In [3]:
```
# QUE 1:> Discuss string slicing and provide examples.

# ANS:> It is methhod of filding the range of nombers or characters present in any string by start index and the end index.bt default it start from 0 index.

a = ("1", "b", "c", "4")
num = slice(3)
print(a[num])
```

('1', 'b', 'c')

In [11]:
```
a = ("1", "b", "c", "4")
num = slice(2,4)
print(a[num])
```

('c', '4')

In [13]:
```
# QUE 2:> Explain the key features of lists in Python.

# ANS:> list is a container of hetrogenious data we can store multiple data in it like variables,strings.

a = ["1", "2", "3"]
b = ["10", "11", "12"]
c = ["anil", "50", "robin", "30"]
a[1: ]
b[2: ]
```

Out[13]: ['12']

In [15]:
```
a[1: ]
```

Out[15]: ['2', '3']

In [16]:
```
# to add elements:-

c.append("laxman")
c
```

Out[16]: ['anil', '50', 'robin', '30', 'laxman']

In [17]:
```
# to remove elements:- it remove the last index present in the list.

c.pop()
c
```

Out[17]: ['anil', '50', 'robin', '30']

In [28]:
```
# to insert at a specific location/index.

c.insert(2,"mausam")
c
# i ran it two times thats why its like this.
```

Out[28]: ['anil', '50', 'mausam', 'mausam', 'robin', '30']

In [32]:
```
# to coutn the list elements

c
```

Out[32]: ['anil', '50', 'mausam', 'mausam', 'robin', '30']

In [33]:
```
c.count("mausam")
```

Out[33]: 2

In [ ]:
```

```

In [ ]:
```

```

In [21]:
```
#QUE  Describe how to access, modify, and delete elements in a list with examples.

# ANS:> These elements are easy to use because list works on the strings.
#to acces the list:

a = ["apple", "grapes", "banana", "guava"]
print(a[1])
```

grapes

In [22]:
```
print(a)
```

['apple', 'grapes', 'banana', 'guava']

In [23]:
```
# to modify the list:

a[1] = "pineapple"
a
```

Out[23]: ['apple', 'pineapple', 'banana', 'guava']

In [24]:
```
# to delete the elements from a list:-

a.remove("banana")
a
```

Out[24]: ['apple', 'pineapple', 'guava']

In [35]:
```
# QUE 4:>  Compare and contrast tuples and lists with examples.

# ANS:> The main difference in between the tuples and the list is the tuples are immutables in nature once created can not be changed if we want to cahnge tuple it will show error so we have
        # list are mutable in nature so we can modify them according to our need and delete the items.

#tuple:-
point = (3,7,4,9,34)
type(point)
```

Out[35]: tuple

In [36]:
```
point[3] = "6"]
```

```
  Cell In[36], line 1
    point[3] = "6"]
                  ^
SyntaxError: unmatched '}'
```

In [37]:
```
point2 = (4,5,2,37)
point2
```

Out[37]: (4, 5, 2, 37)

In [44]:
```
point3 = (66,7)
point3 = (point,point2,point3)
point3
# by mistake point2 was ran/executed by me several times.
```

Out[44]:
```
((3, 7, 4, 9, 34),
 ((((4, 5, 2, 37), (3, 7, 4, 9, 34)), (3, 7, 4, 9, 34)), (3, 7, 4, 9, 34)),
  (3, 7, 4, 9, 34)),
 (66, 7))
```

In [45]:
```
# QUE 5:>  Describe the key features of sets and provide examples of their use.

# ANS:>   this is a data type which consists of unique elemnts.

a = [1,2,3,4,2,3,6,4,5,6,5,3,3,4,5,9,12,2,1,10]
num = set(a)
num
```

Out[45]: {1, 2, 3, 4, 5, 6, 9, 10, 12}

In [46]:
```
# TO add element:-
num.add(13)
num
```

Out[46]: {1, 2, 3, 4, 5, 6, 9, 10, 12, 13}

In [49]:
```
# to remove:-
# we can use both remove and pop also

num.remove(5)
num
# we cam remove one element at one time.
```

Out[49]: {1, 2, 3, 4, 6, 9, 10, 12, 13}

In [61]:
```
#types:- 1=> intersection
b = [2,6,8,0,11,17]
num2 = set(b)
num.intersection(num2)
```

Out[61]: {2, 6}

In [62]:
```
# type 2:> union

num.union(num2)
```

Out[62]: {0, 1, 2, 3, 4, 6, 8, 9, 10, 11, 12, 13, 17}

In [64]:
```
#type 3:=difference


num.difference(num2)
```

Out[64]: {1, 3, 4, 9, 10, 12, 13}

In [1]:
```
# QUE 6:> Discuss the use cases of tuples and sets in Python programming.

# ANS:> # Tuples are used where you do not modify the data, EX. uan no..

a = (2409,1209,1326,2626)
```

In [2]:
```
a
```

Out[2]: (2409, 1209, 1326, 2626)

In [3]:
```
a[2] = 2945
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[3], line 1
----> 1 a[2] = 2945

TypeError: 'tuple' object does not support item assignment
```

In [14]:
```
# stes are used as used in list which helps us to find duplicate values of data which we don't have to do or need.

p = (1,2,7,6,2, "robin", "ashu", "robin",2)
p
```

Out[14]: {1, 2, 6, 7, 'ashu', 'robin'}

In [13]:
```
type(p)
```

Out[13]: set

In [5]:
```
# QUE 7:>Describe how to add, modify, and delete items in a dictionary with examples.

#ANS:> To add the elements:we have to give {} brackets to understand that this is a dict and in this we have to undertand that after every ":" this sign data values present which van be chan

a = {"name" : "robin" , "age" : "24"}
a
```

Out[5]: {'name': 'robin', 'age': '24'}

In [14]:
```
# we can also add a new key in the dictionary.

a = {"name" : "robin" , "age" : "24"}
a
```

Out[14]: {'name': 'robin', 'age': '24'}

In [18]:
```
a["hobby"] = "playing football"
a
```

Out[18]: {'name': 'robin', 'age': '24', 'hobby': 'playing football'}

In [20]:
```
# to modify the dictionary:

#in this we acces the data with the keys only.
a["name"]
a["name"] = "rohit"
a
```

Out[20]: {'name': 'rohit', 'age': '24', 'hobby': 'playing football'}

In [21]:
```
# to delete the items:

del a["age"]
a
```

Out[21]: {'name': 'rohit', 'hobby': 'playing football'}

In [22]:
```
# it can also be done with the pop and remove also
#pop:>

a.pop("hobby")
a
```

Out[22]: {'name': 'rohit'}

In [9]:
```
# QUE 8:>Discuss the importance of dictionary keys being immutable and provide examples

#ANS:> dictionary keys are immutables because of the nature that we difine some values or name or anything,we assign that keys will go for every time but the data can be very like UAN no. or
# keys are made to difine the process/product knowledge which can't be change or accesed.we can not acces the keys it will show us the keyerror.

a = {"name" : "robin" , "age" : "24"}
a[24]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[9], line 7
      1 # QUE 8:>Discuss the importance of dictionary keys being immutable and provide examples
      2
      3 #ANS:> dictionary keys are immutables because of the nature that we difine some values or name or anything,we assign that keys will go for every time but the data can be very like UAN
no. or we can take any documents like adhar card that the name written there will always be the key but data value changes person to person.
      4 # keys are made to difine the process/product knowledge which can't be change or accesed.
      5 a = {"name" : "robin" , "age" : "24"}
----> 7 a[24]
```

```
KeyError: 24
```

In [ ]: