data structures ----> data storing containers

1.lists --->mutable-->changeable-->can be deleted elemnets[] 2.tuple ---->immutable() 3.distionaries--- mutable{} 4.sets --->immutable{} 5.Strings----> immuatble "Anwar", "Text"

1.lists: data structures ----> data storing containers 1.lists --->mutable-->changeable-->can be deleted elemnets 2.tuple ---->immutable 3.distionaries--- mutable 4.sets --->mutable 5.Strings----> immuatble

1.lists: lst=[1,'str',3] --->0,1,2

#how to access elements

#slicing can be done using indexing print(lst[0]) print(lst[0:2]) print(lst[-1]) print(lst[-3:-1])

```
lst=[1,'str',3,11] # --->0,1,2 square bracket
#how to access elements
#slicing can be done using indexing
# In posetive indexing start by zero, and by default last(element)
eliminate,
# In negative indexing starty by -1, and by default -1(element)
eliminate,
# we put element in list with [] square bracket

lst[0] = 13
print(lst[0])
print(lst[1])
print(lst[0:3])
print(lst[-1])
print(lst[-3:-1])
print(type(lst))
```

```
13
str
[13, 'str', 3]
11
['str', 3]
<class 'list'>
```

```
Tuple=(1,'str',3) # --->0,1,2
#how to access elements
#slicing can be done using indexing
print(Tuple[0])
print(Tuple[0:2])
print(Tuple[-1])
print(Tuple[-3:-1])
```

```
1
(1, 'str')
```

```
3
(1, 'str')
```

```
Tuple=(1,'str',3) # --->0,1,2
#how to access elements
#slicing can be done using indexing
Tuple[1] = 2
print(Tuple[0])
print(Tuple[0:2])
print(Tuple[-1])
print(Tuple[-3:-1])
```

```
-----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[7], line 4
      1 Tuple=(1,'str',3) # --->0,1,2
      2 #how to access elements
      3 #slicing can be done using indexing
----> 4 Tuple[1] = 2
      5 print(Tuple[0])
      6 print(Tuple[0:2])

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

```
string= '1str3' # --->0,1,2
#how to access elements
#slicing can be done using indexing

print(string[0])
print(string[0:2])
print(string[-1])
print(string[-3:-1])
```

```
1
1s
3
tr
```

```
string= '1str3' # --->0,1,2
#how to access elements
#slicing can be done using indexing
string[1] = 2
print(string[0])
print(string[0:2])
print(string[-1])
print(string[-3:-1])
```

```
-----------------------------------------------------------------------
-----
```

```
TypeError                                 Traceback (most recent call
last)
Cell In[10], line 4
      1 string= '1str3' # --->0,1,2
      2 #how to access elements
      3 #slicing can be done using indexing
----> 4 string[1] = 2
      5 print(string[0])
      6 print(string[0:2])

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

```
dic= {0:"Anwar",1:"2",2: "Miss Saba"} # --->0,1,2
#how to access elements
#slicing can not be done by using indexing
for i in dic:                          # for loop use for iterating the
elements.
    print(dic[i])
```

```
Anwar
2
Miss Saba
```

```
dic= {0:"Anwar",11:"2",2: "Miss Saba"} # --->0,1,2  The value of key
in dictionary
#how to access elements
#slicing can not be done by using indexing
for i in dic:                          # for loop use for iterating the
elements.
    print(i,dic[i])
```

```
0 Anwar
11 2
2 Miss Saba
```

```
Set = {0,"Anwar",1,2,2, "Miss Saba","karachi"} # --->0,1,2
#how to access elements,Set is immutable
#slicing can not be perform while doing operation of Set in Python.
for i in Set:                          # for loop use for iterating the
elements.
    print(i)
```

```
0
1
2
karachi
Miss Saba
Anwar
```

```
Set = {0,"Anwar",1,2,2, "Miss Saba","karachi"} # --->0,1,2
#how to access elements,Set is immutable
```

```
#slicing can not be perform while doing operation of Set in Python.
for i in Set:                       # for loop use for iterating the
elements.
    print(i)
    Set[0]=10   # because Set is immutable
    print(i)
```

```
0

------------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[20], line 6
      4 for i in Set:                       # for loop use for
iterating the elements.
      5     print(i)
----> 6     Set[0]=10
      7     print(i)

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

```
List = [10,11,12,-13,"Burhan","Anwar","True"]
print(type(List))
List[0]= -10
List[-3]= 13
print(List)
print(List[0:3])
print(List[-3:-1])
```

```
<class 'list'>
[-10, 11, 12, -13, 13, 'Anwar', 'True']
[-10, 11, 12]
[13, 'Anwar']
```

```
lst=[1,'str',3,11] # --->0,1,2
#how to access elements
#slicing can be done using indexing
# In posetive indexing start by zero, and by default last(element)
eliminate,
# In negative indexing starty by -1, and by default -1(element)
eliminate,
# we put element in list with [] square bracket

lst[2] = 13
print(lst)
```

```
[1, 'str', 13, 11]
```

```
lst=[1,'str',3,11] # --->0,1,2
#how to access elements
```

```python
#slicing can be done using indexing
# In posetive indexing start by zero, and by default last(element) eliminate,
# In negative indexing starty by -1, and by default -1(element) eliminate,
# we put element in list with [] square bracket

lst[2] = 13
print(lst[1])
print(lst[0:3])
print(lst[-2])
print(lst[-3:-1])
```

```
str
[1, 'str', 13]
13
['str', 13]
```

Common tasks related to data manipulation in Python include:

Data Cleaning: Removing or handling missing values, correcting errors, and ensuring data consistency.

Data Transformation: Converting data from one format to another, reshaping data, and creating new variables or features.

Data Filtering: Selecting specific rows or columns based on certain conditions.

Merging and Joining: Combining data from multiple sources based on common keys or indices.

Aggregation: Grouping and summarizing data to obtain aggregated statistics.

Sorting: Ordering data based on specific criteria.

Python provides various libraries for data manipulation, such as:

Pandas: A powerful library for data manipulation and analysis that provides data structures like DataFrame.

NumPy: A fundamental package for scientific computing with support for large, multi-dimensional arrays and matrices.

Dask: A parallel computing library that integrates with Pandas and NumPy to enable working with larger-than-memory datasets.

```
# list
# how can access the element
# slicing cand be done by using indexing

# INDEXING

# INTEGERS = z

#   -6  -5 -4 -3 -2 -1   0  +1  +2  +3  +4  +5
# BY DEFAULT IN POSITIVE INDEXING (ELIMINATE THE LAST INDEX ELEMENT )
# BY DEFAULT IN NEGATIVE INDEXING (ELEMINATE THE FIRST INDEX VALUE -1,
INDEX ELEMENT )

list  MUTABLE

list

lst = [-1,-2,-3,"Burhan","Anwar",33,]
print(type(lst))
lst[0] = 1
print(lst[-6:-1])   # by defualt (index)-1 eliminate
print(lst[0:5])     # by default (index)5 eliminate
print(lst[-3:-1])
```

```
<class 'list'>
[1, -2, -3, 'Burhan', 'Anwar']
[1, -2, -3, 'Burhan', 'Anwar']
['Burhan', 'Anwar']
```

```
HOME WORK
# lIST (MUATABLE )
# TUPLE (IMMUTABLE)
# STRING (IMMUTABLE)
# Set (immutable)
# dictionary (mutable)
```

```
Tuple = ("Burhan","Pakistan",-1,True,False,3.3,3.3,6.9,6.9)
print(type(Tuple))
print(Tuple[0:2])
print(Tuple[-9:-7])
print(Tuple)
```

```
<class 'tuple'>
('Burhan', 'Pakistan')
('Burhan', 'Pakistan')
('Burhan', 'Pakistan', -1, True, False, 3.3, 3.3, 6.9, 6.9)
```

```
Tuple = ("Burhan", "Pakistan",-1,True,False,3.3,3.3,6.9,6.9)
print(type(Tuple))
Tuple[-1] = 1
print(Tuple)
```

```
<class 'tuple'>
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[2], line 3
      1 Tuple = ("Burhan", "Pakistan",-1,True,False,3.3,3.3,6.9,6.9)
      2 print(type(Tuple))
----> 3 Tuple[-1] = 1
      4 print(Tuple)

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

```
string = "BurhanDurrani"
print(string)
print(type(string))
print(string[0])
print(string[-1])
```

```
BurhanDurrani
<class 'str'>
B
i
```

```
string = "BurhanDurrani"
print(string)
print(type(string))
string[0] = 100
```

```
BurhanDurrani
<class 'str'>
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[37], line 4
      2 print(string)
      3 print(type(string))
----> 4 string[0] = 100

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

```
bur = 'hjklghjkg'
print(bur)
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[31], line 2
      1 bur = 'hjklghjkg'
----> 2 print(bur)

TypeError: 'str' object is not callable
```

```
string = "BurhanDurrani"
print(string)
print(type(string))
print(string[0])
print(string[-1])
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[32], line 2
      1 string = "BurhanDurrani"
----> 2 print(string)
      3 print(type(string))
      4 print(string[0])

TypeError: 'str' object is not callable
```

```
string = "BurhanDurrani"
print(string)
print(type(string))
print(string[0])
print(string[-1])
```

```
BurhanDurrani
<class 'str'>
B
i
```

```
string = "aghrhan"
print(string)
print(type(string))
# string[0] = 1
print(string[0:5])
print(string[-7:-1])
```

```
aghaBurhan
<class 'str'>
aghaB
aBurha
```

```
string = "aghabuurhan"
print(string)
print(type(string))
print(string[0])
print(string[-1])
```

```
aghabuurhan
<class 'str'>
a
n
```

```
string = "burhan"
print(string)
print(type(string))
print(string[0:3])
```

```
burhan
<class 'str'>
bur
```

```
lst=[1,'str',3]
```

```
lst[2] = 13
print(lst[1])
print(lst[0:3])
print(lst[-2])
print(lst[-3:-1])
```

```
# Practice
# Tuple
# String
# List

# Date:8-12-2023(Friday)

Tuple = (4,'str',6,33,"Pakistan")
print(Tuple)
print(type(Tuple))
print(Tuple[-1])
print(Tuple[0:])
print(Tuple[0:])
```

```
(4, 'str', 6, 33, 'Pakistan')
<class 'tuple'>
Pakistan
(4, 'str', 6, 33, 'Pakistan')
(4, 'str', 6, 33, 'Pakistan')
```

```
Tuple =4,'str',6
print(Tuple)
print(type(Tuple)
```

```
(4, 'str', 6)
<class 'tuple'>
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
Cell In[12], line 4
      2 print(Tuple)
      3 print(type(Tuple))
----> 4 Tuple[0] = 2

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

```
lst = [1,2,"Pakistan"]
lst[0] =-1
print(lst)
print(lst[-1])
print(lst[0:])
```

```
[-1, 2, 'Pakistan']
Pakistan
[-1, 2, 'Pakistan']
```

```
sting = "Pakistan zindabad"
print(sting)
print(sting[-1])
print(sting[0])
print(sting[0:])
print(sting[0:19])
```

```
Pakistan zindabad
d
P
Pakistan zindabad
Pakistan zindabad
```

```
dic= {0:"Anwar",1:"Agha Burhan",2: "Agha Habib"} # --->0,1,2
#how to access elements
#slicing can not be done by using indexing
for i in dic:                       # for loop use for iterating the
elements.
    print(i,dic[i])
```

```
0 Anwar
1 Agha Burhan
2 Agha Habib
```

```
dic={0:"anwar",1:"agha burhan",2:"agha habib",3:"Pakistan Zindabad"}
for i in dic:
    print(i,dic[i])
```

```
0 anwar
1 agha burhan
2 agha habib
3 Pakistan Zindabad
```

```
string="agha burhan"
print(sting)
print(type(string))
print(string[0:5])
print(string[0:])
print(string[-2])
```

```
Pakistan zindabad
<class 'str'>
agha
agha burhan
a
```

```
Set = {22,33,22,22,22,33,4,5,"anwar","Burhan","Pakistan"}
print(type(Set))
for i in Set:                       # for loop use for iterating the
```

```
elements.
    print(i)
```

```
<class 'set'>
33
Burhan
4
5
22
anwar
Pakistan
```

```
dic = 
{0:22,1:33,2:22,3:22,4:22,5:33,6:4,7:5,8:"anwar",9:"Burhan",10:"Pakist
an"}
print(type(dic))
```

```
<class 'dict'>
```

```
Set = {1,1,1,1,2,2,2,2,3,4}
print(type(Set))
print(Set)
for i in Set:
    print(i)
```

```
<class 'set'>
{1, 2, 3, 4}
1
2
3
4
```

```
dic = {-11:"anwar",22:"burhan",33:"canada"}
print(type(dic))
print(dic)
for i in dic:
    print(i,dic[i])
```

```
<class 'dict'>
{-11: 'anwar', 22: 'burhan', 33: 'canada'}
-11 anwar
22 burhan
33 canada
```

```
SET = {1,2,2,2,2,3,3,3,3,"anwar","anwar"}
print(type(SET))
print(SET)
for i in SET:
    print(i)
#set is a immutable data dtructrue
```

```
<class 'set'>
{1, 2, 3, 'anwar'}
1
2
3
anwar
```

```python
dic = {-11:"anwar",22:"Burhan",33:"pakistan"}
print(type(dic))
print(dic)
for i in dic:
    print(i,dic[i])
```

```
<class 'dict'>
{-11: 'anwar', 22: 'Burhan', 33: 'pakistan'}
-11 anwar
22 Burhan
33 pakistan
```

```python
# https://www.vertopal.com/
# convert ipynb into pdf form
```