Untitled

March 29, 2023

0.1 tuples

[24]:
```
t=()
```

[ ]:
```
## representation of tuples is done with ()
```

[2]:
```
t1=(1,2,3,45,45.5,"punith",True)
```

[3]:
```
## tuples can have int,bool,string
```

[4]:
```
type(t1)
```

[4]:
```
tuple
```

[5]:
```
## the major difference in list and tuples is:
## list is represented in [] and tuples is represented in()
```

[7]:
```
t1
```

[7]:
```
(1, 2, 3, 45, 45.5, 'punith', True)
```

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

[8]:
```
1
```

[9]:
```
t1[4]
```

[9]:
```
45.5
```

[10]:
```
t1[::-1]
```

[10]:
```
(True, 'punith', 45.5, 45, 3, 2, 1)
```

[11]:
```
## reverse order, indexing same as lst
```

[14]:
```
t1=t1[::-1]
```

[16]:
```
t1[0:3]
```

[16]:
```
(True, 'punith', 45.5)
```

[17]:
```
t1
```

[17]:
```
(True, 'punith', 45.5, 45, 3, 2, 1)
```

[21]:
```
## tuples have only two operations whereas lst has many functions.
```

[23]:
```
## press tab to know the functions
```

[25]:
```
## 1st function is count and other function is index
```

[18]:
```
t1.count(4)
```

[18]:
```
0
```

[19]:
```
t1.count("abc")
```

[19]:
```
0
```

[26]:
```
t1.index(1)
```

[26]:
```
0
```

[27]:
```
t1.index("punith")
```

[27]:
```
1
```

[28]:
```
t1.count(1)
```

[28]:
```
2
```

[30]:
```
## y two whereas we only 1 is repeating one time
## because the compiler considers the true as 1 .
```

[31]:
```
t1.count(True)
```

[31]:
```
2
```

[33]:
```
##  the above statement is similar example
```

[34]:
```
t2=(False,1,2,56,36,98.2,True,0)
```

[35]:
```
t2.count(0)
```

[35]:
```
2
```

[37]:
```
t2.count(1)
```

[37]: 2

[38]:
```
## count defines the number of availablity of the function in a tuples
## indexfunction helps to find the index of the function
```

0.2 the major difference between tuples and lst is:

[39]:
```
## example
```

[40]:
```
t1
```

[40]: (True, 'punith', 45.5, 45, 3, 2, 1)

[41]:
```
lst=[1,2,3,45,6,0,True,"punith"]
```

[44]:
```
t1[0]=False
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[44], line 1
----> 1 t1[0]=False

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

[46]:
```
lst[0]=100
```

[47]:
```
lst
```

[47]: [100, 2, 3, 45, 6, 0, True, 'punith']

[50]:
```
## so in tuples no changes for the index can be done
## whereas in lst index can be replaced
### so tuples are immutable entity
## lst are mutable
```

[51]:
```
t1
```

[51]: (True, 'punith', 45.5, 45, 3, 2, 1)

[55]:
```
for x in t1:
    print(x)
```

```
True
punith
```

```
45.5
45
3
2
1
```

[53]:
```
## tuples can be itterated
```

[56]:
```
for x in t1:
    print(x,type(x))
```

```
True <class 'bool'>
punith <class 'str'>
45.5 <class 'float'>
45 <class 'int'>
3 <class 'int'>
2 <class 'int'>
1 <class 'int'>
```

[57]:
```
## the datatype can also be obtained by using above syntax
```

[58]:
```
t3=( 1,2,3,4)
```

[59]:
```
t3*3
```

[59]: (1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)

[60]:
```
## concatenation
```

[61]:
```
max(t3)
```

[61]: 4

[62]:
```
min(t3)
```

[62]: 1

[71]:
```
t1=(1,2,3,4)
t2=(7,8,9,5)
```

[74]:
```
t4=(t1,t2)
```

[73]:
```
len(t1)
```

[73]: 4

[75]:
```
t1
```

[75]:
```
(1, 2, 3, 4)
```

[76]:
```
"punith" in t1
```

[76]:
```
False
```

[77]:
```
2 in t1
```

[77]:
```
True
```

[78]:
```
True in t1
```

[78]:
```
True
```

[ ]:
```

```