tuples

February 11, 2023

```
[1]: t=()
```

```
[2]: type(t)
```

```
[2]: tuple
```

```
[5]: t1=(1,2,3,4,5,4.5,1+7j,"kishor",True)
```

```
[6]: type(t1)
```

```
[6]: tuple
```

```
[7]: t1
```

```
[7]: (1, 2, 3, 4, 5, 4.5, (1+7j), 'kishor', True)
```

```
[8]: t1[0]
```

```
[8]: 1
```

```
[10]: t1[0]=="k"
```

```
[10]: False
```

```
[11]: len(t1)
```

```
[11]: 9
```

```
[12]: t1[::-1]
```

```
[12]: (True, 'kishor', (1+7j), 4.5, 5, 4, 3, 2, 1)
```

```
[17]: t1=t1[::-1]
```

```
[18]: t1
```

```
[18]: (True, 'kishor', (1+7j), 4.5, 5, 4, 3, 2, 1)
```

```
[19]: t1[0:3]
```

```
[19]: (True, 'kishor', (1+7j))
```

```
[20]: t1
```

```
[20]: (True, 'kishor', (1+7j), 4.5, 5, 4, 3, 2, 1)
```

```
[21]: t1.count(1)
```

```
[21]: 2
```

```
[23]: t1.index(5)
```

```
[23]: 4
```

```
[27]: t2=["k","a"]
```

```
[29]: t2.sort()
```

```
[30]: t2
```

```
[30]: ['a', 'k']
```

```
[31]: t1
```

```
[31]: (True, 'kishor', (1+7j), 4.5, 5, 4, 3, 2, 1)
```

```
[32]: t1.count(True)
```

```
[32]: 2
```

```
[33]: t1.count(1)
```

```
[33]: 2
```

```
[34]: t1
```

```
[34]: (True, 'kishor', (1+7j), 4.5, 5, 4, 3, 2, 1)
```

```
[36]: l=[1,2,3,4]
```

```
[37]: l
```

```
[37]: [1, 2, 3, 4]
```

```
[39]: l[0:3]
```

[39]:
```
[1, 2, 3]
```

[40]:
```
l[0]="k"
```

[41]:
```
l
```

[41]:
```
['k', 2, 3, 4]
```

[43]:
```
l.append(45)
```

[44]:
```
l
```

[44]:
```
['k', 2, 3, 4, 45]
```

[46]:
```
for i in t1:
    print(i,type(i))
```

```
True <class 'bool'>
kishor <class 'str'>
(1+7j) <class 'complex'>
4.5 <class 'float'>
5 <class 'int'>
4 <class 'int'>
3 <class 'int'>
2 <class 'int'>
1 <class 'int'>
```

[47]:
```
for i in t1:
    print(i)
```

```
True
kishor
(1+7j)
4.5
5
4
3
2
1
```

[48]:
```
t2=(1,2,3,4)
```

[49]:
```
t2
```

[49]:
```
(1, 2, 3, 4)
```

1 Tuple are basically follows imutability concepts where it is not going to allow to change any element at a particular Index

t2*3

```
[51]: max(t2)
```

```
[51]: 4
```

```
[52]: t1=(1,2,3,4,5)
      t2=(4,5,7,6)
```

```
[53]: t3=(t1,t2)
```

```
[54]: t3
```

```
[54]: ((1, 2, 3, 4, 5), (4, 5, 7, 6))
```

```
[55]: del(t2)
```

```
[56]: t2
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[56], line 1
----> 1 t2

NameError: name 't2' is not defined
```

```
[57]: t1
```

```
[57]: (1, 2, 3, 4, 5)
```

```
[58]: len(t1)
```

```
[58]: 5
```

```
[59]: 4 in t1
```

```
[59]: True
```

```
[60]:
```

```
[60]: False
```

1.1 sets

[61]:
```
s={}
```

[62]:
```
s
```

[62]:
```
{}
```

[63]:
```
type(s)
```

[63]:
```
dict
```

[65]:
```
s1={1,2,3,4,5}
```

[66]:
```
type(s1)
```

[66]:
```
set
```

[67]:
```
s2={1,1,1,2,2,3,3,4,2,5,6,6,}
```

[68]:
```
s2
```

[68]:
```
{1, 2, 3, 4, 5, 6}
```

’’’ sets are remove duplicate element’’’

[69]:
```
list(s2)
```

[69]:
```
[1, 2, 3, 4, 5, 6]
```

[70]:
```
tuple(s2)
```

[70]:
```
(1, 2, 3, 4, 5, 6)
```

’’’ set to list and set to tuple conversion applicable in sets’’’

[71]:
```
l=list(s2)
```

[72]:
```
l
```

[72]:
```
[1, 2, 3, 4, 5, 6]
```

[73]:
```
set(l)
```

[73]:
```
{1, 2, 3, 4, 5, 6}
```

[74]:
```
s4={1,2,3,4,[2,3,4,5]}
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[74], line 1
----> 1 s4={1,2,3,4,[2,3,4,5]}

TypeError: unhashable type: 'list'
```

```
[75]: s5={(1,2,3,4),(1,2,3,4,5)}
```

```
[76]: s5
```

```
[76]: {(1, 2, 3, 4), (1, 2, 3, 4, 5)}
```

```
[77]: s6={"Kk","kk",1,2,3,4,8}
```

```
[78]: s6
```

```
[78]: {1, 2, 3, 4, 8, 'Kk', 'kk'}
```

```
[80]: s7={"kk","kk",1,2,3,4,8}
```

```
[81]: s7
```

```
[81]: {1, 2, 3, 4, 8, 'kk'}
```

```
[83]: for i in s7:
          print(i)
```

```
1
2
3
kk
4
8
```

```
[84]: s7.add(9)
```

```
[85]: s7
```

```
[85]: {1, 2, 3, 4, 8, 9, 'kk'}
```

```
[89]: s7.add(2) #''' It always try to hold unique element'''
```

```
[87]: s7
```

```
[87]: {1, 2, 3, 4, 8, 9, 'kk'}
```

```
[90]: s7.pop()
```

```
[90]: 1
```

```
[91]: s7
```

```
[91]: {2, 3, 4, 8, 9, 'kk'}
```

```
[92]: s7.pop()
```

```
[92]: 2
```

```
[93]: s7
```

```
[93]: {3, 4, 8, 9, 'kk'}
```

```
[94]: ## removeing and adding are possible in set
```

```
[95]: s7.clear()
```

```
[96]: s7
```

```
[96]: set()
```

```
[97]: s8={1,2,3,4}
      s9={1,2,3,5}
```

```
[98]: s8.difference(s9)
```

```
[98]: {4}
```

```
[99]: s9.difference(s8)
```

```
[99]: {5}
```

1.2 Dictionary

```
[100]: d={}
```

```
[101]: type(d)
```

```
[101]: dict
```

```
[103]: d={"kk":1,"KJ":2,"LL":6}
```

```
[104]: d
```

```
[104]: {'kk': 1, 'KJ': 2, 'LL': 6}
```

```
[105]: len(d)
```

```
[105]: 3
```

```
[107]: d.keys()
```

```
[107]: dict_keys(['kk', 'KJ', 'LL'])
```

```
[108]: d.values()
```

```
[108]: dict_values([1, 2, 6])
```

```
[109]: d2={"name":"kishor","name":"kurhe"}
```

```
[110]: d2
```

```
[110]: {'name': 'kurhe'}
```

```
[111]: # key should be unique
```

```
[112]: d3={"name":"kishor",123:"kurhe"}
```

```
[113]: d3
```

```
[113]: {'name': 'kishor', 123: 'kurhe'}
```

```
[114]: d4={True:'abc'}
```

```
[115]: d4
```

```
[115]: {True: 'abc'}
```

```
[116]: d5={(1,2,3):"num"}
```

```
[117]: d5
```

```
[117]: {(1, 2, 3): 'num'}
```

```
[118]: set(d5)
```

```
[118]: {(1, 2, 3)}
```

```
[119]: set(d3)
```

```
[119]: {123, 'name'}
```

```
[121]: d6={"fruit":["mango","Banana","Apple"]}

[122]: d6

[122]: {'fruit': ['mango', 'Banana', 'Apple']}

[123]: d7={"fruit":{"mango","Banana","Apple"}}

[136]: d7

[136]: {'fruit': {'Apple', 'Banana', 'mango'}, 'time': (8, 9)}

[137]: d7["time"]=(8,9)

[138]: d7

[138]: {'fruit': {'Apple', 'Banana', 'mango'}, 'time': (8, 9)}

[142]: d7["name"]="Kishor"

[143]: d7

[143]: {'fruit': {'Apple', 'Banana', 'mango'}, 'time': (8, 9), 'name': 'Kishor'}

[144]: d7["name"].upper()

[144]: 'KISHOR'

[145]: type(d7)

[145]: dict

[147]: d7["fruit"]

[147]: {'Apple', 'Banana', 'mango'}

[148]: del d7["name"]

[149]: d7

[149]: {'fruit': {'Apple', 'Banana', 'mango'}, 'time': (8, 9)}

  [1]: d7={"fruit":{"mango","Banana","Apple"}}
       list(d7.keys())

  [1]: ['fruit']
```

[2]:
```python
d7.items()
```

[2]:
```
dict_items([('fruit', {'Apple', 'mango', 'Banana'})])
```

[3]:
```python
list(d7.items())
```

[3]:
```
[('fruit', {'Apple', 'Banana', 'mango'})]
```

[4]:
```python
d7.pop("fruit")
```

[4]:
```
{'Apple', 'Banana', 'mango'}
```

[5]:
```python
d7.fromkeys((1,2,3),("a","b","c"))
```

[5]:
```
{1: ('a', 'b', 'c'), 2: ('a', 'b', 'c'), 3: ('a', 'b', 'c')}
```

#dictionary comprehensions

[8]:
```python
{i:i**2for i in range(1,11)}
```

[8]:
```
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100}
```

[9]:
```python
list(range(1,11))
```

[9]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

[11]:
```python
import math
d21={i:math.log10(i)for i in range(1,11)}
```

d21

[12]:
```python
d21
```

[12]:
```
{1: 0.0,
 2: 0.3010299956639812,
 3: 0.47712125471966244,
 4: 0.6020599913279624,
 5: 0.6989700043360189,
 6: 0.7781512503836436,
 7: 0.8450980400142568,
 8: 0.9030899869919435,
 9: 0.9542425094393249,
 10: 1.0}
```

[17]:
```python
for i in d21.keys():
    if i%2==0:
        print(d21[i],type(d21))
```

```
0.3010299956639812 <class 'dict'>
0.6020599913279624 <class 'dict'>
0.7781512503836436 <class 'dict'>
0.9030899869919435 <class 'dict'>
1.0 <class 'dict'>
```

[ ]:

```

```