



✓ **Congratulations! You passed!**

TO PASS 70% or higher

Keep Learning

GRADE
85.71%

Dictionaries

LATEST SUBMISSION GRADE

85.71%

1. Which of the following expressions corresponds to a dictionary with no elements?

1 / 1 point

☒ {}

✓ Correct

☐ None

☒ dict()

✓ Correct

☐ []

2. Given an existing dictionary **favorites**, what Python statement adds the key "**fruit**" to this dictionary with the corresponding value "**blackberry**"?

1 / 1 point

☐ favorites["fruit" = "blackberry"]

☐ favorites{"fruit" : "blackberry"}

☐ favorites["fruit" : "blackberry"]

☐ favorites = {"fruit" : "blackberry"}

☒ favorites["fruit"] = "blackberry"

✓ Correct

3. Which of the expressions below returns **True** when the dictionary **my_dictionary** contains the key **my_key** and **False** otherwise?

1 / 1 point

☐ my_dictionary contains my_key

☐ my_dictionary.has_key(my_key)

☒ my_key in my_dictionary

✓ Correct

☐ my_dictionary.my_key()

4. Keys in a dictionary can have which of the following types?

0 / 1 point

☒ dict

! This should not be selected
Dictionaries are mutable.

☒ bool

✓ Correct
Booleans are immutable.

☐ list

☒ int

✓ Correct
Integers are immutable.

5. Values in a dictionary can have which of the following types?

1 / 1 point

✓ int

✓ Correct

✓ float

✓ Correct

✓ dict

✓ Correct

✓ bool

✓ Correct

6. Consider the following dictionary:

1 / 1 point

```
1 instructor_ratings = {"Joe" : "awesome", "Scott" : "hmm..."}
```

What happens when Python evaluates the expression `instructor_ratings["John"]`?

- ☒ Since "John" is not a key in the dictionary, Python raises a `KeyError` exception.
- ☐ Since "John" is not a value in the dictionary, Python raises a `KeyError` exception.
- ☐ Since "John" is not a key in the dictionary, Python raises a syntax error.
- ☐ Python returns the value `None` since "John" is not a key in the dictionary.

✓ Correct

7. Write a function `count_letters(word_list)` that takes as input a list of words that are composed entirely of lower case letters. This function should return the lower case letter that appears most frequently (total number of occurrences) in the words in `word_list`. (In the case of ties, return the earliest letter in alphabetical order.)

1 / 1 point

The Python code snippet below represents a start at implementing `count_letters` using a dictionary `letter_count` whose keys are the lower case letters and whose values are the corresponding number of occurrences of each letter in the strings in `word_list`.

```
1 def count_letters(word_list):
2     """ See question description """
3
4     ALPHABET = "abcdefghijklmnopqrstuvwxyz"
5
6     letter_count = {}
7     for letter in ALPHABET:
8         letter_count[letter] = 0
9
10    # enter code here
```

Complete your implementation of `count_letters` based on this snippet. As a test, `count_letters(["hello", "world"])` should return the letter 'l' since 'l' appears 3 times total in the strings "hello" and "world".

When you are confident in your code, compute the lower case letter return by `count_letters(monty_words)` where `monty_words` is defined as shown.

```
1 monty_quote = "listen strange women lying in ponds distributing swords is no
2 basis for a system of government supreme executive power derives from a
3 mandate from the masses not from some farcical aquatic ceremony"
```

Enter this **single letter** in the text box below. Do not include any spaces or enclosing quotes around the letter.

e

✓ Correct
The letter 'e' appears 20 times in the quote.