Data- Structure Tuple

A tuple similar to list.

the difference b/w the two is that can't change the elements of tuple once it is assigned whereas in the list, elements can be changed.

ordered data structure

Tuple Creation

```
t = () # empty tuple
print(t)

t1 = (1,2,3, 4) # tuple having intergers
print(t1)

t2 = (1, 'data', 28, 'features', 'A') # tuple with mixed datatypes.
print(t2)

t3 = (1, (2,3,4), [1, 'data', 28, 'features'], 7, 8)
print(t3)
```

```
()
(1, 2, 3, 4)
(1, 'data', 28, 'features', 'A')
(1, (2, 3, 4), [1, 'data', 28, 'features'], 7, 8)
```

```
# only parenthesis is not enough
t = ('harsh')
print(type(t))
p = (1)
print(type(p))
```

```
<class 'str'>
<class 'int'>
```

```
# need a comma at the end
t = ('harsh',)
type(t)
```

```
tuple
```

```
# parenthesis is optional
t = 'harsh',
print(type(t))
print(t)
```

```
<class 'tuple'>
('harsh',)
```

```
p = (1,2,3,4)
print(list(p))
```

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

Accessing Elements in Tuple

```
t = ('harsh', 'raj', 'RV', 'sumit', 'alok', 'Himanshu')
print(t[0])
print(t[-3])
```

```
harsh
sumit
```

```
# nested tuple
t = ('harsh', 'raj', ('RV', 'sumit', 'alok', 'Himanshu'))
print(t[2])
print(t[2][2])
```

```
('RV', 'sumit', 'alok', 'Himanshu')
alok
```

```
# slicing
t = (1,2,3,4,5,6,7)
print(t[1:5])

# print the element form starting to 2nd last elements
print(t[:-3])

# print the element form starting to end
print(t[::-1])
```

```
(2, 3, 4, 5)
(1, 2, 3, 4)
(7, 6, 5, 4, 3, 2, 1)
```

Changing a Tuple

unlike lists, Tuples are immutable.

this means that a elements of tuple cannot be change once it has been assigned. But is the element is itself a mutable datatype like list, its nested items can be changed.

```
t = (1, 2, 3, 4, [5, 6, 7])
t[3] = 1
```

```
----------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
<ipython-input-9-2c3bd42420b4> in <module>
      1 t = (1, 2, 3, 4, [5, 6, 7])
----> 2 t[3] = 1

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

```
t[4][2] = 'Harsh'
print(t)
```

```
(1, 2, 3, 4, [5, 6, 'Harsh'])
```

```
# concatinating tuples
t1 = (1, 2, 3, 4)
t2 = (5, 6, 7 )
t  = t1 + t2
print(t)
```

```
(1, 2, 3, 4, 5, 6, 7)
```

```
# repeat the element in a tuple for a given number of time using *
operator
t = (("harsh",)*5)
print(t)
```

```
('harsh', 'harsh', 'harsh', 'harsh', 'harsh')
```

Tuple Delection

```
# we cannot change the elements in a tuple.
# that also means we cannot delete or remove items from of tuple.

# delete entire tuple using del keyword
t = (1, 2, 3, 4, 5, 6)
print(t)
a = t
del t
print(a)
```

```
(1, 2, 3, 4, 5, 6)
(1, 2, 3, 4, 5, 6)
```

Touple count

```
t = (1, 2, 3, 1, 2, 2, 3, 4)
t.count(2)
```

```
3
```

Tuple index

```
t = (1, 3, 1, 2, 2, 3, 4)
print(t.index(2)) # return index of first element is equal to 0
```

```
3
```

Tuple Memebership

```
t = (1, 2, 3, 4, 5, 6)
print( 2 in t )
```

```
True
```

```
print ( 8 in t)
```

```
False
```

Built in function

tuple length

```
t = (1, 2, 3, 4, 5, 6)
print(len(t))
```

```
6
```

tuple sort

```
t = (7, 2, 5, 4, 1, 6, 9, 10, 3)
#print(id(t))
new_t = sorted(t)
print(new_t)     # take elements in the tuple and return a new sorted
list
                 # does not sort the tuple itself.
t = tuple(new_t)
print(t)
print(id(t))
```

```
[1, 2, 3, 4, 5, 6, 7, 9, 10]
(1, 2, 3, 4, 5, 6, 7, 9, 10)
2267507339936
```

```
t = ("Hello", "Hye", "HAme")
a = sorted(t)
print(a)
```

```
['HAme', 'Hello', 'Hye']
```

```python
# get largest element in tuple
t = (2, 5, 4, 1, 6)
print(max(t))
```

```
6
```

```python
# get smallest element in tuple
t = (2, 5, 4, 1, 6, 100,-10)
print(min(t))
```

```
-10
```

```python
# get sum of elements in tuple
t = (2, 5, 4, 1, 6, 100, 0,-10)
print(sum(t))
```

```
108
```