

```

1 ### Day objectives
2 ### Date:1-oct-2019
3 * Tuples
4 * Dictionaries
5 * Sets

```

```

1 ## Tuple:
2

```

In [1]:

```

1 # Tuple is a collection of heterogenous data and it is unchangable (immutable)
2 # Syntax : Tuple_name=()
3
4 t=(1,5.5,True,"python")
5 t

```

Out[1]:

```
(1, 5.5, True, 'python')
```

In [4]:

```

1 # If u want to access the tuple items we have two ways
2 # Positive index and negative index
3 print(t[0])
4 print(t[2])
5 print(t[0:3])
6 print(t[::-1]) # to reverse the tuple similiar to list and strings
7 print(t[::2]) # to get alternative elements of the tuple

```

```

1
True
(1, 5.5, True)
('python', True, 5.5, 1)
(1, True)

```

In [26]:

```
1 len(t) # Length of the tuple
```

Out[26]:

```
4
```

In [7]:

```
1 print(dir(tuple),end=" ")
```

```

['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__',
 '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__',
 '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__',
 '__lt__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
 '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__', 'count', 'index']

```

In [11]:

```
1 t.count("python") # To count the occurrence of the element
```

Out[11]:

1

In [12]:

```
1 t.index(5.5) # To return the index value of the element
```

Out[12]:

1

Dictionaries:

In [14]:

```
1 # It is collection of heterogenous data and are mutable data types
2 # If you want to store the data into the dictionaries we access through keys and values
3 # Syntax : d={"key1":val1,"key2":val2,.....}
4
5 d = {"k1":1,"k2":55.5,"k3":True,"k4":"APSSDC"}
6 d # keys are unique (should not be same)
```

Out[14]:

```
{'k1': 1, 'k2': 55.5, 'k3': True, 'k4': 'APSSDC'}
```

In [17]:

```
1 print(d["k4"]) # We can access the elements from the dictionaries by keys only
```

APSSDC

In [21]:

```
1 d1={2.5:250} # values may be of any type
2 d1[2.5] # Keys should be only strings ,numbers ,float values
```

Out[21]:

250

In [25]:

```
1 d1 = {}
2 d1["a"]=20
3 d1["b"]=30
4 d1
```

Out[25]:

```
{'a': 20, 'b': 30}
```

In [27]:

```
1 len(d1)    # Length of the dictionary
```

Out[27]:

2

In [33]:

```
1 d1.keys()    # If you want to access the only keys use => dict.keys()
2 d1.keys()
```

Out[33]:

dict_keys(['a', 'b'])

In [34]:

```
1 d1.values()    # If you want to access the only values use => dict.values()
```

Out[34]:

dict_values([20, 30])

In [35]:

```
1 # To access the items
2 # Item => key:values          # Syntax => dict.items()
3 d1.items()
```

Out[35]:

dict_items([('a', 20), ('b', 30)])

In [37]:

```
1 print(dir(dict),end=" ")
```

```
['__class__', '__contains__', '__delattr__', '__delitem__', '__dir__', '__doc__', '__eq__', '__format__', '__ge__', '__getattribute__', '__getitem__', '__gt__', '__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__setattr__', '__setitem__', '__sizeof__', '__str__', '__subclasshook__', 'clear', 'copy', 'fromkeys', 'get', 'items', 'keys', 'pop', 'popitem', 'setdefault', 'update', 'values']
```

In [38]:

```
1 d1
```

Out[38]:

{'a': 20, 'b': 30}

In [40]:

```
1 d1.clear()
2 d1          # To clear the elements in the dict
```

Out[40]:

```
{}
```

In [46]:

```
1 d1= {'a':40,'b':50,'c':30}
2 d2=d1.copy()      # To copy the elements in the dict into another dict
3 d2
```

Out[46]:

```
{'a': 40, 'b': 50, 'c': 30}
```

In [48]:

```
1 d2.get('a')      # to get the values of the key d2['a']
```

Out[48]:

```
40
```

In [52]:

```
1 d2.pop('b')# to pop out the elements out of dict
2 d2
```

Out[52]:

```
{'a': 40}
```

In [53]:

```
1 d2.popitem() # to remove entire last item from dict
```

Out[53]:

```
('a', 40)
```

In [55]:

```
1 d2.setdefault('a',7) # It assign to the dict by taking two argumets or dict['key']-
```

Out[55]:

```
7
```

In [58]:

```

1 d2.update({'b':20})
2 d2          # to update any existing value or new values to the dict

```

Out[58]:

```
{'a': 7, 'b': 20}
```

In [59]:

```

1 for item in d2.items():
2     print(item)

```

```

('a', 7)
('b', 20)

```

- The value in the dictionary may be any data type

In []:

```

1 # Tasks:
2 # 1.create a dictionary to store the employee details
3     # name: ----
4     # mobile:----
5     # email id:---
6     # empid:-----
7     # age:---
8 # 2.create a function to create a dictionary to store the multiple emp deatils through
9     # emp={123:["charan",9441705788,23,"email"]}
10 # 3.L=[10,20,10,40,40,30,20,50,70] by using this list create a dict to store the number
11     # d={10:2,20:2,40:2,30:1}
12     # keys= list element    # values =number of occurance of element
13 # 4."pythonprogramming" by using this string create a dict with charecter and number of
14     #d={'p':2,'y':1,.....}
15 # 5.string= "mastan narmada ranga charan sai mastan "
16     # d={'mastan':2,'narmada':1}
17 # 6.create a dictionary to store the numbers of the squares in between 1 to 20 both are
18     #d={1:1,2:4,3:9,.....}
19 # 7.L=[10,20,10,30,80,30,10,40,55,60] find the highest frequency element in function

```

In [61]:

```

1 # 1.create a dictionary to store the employee details
2     # name: ----
3     # mobile:----
4     # email id:---
5     # empid:-----
6     # age:----
7
8 dic1={'name':'charan','mobile':9441705788,'emailid':'saicharan6102@gmail.com','empid':
9 dic1

```

Out[61]:

```

{'name': 'charan',
 'mobile': 9441705788,
 'emailid': 'saicharan6102@gmail.com',
 'empid': '174g5a0221',
 'age': 21}

```

In [62]:

```

1 # 2.create a function to create a dictionary to store the multiple emp deatils through
2     # emp={123:["charan",9441705788,23,"email"]}
3 dic2={}
4 def empdetails(n):
5     for i in range(0,n+1):
6         empid=int(input("enter employee id"))
7         lst=[]
8         lst=input().split()
9         dic2.update({empid:lst})
10    print(dic2)
11    n=int(input("enter the number of employees details to store"))
12    empdetails(n)

```

```

enter the number of employees details to store2
charan 9441705788 23 sai@gmail.com
sai 8919124677 20 charan@gmail.com
car 9441 25 car@gmail.com

```

In [19]:

```
1  # 3.l=[10,20,10,40,40,30,20,50,70] by using this list create a dict to store the number
2      # d={10:2,20:2,40:2,30:1}
3      # keys= list element      # values =number of occurrence of element
4
5
6  def occurancedict(lst):
7      dic3={}
8      count=0
9      unilist=[]
10     for i in lst:
11         if i not in unilist:
12             unilist.append(i)
13     unilist
14     for i in unilist:
15         count=lst.count(i)
16         dic3.update({i:count})
17     print(dic3)
18 lst1=[10,20,10,40,40,30,20,50,70]
19 occurancedict(lst1)
20
```

```
{10: 2, 20: 2, 40: 2, 30: 1, 50: 1, 70: 1}
```

In [31]:

```
1  # another method for the above program
2  def listoccur(l1):
3      d={}
4      for i in l1:
5          if i not in d.keys():
6              d[i]=l1.count(i)
7      return d
8  li=[10,20,10,40,40,30,20,50,70]
9  listoccur(li)
```

Out[31]:

```
{10: 2, 20: 2, 40: 2, 30: 1, 50: 1, 70: 1}
```

In [25]:

```

1  # 4. "pythonprogramming" by using this string create a dict with charecter and number of
2      #d={'p':2, 'y':1,.....}
3
4  def charoccur(s):
5      dic4={}
6      count=0
7      for i in range(len(s)):
8          count=s.count(s[i])
9          dic4.update({s[i]:count})
10     print(dic4)
11     string="pythonprogramming"
12     charoccur(string)
13

```

```

{'p': 2, 'y': 1, 't': 1, 'h': 1, 'o': 2, 'n': 2, 'r': 2, 'g': 2, 'a': 1,
'm': 2, 'i': 1}

```

In [28]:

```

1  # 5.string= "mastan narmada ranga charan sai mastan "
2      # d={'mastan':2, 'narmada':1}
3
4  def stringoccur(st):
5      dic5={}
6      for i in st.split():
7          dic5[i]=st.count(i)
8      return dic5
9  st=input()
10 stringoccur(st)

```

mastan narmada ranga charan sai mastan

Out[28]:

```

{'mastan': 2, 'narmada': 1, 'ranga': 1, 'charan': 1, 'sai': 1}

```

In [29]:

```

1  # 6.create a dictionary to store the numbers of the squares in between 1 to 20 both are
2      #d={1:1,2:4,3:9,.....}
3
4  def squaredict(n):
5      dic6={}
6      for i in range(0,n+1):
7          dic6[i]=i**2
8      return dic6
9  n=int(input("enter the range"))
10 squaredict(n)
11

```

enter the range5

Out[29]:

```

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

```


In [51]:

```

1  # 7. l=[10,20,10,30,80,30,10,40,55,60] find the highest frequency element in function
2
3  def frequency(lst):
4      dic7={}
5      m=[]
6      for i in lst:
7
8          dic7[i]=lst.count(i)
9
10         m=dic7.values()
11         maxval=max(m)
12         return[key for key,v in dic7.items if v==maxval]
13
14
15     return dic7
16
17
18
19 ls=[10,20,10,30,80,30,10,40,55,60]
20 frequency(ls)

```

TypeError Traceback (most recent call last)

<ipython-input-51-15ce4cd6c8f9> in <module>

18

19 ls=[10,20,10,30,80,30,10,40,55,60]

--> 20 frequency(ls)

<ipython-input-51-15ce4cd6c8f9> in frequency(lst)

10 m=dic7.values()

11 maxval=max(m)

--> 12 return[key for key,v in dic7.items if v==maxval]

13

14

TypeError: 'builtin_function_or_method' object is not iterable

In []:

1