28th April 21

Previous Day

- List reverse()
- Taking a list as user input
- How to write your own reverse function
- list operations vs list item operations
- Pass by value vs pass by reference
- Tuples
- mcqs

Python Tuple Dictionary and Set

Lecture Flow

Topics and Explanations

Tuple

- tuple is just an ordered sequence of items of any datatype enclosed by ().
- items of any datatype: list, string, integer, float, boolean, tuple
- order is important

Program

```
18  T = (True, 23, 3.4, "Hello Hello", [1,2,3])
19  print(T)
20  print(type(T))
21  T2 = (True, 23, [1,2,3], 3.4, "Hello Hello")
22  print(T==T2)
```

```
28    T = tuple("123123")
29    print(T)
```

Output

```
~/RichBiodegradableLicenses$ python3 lecture18_notes.py
(True, 23, 3.4, 'Hello Hello', [1, 2, 3])
<class 'tuple'>
False
('1', '2', '3', '1', '2', '3')
~/RichBiodegradableLicenses$
```

```
23    # Test for ordered, if T==T2 is true then it is unordered else it is
      ordered.
24
25    # Benefit of having any ordered sequence is you can use indexes to fetch a
      single value from the sequence.
26
27    # tuple() function is used to convert anyother datatype into a tuple
30    # tuple function on a list
```

Program

```
31    # 1. tuple and list
32    # a) convert a list into a tuple
33    T = tuple([1, 2, 3, 4, 5])
34    print(T)
35    print(type(T))
```

Output

```
(1, 2, 3, 4, 5)
<class 'tuple'>
```

Program

```
37  # b) convert a tuple into a list
38  L = list((1,2,3,4))
39  print(L)
40  print(type(L))
```

Output

```
[1, 2, 3, 4]
<class 'list'>
~/RichBiodegradableLicenses$
```

Program

```
42  # we will learn how to take a tuple as input from the user.
43  s = input("Enter a tuple: ")
44  T = tuple(s)
45  print(T)
46  L = s.split(" ")
47  for idx in range(0, len(L)):
48      L[idx] = int(L[idx])
49  T2 = tuple(L)
50  print(T2)
```

Output

```
Enter a tuple: 12 33 1231
('1', '2', ' ', '3', '3', ' ', '1', '2', '3', '1')
(12, 33, 1231)
~/RichBiodegradableLicenses$
```

```
59  # Lists had a lot of functions: append() [add an item to the end of the
    list], insert() [add an item in between the list], pop() [removes an item
    from a particular index in the list], remove() [removes the first
    occurence of a value in the list], extend() [join the items of 2 lists
    together]
60
61  # A tuple does not support any of these operations because it is
    immutable. [once created cannot be changed]
62
63  # Tuple, string, integer, float, boolean: immutable
64  # List, [Dictionary, we will learn this later]: mutable
65
66  # Tuple supports indexing
67  # index is basically the memory location of an item wrt to the beginning
    of the list/tuple/...
```

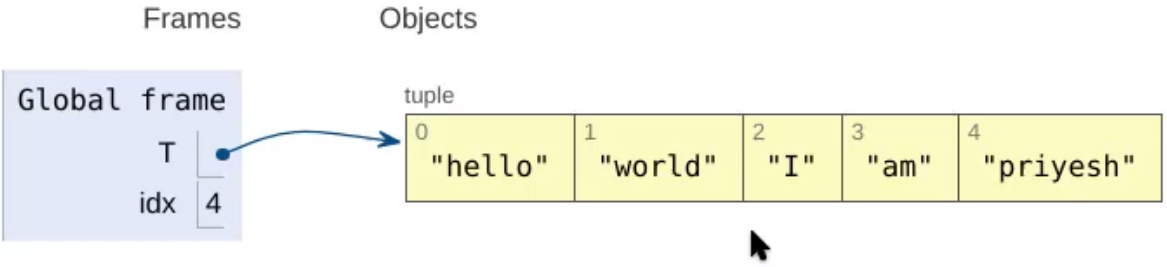
Program

```
1  T = ("hello", "world", "I", "am", "priyesh")
2
3  for idx in range(0, len(T), 2):
4    print(T[idx].upper())
```

Output

Print output (drag lower right corner to resize)

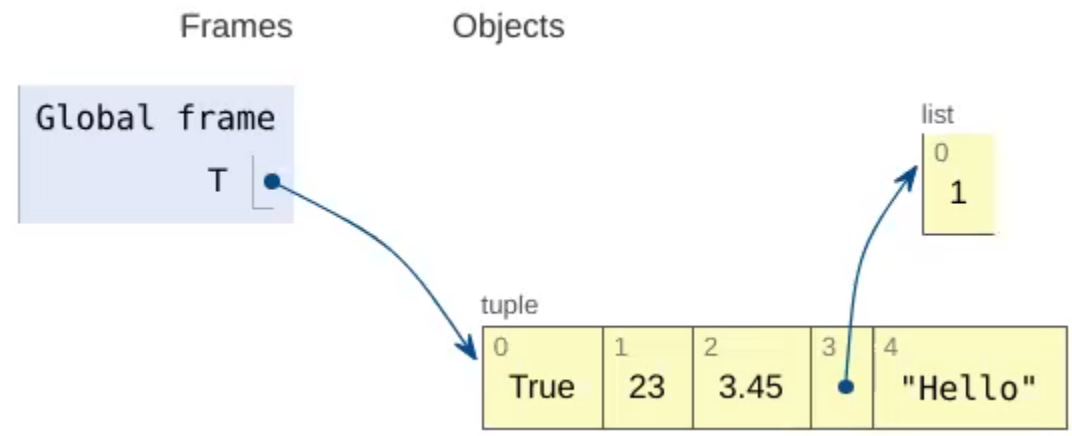
```
HELLO
I
PRIYESH
```



Program

```
1  T = (True, 23, 3.45, [1, "ABC", "DEF"], "Hello")
2  T[3].pop()
3  T[3].pop()
4  T[3].pop()
```

Output

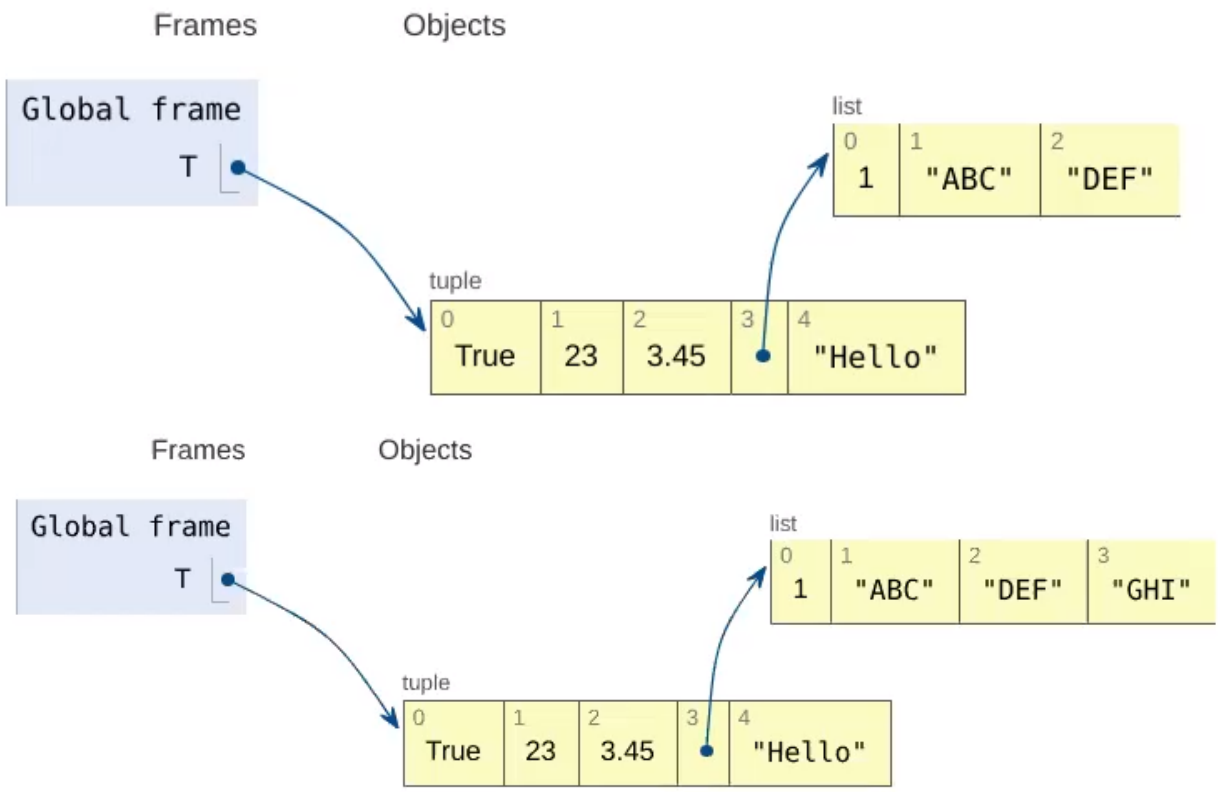



```
80  # python just creates a list in your RAM: [1, "ABC", "DEF"]
81  # it returns a sort of memory location from where the list starts.
```

Program

```
1  T = (True, 23, 3.45, [1, "ABC", "DEF"], "Hello")
2  T[3].append()
```

Output



Dictionary

```
113   # an english dictionary had a word and its meaning
114   # a python dictionary is the same except that is a key and a value
115
116   # a python dictionary is an unordered sequence of keys and their values
      enclosed in { }
117
118   # Dictionary = dict()
119   # Dictionary = {}
120   # print(type(Dictionary))
```

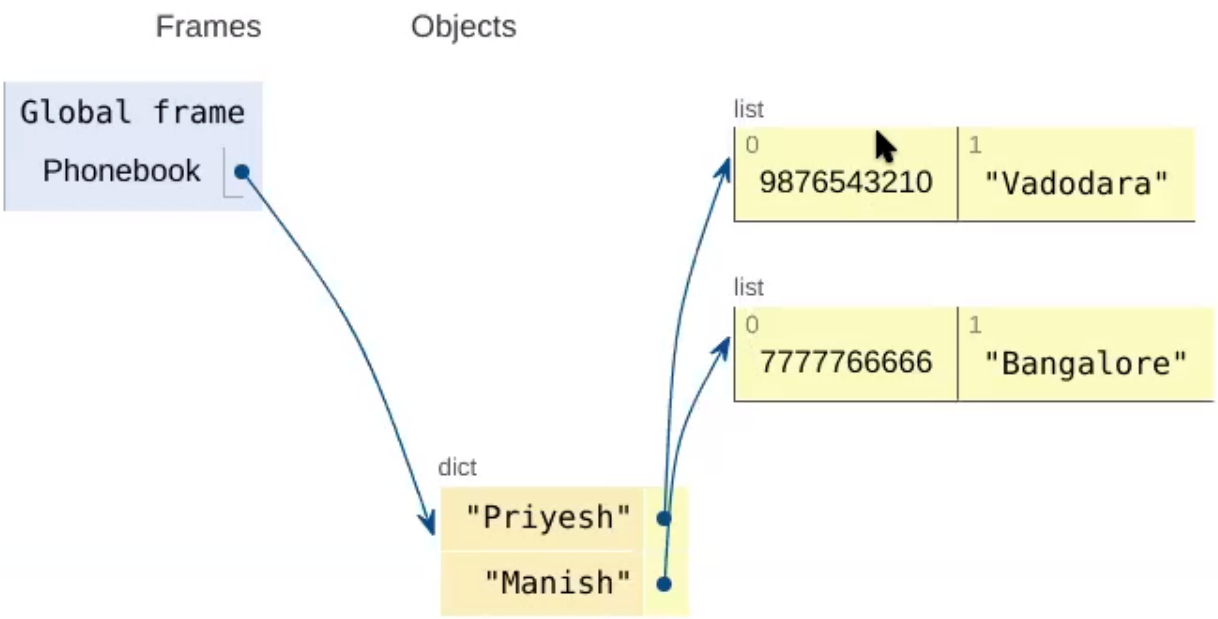
```
123   Phonebook = {"Priyesh": [9876543210, "Vadodara"], "Manish": [7777766666,
      "Bangalore"]}
124
125   Phonebook2 = {"Manish": [7777766666, "Bangalore"], "Priyesh": [9876543210,
      "Vadodara"]}
126
127   print(Phonebook==Phonebook2)
```

```
129   # Since dictionaries are unordered they have no indexing
130
131   # So, to look for meanings of words in an english dictionary, you will
      find the meaning using the word and not page number+paragraph order
```

Program

```
1  Phonebook = {"Priyesh": [9876543210, "Vadodara"], "Manis
2  details_Priyesh = Phonebook["Priyesh"]
```

Output




```
142   # In a dictionary keys and values can have different datatypes.
```

Program

```
146   AgeLog = {"Vaibhav": 30, "Megha": 28, "Amol": 26, "Priyesh": 25}
150
151   # how to add data into a dictionary: append()/insert()
152   # DictionaryName[Key] = Value
153   AgeLog["Sravan"] = 24
154   print(AgeLog)
155
156   AgeLog["Priyesh"] = 22
157   print(AgeLog)
158
159   # In a dictionary if you want to add a new value, you need a unique key of
      an immutable datatype
```

Output

```
{'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 25, 'Sravan': 24}
{'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 22, 'Sravan': 24}
~/RichBiodegradableLicenses$
```

Program

```
170   # Delete all items from a dictionary
171   # AgeLog = {}
172
173   # delete the entire dictionary
174   # del AgeLog
175
176   # extend() from lists is update in dictionary
177   AgeLog.update(AgeLog2)
178   print(AgeLog)
```

Output

```
{'Vaibhav': 30, 'Megha': 28, 'Amol': 26, 'Priyesh': 22, 'Shubham': 30, 'Div
yam': 30}
```

Program

```
182    print(AgeLog.get("Abhishek", None))
```

Output

```
None
~/RichBiodegradableLicenses$
```

MCQs

What is the output:

```
L = [1 , 2, 3, 4]
for I in range(0,4):
   if I == 2:
      break
   L[I], L[4-I] = L[4-I], L[I]
L2 = [1,2,3,4]
L2. reverse()
print(L==L2)
```

Attempted - 32 (61.54%) | EASY

- [ ] True 15.63%
- [ ] False 12.5%
- [ ] None
- [x] error 75%

Can a tuple contain items of mutable datatypes? If so, can we perform operations on these items that edit their content?

Attempted - 31 (59.62%) EASY

☑ T, T	80.65%
☐ T, F	9.68%
☐ F, T	6.45%
☐ F, F	6.45%

Is the following dictionary valid:
Phonebook = {{"Priyesh":"Vadodara"}: 9876543210, {"Priyesh": "Bangalore"}: 8765432109, "Amol": 7654321098, "Durjoy": False}

Attempted - 33 (63.46%) EASY

☐ True	12.12%
☑ False	87.88%

What is the output of:
Phonebook = {("Priyesh", "Vadodara"): 9876543210, ("Priyesh", "Bangalore"): 8765432109, "Amol": 7654321098, "Durjoy": False}
del Phonebook
print(Phonebook)

Attempted - 31 (59.62%) EASY

☐ {("Priyesh", "Vadodara"): 9876543210, ("Priyesh", "Bangalore"): 8765432109, "Amol": 7654321098, "Durjoy": False}	12.9%
☑ Error	67.74%
☐ {}	12.9%
☐ None	6.45%