Python Collections

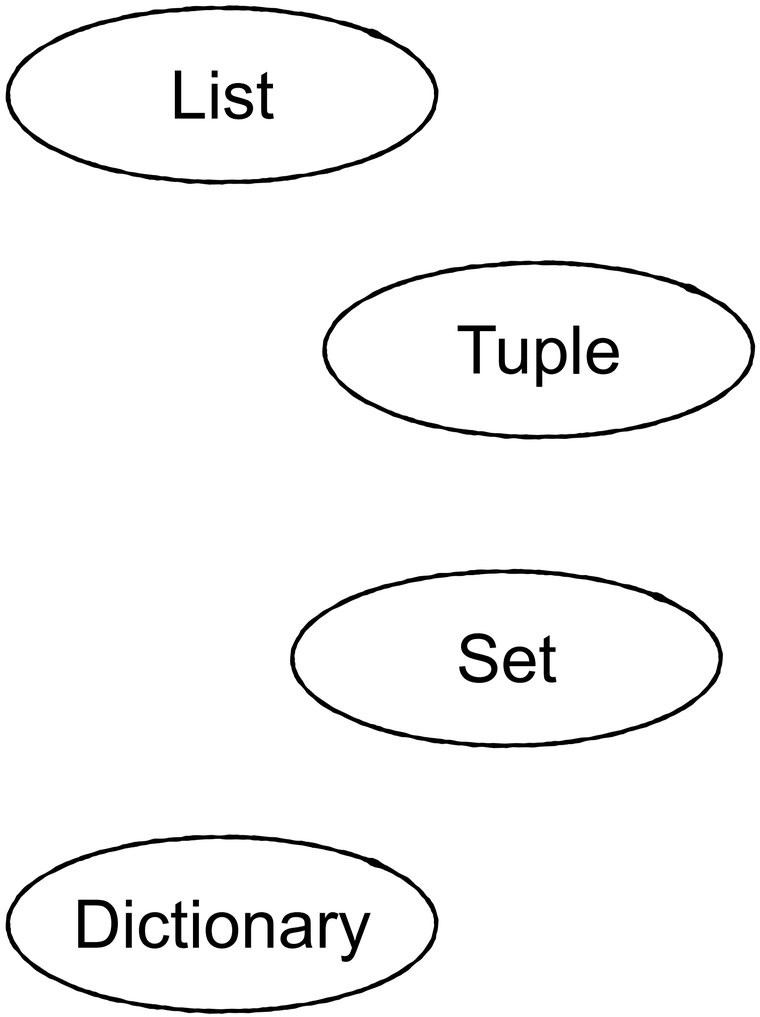
GopalKumar Katariya, 210133062005

Outline

- What are collections in python ?
  - List, Tuple, Set, Dictionary
- Specialised collection data types
  - namedtuple()
  - Duque
  - ChainMap
  - Counter
  - OrderedDict
  - defaultdict
  - UserDict
  - UserList
  - UserString

What are collections in python ?

There are four collections data type in python witch are used to store collections of data.



List

- Lists are used to store multiple items in a single variable.
- Lists are one of 4 built-in data types in Python used to store collections of data, the other 3 are Tuple, Set, and Dictionary, all with different qualities and usage.
- Lists are created using square brackets.

Tuple

- Tuples are used to store multiple items in a single variable.
- Tuple is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Set, and Dictionary, all with different qualities and usage.
- A tuple is a collection which is ordered and **unchangeable**.
- Tuples are written with round brackets.

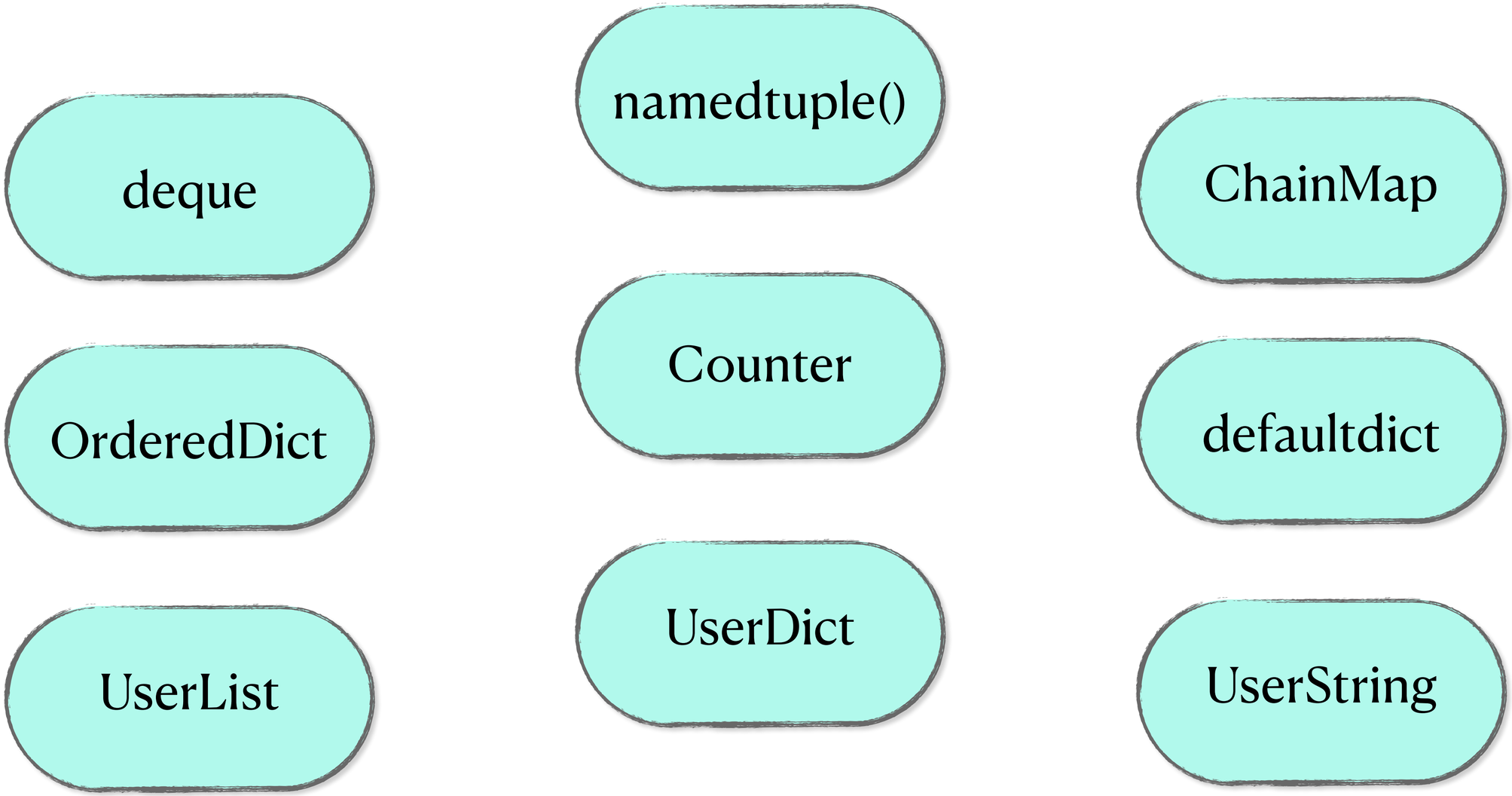
Set

- Sets are used to store multiple items in a single variable.
- Set is one of 4 built-in data types in Python used to store collections of data, the other 3 are List, Tuple, and Dictionary, all with different qualities and usage.
- A set is a collection which is *unordered*, *unchangeable**, and *unindexed*.
- Sets are written with curly brackets.

Dictionary

- Dictionaries are used to store data values in key:value pairs.
- A dictionary is a collection which is ordered*, changeable and do not allow duplicates.
- As of Python version 3.7, dictionaries are *ordered*. In Python 3.6 and earlier, dictionaries are *unordered*.
- Dictionaries are written with curly brackets, and have keys and values:

Specialised collection data types



namedtuple()

nametuple() returns a value for each element in the tuple.

```
from collections import namedtuple
tpl_a = namedtuple('courses','name, technology')
tpl_b = tpl_a('MLAI','python')
list_b = tpl_a._make([‘AI','R'])
```

```
cources(name='AI', technology='R')
```

deque

deque pronounced as "deck' is an optimised list to perform insertion and deletion easily.

```
from collections import deque
tpl = [‘g’,'o','p','a','l']
dq_tpl =  dq(tpl)
dq_tpl
```

```
deque(['g', 'o', 'p', 'a', 'l'])
```

```
dq_tpl.appendleft(‘katariya')
dq_tpl
```

```
deque([‘katariya', 'g', 'o', 'p', 'a', 'l'])
```

Chainmap

Chainmap is a dictionary like class for creating a single view of multiple mappings.

```
from collections import ChainMap
a = {"name":"gopal", “surname":"katariya"}
b = {"branch":"MLAI", “class":"B"}
ChainMap(a,b)
```

```
ChainMap({'name': 'gopal', 'surname': 'katariya'}, {'branch': 'MLAI', 'class': 'B'})
```

Counter

Counter is a dictionary subclass for counting hashable objects.

```
from collections import Counter
list_a = [1,1,2,3,4,4,5,5,5,6,6,7,7,7,8,8,8,9]
counter_list = Counter(list_a)
counter_list
```

```
Counter({1: 2, 2: 1, 3: 1, 4: 2, 5: 3, 6: 2, 7: 3, 8: 3, 9: 1})
```

OrderedDict

OrderedDict is dictionary subclass which remembers the order in which the entries were done.

```
from collections import OrderedDict
d =  OrderedDict()
d[0] = ‘g'
d[1] = ‘o'
d[2] = ‘p'
d[3] = ‘a'
d[4] = ‘l’
print(d)
d[0] = 'katariya'
print(d)
```

```
OrderedDict([(0, 'g'), (1, 'o'), (2, 'p'), (3, 'a'), (4, 'l')])
OrderedDict([(0, 'katariya'), (1, 'o'), (2, 'p'), (3, 'a'), (4, 'l')])
```

defaultdict

defaultdict is a dictionary subclass which calls a factory function to supply missing values.

```
from collections import defaultdict
d = defaultdict(int)
d[0] = ‘Python'
d[1] = ‘Java'
d[2] = ‘R’
print(d[3])
print(d[0])
```

```
0
Python
```

UserDict, UserList, UserString

UserDict is a wrapper around dictionary objects for easier dictionary sub-classing.

UserList is a wrapper around list objects for easier List sub-classing.

UserString is a wrapper around string objects for easier string sub-classing.

Thank You