**Week 6 : Quiz**

Question 1

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



Lists are mutable and tuples are not mutable



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

**Ans : Lists are mutable and tuples are not mutable**

Question 2

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



sort()



reverse()



pop()



index()



append()

**Ans : index()**

Question 3

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



1

x , y = 3, 4



4



3



A two item list



A two item tuple



A dictionary with the key 3 mapped to the value 4

**Ans : 4**

Question 4

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



1

2

x = { 'chuck' : 1 , 'fred' : 42, 'jan': 100}

y = x.items()



A list of integers



A list of tuples



A tuple with three integers



A list of strings

**Ans : A list of tuples**

Question 5

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



1

2

3

x = (5, 1, 3)

if ??? > x :

...



(6, 0, 0)



(5, 0, 300)



(4, 100, 200)



(0, 1000, 2000)

**Ans : (6, 0, 0)**

Question 6

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



1

2

3

tmp = list()

for k, v in c.items() :

tmp.append( (v, k) )



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



It sorts the dictionary based on its key values



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



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

**Ans : It creates a list of tuples where each tuple is a value, key pair**

Question 7

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



data = sortrev(data)



data = data.sort(-1)



data.sort(reverse=True)



data.sort.reverse()

**Ans : data.sort(reverse=True)**

Question 8

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



1

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



print(days{2})



print[days(2)]



print(days[1])



print(days.get(1,-1))



print(days(2))



print(days[2])

**Ans : print(days[2])**

Question 9

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



1

2

3

c = {'a':10, 'b':1, 'c':22}

for k, v in c.items() :

...



Because the items() method in dictionaries returns a list of tuples



Because the keys for the dictionary are strings



Because there are two items in the dictionary



Because for each item we want the previous and current key

**Ans : Because the items() method in dictionaries returns a list of tuples**

Question 10

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 you intend to sort in place



For a list of items that will be extended as new items are found



For a list of items that want to use strings as key values instead of integers



For a temporary variable that you will use and discard without modifying

**Ans : For a temporary variable that you will use and discard without modifying**