



[Course](#) > [Chapter Ten: Tuples](#) > [Review: Chapter 10](#) > Chapter 10 Quiz

Chapter 10 Quiz

Question 1

1/1 point (graded)

What is the difference between a Python tuple and Python list?

- ☐ Tuples can be expanded after they are created and lists cannot
- ☐ Lists maintain the order of the items and tuples do not maintain order
- ☐ Lists are indexed by integers and tuples are indexed by strings
- ☒ Lists are mutable and tuples are not mutable ✓

Submit

Question 2

1/1 point (graded)

Which of the following methods work both in Python lists and Python tuples?

☐ sort()

☐ append()

☐ pop()

☐ reverse()

☒ index() ✓

Submit

Question 3

1/1 point (graded)

What will end up in the variable **y** after this code is executed?

```
x , y = 3, 4
```

☐ A two item list

☐ 3

☐ A two item tuple

☐ A dictionary with the key 3 mapped to the value 4

☒ 4 ✓

Submit

Question 4

1/1 point (graded)

In the following Python code, what will end up in the variable **y**?

```
x = { 'chuck' : 1 , 'fred' : 42, 'jan': 100}  
y = x.items()
```

☐ A tuple with three integers

☐ A list of strings

☐ A list of integers

☒ A list of tuples ✓

Submit

Question 5

1/1 point (graded)

Which of the following tuples is greater than **x** in the following Python sequence?

```
x = (5, 1, 3)
if ??? > x :
    ...
```

☐ (0, 1000, 2000)

☐ (4, 100, 200)

☐ (5, 0, 300)

☒ (6, 0, 0) ✓

Submit

Question 6

1/1 point (graded)

What does the following Python code accomplish, assuming the **c** is a non-empty dictionary?

```
tmp = list()
for k, v in c.items() :
    tmp.append( (v, k) )
```

☐ It sorts the dictionary based on its key values

☐ It computes the average of all of the values in the dictionary

☐ It computes the largest of all of the values in the dictionary

☒ It creates a list of tuples where each tuple is a value, key pair ✓

Submit

Question 7

1/1 point (graded)

If the variable **data** is a Python list, how do we sort it in reverse order?

☐ `data = sortrev(data)`

☐ `data.sort.reverse()`

☐ `data = data.sort(-1)`

☒ `data.sort(reverse=True)` ✓

Submit

Question 8

1/1 point (graded)

Using the following tuple, how would you print 'Wed'?

```
days = ('Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat', 'Sun')
```

- ☐ `print(days(2))`
- ☐ `print(days{2})`
- ☐ `print[days(2)]`
- ☐ `print(days[1])`
- ☐ `print(days.get(1,-1))`
- ☒ `print(days[2])` ✓

Submit

Question 9

1/1 point (graded)

In the following Python loop, why are there two iteration variables (k and v)?

```
c = {'a':10, 'b':1, 'c':22}
for k, v in c.items() :
    ...
```

- ☐ Because the keys for the dictionary are strings
- ☐ Because for each item we want the previous and current key
- ☐ Because there are two items in the dictionary
- ☒ Because the items() method in dictionaries returns a list of tuples ✓

Submit

Question 10

1/1 point (graded)

Given that Python lists and Python tuples are quite similar - when might you prefer to use a tuple over a list?

- ☐ For a list of items that want to use strings as key values instead of integers
- ☐ For a list of items you intend to sort in place
- ☐ For a list of items that will be extended as new items are found
- ☒ For a temporary variable that you will use and discard without modifying ✓

Submit

© All Rights Reserved