Chapter 14 Tuples, Sets, and Dictionaries

1

A list is mutable, but a tuple is immutable. To create a tuple from a list, use tuple(list). To create a list from a tuple, use list(tuple).

2

You cannot add an element to a tuple, cannot delete an element from a tuple, and cannot replace an element in a tuple.

3

The statement t1 = t2 assigns t2 to t1. Now t1 and t2 refer to the same tuple.

4

```
(1, 2, 3, 7, 9, 0, 5)
1
(2, 3)
5
(1, 2, 3, 7, 9, 0)
(2, 3, 7, 9, 0)
```

5

```
9
0
27
7
```

6

```
False
True
False
True
```

7

To create an empty set, use s = set().

8

Yes.

9

```
s = {1, 3, 4}    # Correct
s = {{1, 2}, {4, 5}} # Incorrect, because sets {1, 2}, {4, 5} are mutable
s = {[1, 2], [4, 5]} # Incorrect, because lists {1, 2}, {4, 5} are mutable
s = {(1, 2), (4, 5)} # Correct (1, 2), (4, 5) are tuples. They are mutable
```

10

Sets are like lists to store a collection of items. Unlike lists, the elements in a set are unique and are not placed in any particular ordered. To create a set from a list, use set(list). To create a list from a set, use list(set).

11

```
{'john', 'peter'}
{'john', 'peter'}
{'peterson', 'john', 'peter'}
{'peterson', 'john'}
```

12

Since "johnson" is not in the set, invoking remove("Johnson") will throw an exception.

13

```
True
False
False
False
True
True
```

14

```
4
6
1
16
```

15

```
{1, 3, 4, 5, 6, 7}
{1, 3, 4, 5, 6, 7}
{1, 6}
{1, 6}
{4, 5}
{4, 5}
{3, 4, 5, 7}
{3, 4, 5, 7}
```

16.

```
False
True
4
11
2
23
True
True
```

17.

```
     {1, 2, 3} {3, 4, 5} {1, 2, 3, 4, 5}
     {1, 2, 3} {3, 4, 5} {1, 2}
     {1, 2, 3} {3, 4, 5} {3}
     {1, 2, 3} {3, 4, 5} {1, 2, 4, 5}
```

18.

```
     d = {} or d = dict()
```

19.

```
d = {1:[1, 2], 3:[3, 4]}  # Correct
#d = {[1, 2]:1, [3, 4]:3}  # Incorrect, key must be immutable
d = {(1, 2):1, (3, 4):3}  # Correct
d = {1:"john", 3:"peter"} # Correct
d = {"john":1, "peter":3} # Correct
```

20

They are called key and value.

21

     a. Assign 5 to associate with key "susan"
     b. Assign 5 to associate with key "peter"
     c. Add the value for key "peter" by 5
     d. Del the entry with key "peter"

22

```
     a. print(len(students)) # print 2

     b. print(students.keys()) # print keys 'john', 'peter'

     c. print(students.values()) # print values 3, 2

     d. print(students.items()) # print items ('john', 3),
     ('peter', 2)
```

23.

```
     4
     ['blue', 'green', 'yellow', 'red']
     [1, 14, 2, 4]
     True
     False
     11
```

24.

     4

25

The get(key) method is similar to dictionary_name[key] except that the get method returns None if the key is not in the dictionary rather than raising an exception.