

[Course](#) > [Week 4...](#) > [Proble...](#) > Problem...

Audit Access Expires Aug 9, 2019

You lose all access to this course, including your progress, on Aug 9, 2019.

Problem 2 - Dealing with Hands

Problem 2 - Dealing with Hands

0.0/10.0 points (graded)

****Please read this problem entirely!!**** The majority of this problem consists of learning how to read code, which is an incredibly useful and important skill. At the end, you will implement a short function. Be sure to take your time on this problem - it may seem easy, but reading someone else's code can be challenging and this is an important exercise.

Representing hands

A **hand** is the set of letters held by a player during the game. The player is initially dealt a set of random letters. For example, the player could start out with the following hand:

`a, q, l, m, u, i, l`. In our program, a hand will be represented as a dictionary: the keys are (lowercase) letters and the values are the number of times the particular letter is repeated in that hand. For example, the above hand would be represented as:

```
hand = {'a':1, 'q':1, 'l':2, 'm':1, 'u':1, 'i':1}
```

Notice how the repeated letter `'l'` is represented. Remember that with a dictionary, the usual way to access a value is `hand['a']`, where `'a'` is the key we want to find.

However, this only works if the key is in the dictionary; otherwise, we get a `KeyError`. To avoid this, we can use the call `hand.get('a', 0)`. This is the "safe" way to access a value if



we are not sure the key is in the dictionary. `d.get(key,default)` returns the value for `key` if `key` is in the dictionary `d`, else `default`. If `default` is not given, it returns `None`, so that this method never raises a `KeyError`. For example:

```
>>> hand['e']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'e'
>>> hand.get('e', 0)
0
```

Converting words into dictionary representation

One useful function we've defined for you is `getFrequencyDict`, defined near the top of `ps4a.py`. When given a string of letters as an input, it returns a dictionary where the keys are letters and the values are the number of times that letter is represented in the input string. For example:

```
>>> getFrequencyDict("hello")
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
```

As you can see, this is the same kind of dictionary we use to represent hands.

Displaying a hand

Given a hand represented as a dictionary, we want to display it in a user-friendly way. We have provided the implementation for this in the `displayHand` function. Take a few minutes right now to read through this function carefully and understand what it does and how it works.

Generating a random hand

The hand a player is dealt is a set of letters chosen at random. We provide you with the implementation of a function that generates this random hand, `dealHand`. The function takes as input a positive integer `n`, and returns a new object, a hand containing `n` lowercase letters. Again, take a few minutes (right now!) to read through this function carefully and understand what it does and how it works.

Removing letters from a hand (you implement this)



The player starts with a hand, a set of letters. As the player spells out words, letters from this set are used up. For example, the player could start out with the following hand: `a, q, l, m, u, i, l`. The player could choose to spell the word `quail`. This would leave the following letters in the player's hand: `l, m`. Your task is to implement the function `updateHand`, which takes in two inputs - a `hand` and a `word` (string). `updateHand` uses letters from the hand to spell the word, and then returns a copy of the `hand`, containing only the letters remaining. For example:

```
>>> hand = {'a':1, 'q':1, 'l':2, 'm':1, 'u':1, 'i':1}
>>> displayHand(hand) # Implemented for you
a q l l m u i
>>> hand = updateHand(hand, 'quail') # You implement this function!
>>> hand
{'a':0, 'q':0, 'l':1, 'm':1, 'u':0, 'i':0}
>>> displayHand(hand)
l m
```

Implement the `updateHand` function. Make sure this function has no side effects: i.e., it must not mutate the hand passed in. Before pasting your function definition here, be sure you've passed the appropriate tests in `test_ps4a.py`.

Hints

Testing

Testing: Make sure the `test_updateHand()` tests pass. You will also want to test your implementation of `updateHand` with some reasonable inputs.

Copying Dictionaries

You may wish to review the `".copy"` method of Python dictionaries (review this and other Python dictionary methods [here](#)).

Your implementation of `updateHand` should be short (ours is 4 lines of code). It does not need to call any helper functions.

```
1 def updateHand(hand, word):
2     """
3     Assumes that 'hand' has all the letters in word.
4     In other words, this assumes that however many times
5     a letter appears in 'word', 'hand' has at least as
6     many of that letter in it.
7
```

```

8     Updates the hand: uses up the letters in the given word
9     and returns the new hand, without those letters in it.
10
11     Has no side effects: does not modify hand.
12
13     word: string
14     hand: dictionary (string -> int)
15     returns: dictionary (string -> int)

```

Press ESC then TAB or click outside of the code editor to exit

Unanswered

Submit

You have used 0 of 30 attempts


Problem 2 - Dealing with Hands


Hide Discussion


Topic: Problem Set 4 / Problem 2


Show all posts ▼


by recent activity ▼


- 
[Apparently this can be done with a \(very ugly\) one-liner](#)
16 ▼


[Tried squeezing my original four-line solution into a one-liner. It works, but boy is it ugly.](#)
- 
[import functions of other files](#)
1 ▼

[iam trying to import funs from other file like ps4a.py and give it's returned value to another fun in afi...](#)
- 
[Best way to make a copy?](#)
8 ▼

[To make a copy of the hand I did `updated hand = hand.copy\(\)` but I'm not sure that's the right thin...](#)
- 
[getFrequencyDict function never used](#)
5 ▼

[I completed all parts of ps4a. However, not once did I use the helper function getFrequencyDict, not...](#)
- 
[Possible error in docstring of dealHand\(n\)](#)
2 ▼

[I think the function dealHand\(n\) has an error in the docstring. It says: > At least n/3 the letters in the...](#)
- 
[need to fix the position of two arguments?](#)
3 ▼

[Hi, I'm confused now - - I test the updateHand, but it said FAILURE because my return is None. I see t...](#)
- 
[Not mutating hand](#)
10 ▼

[Hello, Can someone explain to me why hand should not be mutated? Well, I have solved this proble...](#)