1-Tuple

- Ordered collection of elements
- Enclosed in round brackets ()
- Can store different type elements
- Can not change elements once stored (unmutatable)

In [253…
```
# creating a tuple

tup1 = ("Million",10,True,30.5)
tup1
```

Out[253…
```
('Million', 10, True, 30.5)
```

In [254…
```
type(tup1)
```

Out[254…
```
tuple
```

In [255…
```
#indexing of tuples

# To check the value of 2 index

tup1[2]
```

Out[255…
```
True
```

In [256…
```
# To check the value of 1 index

tup1[1]
```

Out[256…
```
10
```

In [257…
```
# element count in tuple

len(tup1)
```

Out[257…
```
4
```

In [258…
```
# last element is exclusive (it skips the last element 3 to print all the ele

tup1[0:3]
```

Out[258…
```
('Million', 10, True)
```

In [259…
```
# creating a new tuple

tup2 = (40, "Coders", 9.5, False)
tup2
```

Out[259…
```
(40, 'Coders', 9.5, False)
```

In [260…
```
# concatination of tuples
# merge the elements of tuples

# creating a tupletup1 + tup2
```

In [261…
```
# creating a new tuple

tup3 = (30,55,40,29,20,70)
tup3
```

Out[261…
```
(30, 55, 40, 29, 20, 70)
```

In [262…
```
# to check the minimun value in a tuple

min(tup3)
```

Out[262…
```
20
```

In [263…
```
# to check the maximum value in a tuple

max(tup3)
```

Out[263…
```
70
```

In [264…
```
# we can multiply tuples too, but it means repeatation only
# It will repeat the same tuple twice
tup3*2
```

Out[264…
```
(30, 55, 40, 29, 20, 70, 30, 55, 40, 29, 20, 70)
```

2-List

- Ordered collection of elements
- Enclosed in sqaure brackets [ ]
- Can change values (mutatable)

In [265…
```
# creating a list

list1 = [5, "Million", False]
list1
```

Out[265…
```
[5, 'Million', False]
```

In [266…
```
# type of List

type(list1)
```

Out[266…
```
list
```

In [267…
```
# length of List
```

```
len(list1)
```

Out[267… 3

In [268…
```
# indexing of list
```

In [269…
```
# to print the value of index 1
list1[1]
```

Out[269… 'Million'

In [270…
```
# creating a new list

list2 = [15, "Coders", True, 30.4]
list2
```

Out[270… [15, 'Coders', True, 30.4]

In [271…
```
# concatination of List
# merge the elements of lists

list1 + list2
```

Out[271… [5, 'Million', False, 15, 'Coders', True, 30.4]

In [272…
```
# we can multiply lists too, but it means repeatation only
# It will repeat the same list twice

list1*2
```

Out[272… [5, 'Million', False, 5, 'Million', False]

In [273…
```
# To reverse the list elements

list1.reverse()
list1
```

Out[273… [False, 'Million', 5]

In [274…
```
# To append any word or character with list

list1.append("Python Programming")
list1
```

Out[274… [False, 'Million', 5, 'Python Programming']

In [275…
```
# To count the occurance of specific list element

list1.count("Python Programming")
```

Out[275… 1

In [276…
```
# creating a new list

list3 = [30,10,4,33,55,78,89,60,90]
list3
```

Out[276…
```
[30, 10, 4, 33, 55, 78, 89, 60, 90]
```

In [277…
```
len(list3)
```

Out[277…
```
9
```

In [278…
```
# Sorting a list (To arrange in ascending orders)

list3.sort()
list3
```

Out[278…
```
[4, 10, 30, 33, 55, 60, 78, 89, 90]
```

3-Dictionaries

- Un-Ordered collection of elements
- Enclosed in curly brackets {}
- Store Key and value
- Can change values (mutatable)

In [279…
```
# creating a dictionary

dict1 = {"mac":100000, "dell":70000, "hp":60000, "lenovo":50000}
dict1
```

Out[279…
```
{'mac': 100000, 'dell': 70000, 'hp': 60000, 'lenovo': 50000}
```

In [280…
```
# To check the type

type(dict1)
```

Out[280…
```
dict
```

In [281…
```
# Extract data

# Extracting keys (show only keys of dictionary)

dict1.keys()
```

Out[281…
```
dict_keys(['mac', 'dell', 'hp', 'lenovo'])
```

In [282…
```
# Extracting values (show only values of dictionary)

dict1.values()
```

Out[282…
```
dict_values([100000, 70000, 60000, 50000])
```

In [283…
```
# adding new element

dict1["acer"]=40000
dict1
```

Out[283…
```
{'mac': 100000, 'dell': 70000, 'hp': 60000, 'lenovo': 50000, 'acer': 40000}
```

In [284…
```
# update the value of mac

dict1["mac"]=120000
dict1
```

Out[284…
```
{'mac': 120000, 'dell': 70000, 'hp': 60000, 'lenovo': 50000, 'acer': 40000}
```

In [285…
```
# create a new dictionary

dict2 = {"mouse":300, "keyboard":400, "headphone":1000}
dict2
```

Out[285…
```
{'mouse': 300, 'keyboard': 400, 'headphone': 1000}
```

In [286…
```
# concatinate dic1 with dict2
# need to use update to concatenate

dict1.update(dict2)
dict1
```

Out[286…
```
{'mac': 120000,
 'dell': 70000,
 'hp': 60000,
 'lenovo': 50000,
 'acer': 40000,
 'mouse': 300,
 'keyboard': 400,
 'headphone': 1000}
```

4-Set

- Un-Ordered and un-indexed
- Enclosed in curly brackets {}
- No duplicate allow

In [287…
```
# creating a set

set1 = {10, 15.5, "Million" , True, "Coders"}
set1
```

Out[287…
```
{10, 15.5, 'Coders', 'Million', True}
```

In [288…
```
# to add value in set

set1.add("Python")
set1
```

Out[288…
```
{10, 15.5, 'Coders', 'Million', 'Python', True}
```

In [289…

```
# to remove value in set

set1.remove("Coders")
set1
```

Out[289…

```
{10, 15.5, 'Million', 'Python', True}
```

In [ ]:

In [ ]:

In [ ]: