a specific position

```
details.insert(2,"15-05-2003")

details
```

o/p:

```
['tanu', '9786575643', '15-05-2003', 10.8, False, 9686, (4+7j), 5.8, 1, 2, 3]
```

- .insert(position, value) adds a new value at the specified index (here index 2).

Deleting an element using index

```
details.pop(4)

details
```

o/p:

```
['tanu', '9786575643', '15-05-2003', 10.8, 9686, (4+7j), 5.8, 1, 2, 3]
```

- .pop(index) removes the element at a specific index (here index 4).

Deleting an element by value

```
details.remove("tanu")

details
```

o/p:

```
['9786575643', '15-05-2003', 10.8, 9686, (4+7j), 5.8, 1, 2, 3]
```

- .remove(value) deletes the first matching element from the list.

Deletes Elements, Not the List

```
details.clear()

details
```

Deletes the Entire List

```
del details
```

Create square number list from given list number.

```
l1 = [5, 2, 9, 7, 1]

l2=[]

for i in l1:

   s=i*i

   l2.append(s)

l2
```

```
o/p:

[25, 4, 81, 49, 1]
```

Create even,odd,prime number list from 1 to 20

```
even=[]

odd=[]

prime=[]

for i in range(1,21,1):

    if i%2==0:

        even.append(i)

    else:

        odd.append(i)

    for j in range(2,i,1):

        if i%j==0:

            break

    else:

        prime.append(i)

print("even:",even)

print("odd:",odd)

print("prime:",prime)
```

```
o/p:

even: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

odd: [1, 3, 5, 7, 9, 11, 13, 15, 17, 19]

prime: [1, 2, 3, 5, 7, 11, 13, 17, 19]
```