Data structures in Python

Week 7A: Tuples

- Tuple
  - It is a collection of ordered items and Tuple is like Lists [ ].
  - But values for tuple are listed within ( ) and items/objects in the tuple **are not changeable**.

```
tuple2 = ("orange", "apple", -45, 24.5)
list1 = [45.0, 89.0,"Banana","LUT", 20]
print(tuple2)
print(list1)
```

```
Shell
Python 3.7.9 (bundled)
>>> %Run 'example 1t.py'
(
('orange', 'apple', -45, 24.5)
[45.0, 89.0, 'Banana', 'LUT', 20]
```

example 2t.py

```
tuple1 = (2, 3, 4, 5, -6)
tuple2 = ("orange", "apple", -45, 24.5)
tuple3 = tuple1+tuple2
print(tuple3)
```

```
Shell
Python 3.7.9 (bundled)
>>> %Run 'example 2t.py'
(2, 3, 4, 5, -6, 'orange', 'apple', -45, 24.5)
```

xample 3t.py

```
tuple1 = (2, 3, 4, 5, -6)
list1 = [2, 3, 4, 5, -6]

list1[3] = 100
tuple1[3] = 100

print(list1)
print(tuple1)
```

What is the output of this code?

Line no. 5 will throw error. Because tuple objects are not changeable

Tuples are immutable which means you cannot update or change the values of tuple elements.

- Accessing and deleting tuple elements:

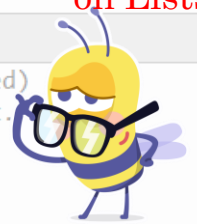
example 4t.py

```
1 tup1 = (45, 89, 34, -20, 23,56)
2 print(tup1[2]) # which is same as list
3 print(len(tup1))
4 print(min(tup1))
5 print(max(tup1))
```

Can I use these min(), max() on Lists? Yes, of course.

Shell

```
Python 3.7.9 (bundled)
>>> %Run 'example 4t.
  34
  6
  -20
  89
```



```
1 tup1 = (45, 89, 34, -20, 23,56)
2 print(tup1[1:3])
3 print(tup1[:])
4 print(tup1[len(tup1)-1])
5
```

Shell

```
Python 3.7.9 (bundled)
>>> %Run 'example 5t.py'
  (89, 34)
  (45, 89, 34, -20, 23, 56)
  56
```

I did these already on Lists too in lecture slides Week 7!

example 5t.py

```
1 tup1 = ("Ashok", "Xiaboo", "Chen", "Wali")
2 print(tup1)
3 del(tup1)
4 print(tup1)
5
```

Shell

```
Python 3.7.9 (bundled)
>>> %Run 'example 5t.py'
  ('Ashok', 'Xiaboo', 'Chen', 'Wali')
  Traceback (most recent call last):
    File "Z:\Python 2021 Fall\Fall 2021 CT60A0203\Week 7\example 5t.py", line 4, in <module>
      print(tup1)
  NameError: name 'tup1' is not defined
>>>
```

Removing individual tuple elements is not possible but deleting an entire tuple is possible.

del()

example 5t.py

```
1  tup1 = (45.0,24.5, -20.0, 89)
2  for i in tup1:
3      if i>0:
4          print(i)
5
```

Shell

```
Python 3.7.9 (bundled)
>>> %Run 'example 5t.py'

  45.0
  24.5
  89

>>>
```

Ok, How to remove part of elements from the tuple? Suppose you want to remove negative values only from the tup1. Check the code given below:

example 5t.py

```
1  #code to remove negative elements from a tuple
2  tup1 = (45.0, 24.5, -20.0, 89)
3  l1 = list(tup1) # converting tuple as list
4  for i in l1[:]:
5      if i<0:
6          l1.remove(i) # removing negative values
7  tup1 = tuple(l1) #reassigning tup1 as l1
8  print(tup1)
9
```

Shell

```
Python 3.7.9 (bundled)
>>> %Run 'example 5t.py'

  (45.0, 24.5, 89)
```

Can we sort the elements of tuple?

Well, it is possible to display the values of tuple in sorted form but can not be ordered directly. But can be done by transferring the values to List then rewrite it on tuple.

```
#tuple
tup1 = (16,41, -5.6, 8.912)
print(sorted(tup1)) # displays in sorted form but tup1 order have no changes.
print(sorted(tup1, reverse=True))
```

sort() will not work on tuple like we do in **Lists**.