Quiz - Dictionaries

Due Oct 31 at 11:59pm **Points** 1 **Questions** 10
Available Oct 24 at 8am - Oct 31 at 11:59pm 8 days **Time Limit** None
Allowed Attempts Unlimited

Instructions

Try to answer the questions without using your interpreter first to check your understanding.

You may verify your answer with the interpreter when you think you've got it.

Take the Quiz Again

Attempt History

	Attempt	Time	Score
LATEST	Attempt 1	14 minutes	0.7 out of 1

⚠ Correct answers will be available on Nov 2 at 8am.

Score for this attempt: **0.7** out of 1
Submitted Oct 25 at 6:30pm
This attempt took 14 minutes.

Question 1 — 0.1 / 0.1 pts

Consider the following dictionary:

words = {1: 'one', 2: 'two', 3: 'three'}

What is words[3]?

- {1: 'one', 2: 'two', 3: 'three'}
- (•) 'three'

undefined - the index is out of bounds

3: 'three'

Dictionaries are indexed by key not by position.

Question 2

0.1 / 0.1 pts

What do the following lines of code print?

```
some_dictionary = {1:2, 2:3, 3:0}

for i in sorted (some_dictionary.values()):
    print(i)
```

- 0
  2
  3 (selected)
- 3
  2
  0
- 1
  2
  3
- 2
  3
  0

some_dictionary.values() contains the values in the dictionary: 2, 3 and 0.

Question 3

0.1 / 0.1 pts

What do the following lines of code print?

```
some_dictionary = {1:2, 2:3, 3:0}
for i in sorted (some_dictionary, reverse = True):
    print (i)
```

- 0
  2
  3
- 3
  2
  0
- 3
  2
  1 (selected)
- 1
  2
  3

Iterating over a dictionary is equivalent to iterating over the keys.

When we call sorted() on a dictionary, we get a sorted list of the keys.

Incorrect

Question 4

0 / 0.1 pts

A dictionary is another sequence data type.

- False
- True (selected)

Incorrect

Question 5

0 / 0.1 pts

A set is a valid dictionary key.

- False
- (•) True

Incorrect

Question 6

0 / 0.1 pts

A list is a valid dictionary value.

- True
- (•) False

Question 7

0.1 / 0.1 pts

Consider the following dictionary:

grades = { 'Brian': [100, 95, 99], 'Alex': [89, 98, 100]}

How do I access the value 89 in this dictionary?

- grades[1][0]
- (•) grades['Alex'][0]
- grades[1]
- grades[0]['Alex']

grades['Alex']

grades is a dictionary. To access the list corresponding to 'Alex', we write: grades['Alex'].

To access the first item in that list, we write:

grades['Alex'][0]

Question 8

0.1 / 0.1 pts

Consider the following dictionary:

grades = { 'Brian': [100, 95, 99], 'Alex': [89, 98, 100]}

What is the value of the following expression?

sorted(grades, key=lambda student: grades[student][1])

- [98, 95]
- ['Alex', 'Brian']
- [95, 98]
- ['Brian', 'Alex'] (selected)

The sorted key argument here is a lambda function that returns the middle grade of a given student.

So the proxy value for 'Alex' is grades['Alex'][1] which is 98.

The proxy value for 'Brian' is grades['Brian'][1] which is 95.

Since 95 is less than 98, Brian comes first in the sort.

Question 9

0.1 / 0.1 pts

Consider the following dictionary:

grades = { 'Brian': [100, 95, 99], 'Alex': [89, 98, 100]}

What is the value of the following expression?

sorted(grades, key=lambda student: sum(grades[student]), reverse=True)

- (●) ['Brian', 'Alex']
- ( ) [294, 287]
- ( ) ['Alex', 'Brian']
- ( ) [100, 100]
- ( ) [287, 294]

The sorted key argument here is a lambda function that returns the sum of the grades of a given student.

So the proxy value for 'Alex' is sum([89, 98, 100]) which is 287.

The proxy value for 'Brian' is 294.

Since we specified reverse, we'll get ['Brian', 'Alex']

Question 10

0.1 / 0.1 pts

Consider the following dictionary:

gpa = { 'Michael': 4, 'Amy': 2, 'Zoe': 3}

What is the value of the following expression?

max(gpa, key=gpa.get)

- ( ) 'Amy'
- ( ) 4

- (•) 'Michael'
- ( ) 2
- ( ) 3
- ( ) 'Zoe'

gpa.get(student) returns the value associated with a given student in the dictionary. That's the student's gpa.

The comparison is then based on the proxy values 4, 2 and 3. Since 4 is the greatest proxy value, the key corresponding to the value 4 ('Michael') is the maximum here.

Quiz Score: **0.7** out of 1