course_2_assessment_3

Due: 2018-11-25 01:30:00

Description: Assessment for Dictionary Accumulation Lesson Score: 0 of 9 = 0.0%

Questions

Not yet graded

The dictionary `Junior` shows a schedule for a junior year semester. The key is the course name and the value is the number of credits. Find the total number of credits taken this semester and assign it to the variable `credits`. Do not hardcode this – use dictionary accumulation!

Save & Run 5/14/2021, 9:27:41 AM - 2 of 2 Show in CodeLens

```
1 Junior = {'SI 206':4, 'SI 310':4, 'BL 300':3, 'TO 313':3, 'BCOM 350':1, 'MO 300':3}
2 credits = 0
3 for v in Junior.values():
4     credits += v
5     print(credits)
```

```
4
8
11
14
15
18
```

ActiveCode (ac10_9_9)

Result	Actual Value	Expected Value	Notes
Pass	18	18	Testing that credits is assigned to correct values

You passed: 100.0% of the tests

Not yet graded

Create a dictionary, `freq`, that displays each character in string `str1` as the key and its frequency as the value.

Save & Run 5/14/2021, 9:28:55 AM - 3 of 3 Show in CodeLens

```
1 from collections import Counter
2 str1 = "peter piper picked a peck of pickled peppers"
3
4 freq = Counter(str1)
5
6 for i in str1:
7     print(i, freq[i])
```

```
p 9
e 8
t 1
e 8
r 3
  7
p 9
i 3
p 9
e 8
r 3
  7
p 9
i 3
c 3
k 3
e 8
d 2
  7
a 1
  7
p 9
e 8
c 3
k 3
```

```
  7
o 1
f 1
  7
p 9
i 3
c 3
k 3
l 1
e 8
d 2
  7
p 9
e 8
p 9
p 9
e 8
r 3
s 1
```

ActiveCode (ac10_9_10)

Result	Actual Value	Expected Value	Notes
Pass	[(' ..., 1)]	[(' ..., 1)]	Testing that freq is correct.

Expand Differences

You passed: 100.0% of the tests

Not yet graded

Provided is a string saved to the variable name `s1`. Create a dictionary named `counts` that contains each letter in `s1` and the number of times it occurs.

Save & Run 5/14/2021, 9:29:19 AM - 2 of 2 Show in CodeLens

```
s1 = "hello"
def char_frequency(s1):
    dict = {}
    for n in s1:
        keys = dict.keys()
        if n in keys:
            dict[n] += 1
        else:
            dict[n] = 1
    return dict

counts = char_frequency(s1)
print(counts)

```

```
{'e': 1, 'o': 1, 'l': 2, 'h': 1}
```

ActiveCode (ac10_9_11)

Result	Actual Value	Expected Value	Notes
Pass	[('e'..., 1)]	[('e'..., 1)]	Testing that counts was created correctly.

Expand Differences

You passed: 100.0% of the tests

Not yet graded

Create a dictionary, `freq_words`, that contains each word in string `str1` as the key and its frequency as the value.

Save & Run 5/14/2021, 9:29:46 AM - 2 of 2 Show in CodeLens

```
str1 = "I wish I wish with all my heart to fly with dragons in a land apart"
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts

freq_words = word_count(str1)
print(freq_words)
```

```
{'a': 1, 'I': 2, 'wish': 2, 'with': 2, 'all': 1, 'my': 1, 'heart': 1, 'to': 1, 'fly': 1, 'dragons':
```

ActiveCode (ac10_9_12)

Result	Actual Value	Expected Value	Notes
Pass	[('I'..., 2)]	[('I'..., 2)]	Testing that freq_words was created correctly.

Expand Differences

You passed: 100.0% of the tests

Not yet graded

Create a dictionary called `wrd_d` from the string `sent`, so that the key is a word and the value is how many times you have seen that word.

Save & Run 5/14/2021, 9:30:03 AM - 2 of 2 Show in CodeLens

```
sent = "Singing in the rain and playing in the rain are two entirely different situation
def word_count(str):
    counts = dict()
    words = str.split()

    for word in words:
        if word in counts:
            counts[word] += 1
        else:
            counts[word] = 1

    return counts
wrd_d  = word_count(sent)
print(wrd_d)
```

```
{'in': 2, 'Singing': 1, 'the': 2, 'rain': 2, 'and': 1, 'playing': 1, 'are': 1, 'two': 1, 'entirely':
```

ActiveCode (ac10_9_13)

Result	Actual Value	Expected Value	Notes
Pass	[('Si..., 1)]	[('Si..., 1)]	Testing that wrd_d has been created correctly.

Expand Differences

You passed: 100.0% of the tests

Not yet graded

Create the dictionary `characters` that shows each character from the string `sally` and its frequency. Then, find the most frequent letter based on the dictionary. Assign this letter to the variable `best_char`.

Save & Run 5/14/2021, 9:30:22 AM - 2 of 2 Show in CodeLens

```
sally = "sally sells sea shells by the sea shore"
characters = {}
for i in sally:
    characters[i]=characters.get(i,0)+1

sorted(characters.items(), key=lambda x: x[1])
best_char = sorted(characters.items(), key=lambda x: x[1])[-1][0]

```

ActiveCode (ac10_9_14)

Result	Actual Value	Expected Value	Notes
Pass	[(' '..., 2)]	[(' '..., 2)]	Testing that characters has correct values.
Pass	's'	's'	Testing that best_char is assigned to correct value.

Expand Differences

You passed: 100.0% of the tests

Not yet graded

Find the least frequent letter. Create the dictionary `characters` that shows each character from string `sally` and its frequency. Then, find the least frequent letter in the string and assign the letter to the variable `worst_char`.

Save & Run 5/14/2021, 9:30:38 AM - 2 of 2 Show in CodeLens

```
sally = "sally sells sea shells by the sea shore and by the road"
characters = {}
for i in sally:
    characters[i]=characters.get(i,0)+1

sorted(characters.items(), key=lambda x: x[1])
worst_char = sorted(characters.items(), key=lambda x: x[1])[-13][0]

```

ActiveCode (ac10_9_15)

Result	Actual Value	Expected Value	Notes
Pass	[(' '..., 3)]	[(' '..., 3)]	Testing that characters has been updated correctly.
Pass	'n'	'n'	Testing that worst_char is assigned to correct value.

Expand Differences

You passed: 100.0% of the tests

Not yet graded

Create a dictionary named `letter_counts` that contains each letter and the number of times it occurs in `string1`. **Challenge:** Letters should not be counted separately as upper-case and lower-case. Intead, all of them should be counted as lower-case.

Save & Run 5/14/2021, 9:36:58 AM - 4 of 4 Show in CodeLens

```
string1 = "There is a tide in the affairs of men, Which taken at the flood, leads on to fo
string1.lower()

letter_counts = {}

for c in string1.lower():
    if c not in letter_counts:
        letter_counts[c] = 0
    letter_counts[c] = letter_counts[c] + 1

```

ActiveCode (ac10_9_16)

Result	Actual Value	Expected Value	Notes
Pass	17	17	Testing that the letter 'a' has the correct value.
Pass	19	19	Testing that the letter 't' has the correct value.
Pass	17	17	Testing that the letter 'o' has the correct value.
Pass	6	6	Testing that the letter 'w' has the correct value.

You passed: 100.0% of the tests

Not yet graded

Create a dictionary called `low_d` that keeps track of all the characters in the string `p` and notes how many times each character was seen. Make sure that there are no repeats of characters as keys, such that "T" and "t" are both seen as a "t" for example.

Save & Run 5/14/2021, 9:37:36 AM - 8 of 8 Show in CodeLens

```
p = "Summer is a great time to go outside. You have to be careful of the sun though becaus
p.lower()

low_d = {}

for c in p.lower():
    if c not in low_d:
        low_d[c] = 0
    low_d[c] = low_d[c] + 1

```

ActiveCode (ac10_9_17)

Result	Actual Value	Expected Value	Notes
Pass	5	5	Testing the key s
Pass	1	1	Testing the key y

You passed: 100.0% of the tests

Score Me