

# Computer Science 1 (../index.html) CPSC 1301K (../index.html)

## Dictionaries Practice Assignments

### Overview

Dictionaries are an amazing data structure that allow use to add and retrieve an item very quickly.

### Submission Instructions

For each of the following practice assignments, save your solution in a .py file, with the name being dicts and the number of the assignment. For example, for Dicts01: Names, save your solution in a file named dicts01.py. Then, submit that file the respective assignment on codePost.io (<https://codePost.io>).

To register for a free account, go to <https://codepost.io/signup/join?code=I5039NQNWJ> (<https://codepost.io/signup/join?code=I5039NQNWJ>). Register with your CSU email address. Sometimes it takes more than one try, so please work on this well before the deadline. Additionally, some students have had better luck with using the “Forgot password” link.

### Practice Assignments

#### Dicts01: Login Validation

Complete the passwordCheck() function that will authenticate a user's login by accepting that user's name, their password, and a dictionary of users' login credentials. The function will return the str 'login successful' if the user's login information is correct and the str 'login failed' otherwise.

**Examples:**

```
passwordCheck( 'user2', 'P@ssword2', {'user1':'password1', 'user2':'P@ssword2',  
'user3':'p#ssword33', 'user4':'P&ssw0rd4'} ) returns 'login successful'  
passwordCheck( 'user3', 'P#ssword33', {'user1':'password1', 'user2':'P@ssword2',  
'user3':'p#ssword33', 'user4':'P&ssw0rd4'} ) returns 'login failed'
```

Provided code:

```
def passwordCheck(username, password, loginCredentials)
```

## Dicts02: Character Frequency

Complete the function `charFrequency( )` so that it will accept a `str`, and use a dictionary to store each character's frequency. Return the dictionary.

Suggested steps:

- Create an empty dictionary
- Traverse the string of characters, one at a time. For each character, use the character as the key. If the character's not in the dictionary, then add it. Otherwise, increment it's count.
- Return the dictionary

### Examples:

`charFrequency( 'aaaabbbbccdddeehhhh' )` returns `{'a': 4, 'b': 5, 'c': 2, 'd': 3, 'e': 2, 'h': 4}`

`charFrequency( 'PythonRocks' )` returns `{'P': 1, 'y': 1, 't': 1, 'h': 1, 'o': 2, 'n': 1, 'R': 1, 'c': 1, 'k': 1, 's': 1}`

Provided code:

```
def charFrequency( characters ):  
  
    '''Enter code here'''  
  
print( charFrequency('aaaabbbbccdddeehhhh') )  
print( charFrequency('PythonRocks') )
```

## Dicts03: Most Common Name

Complete the function `mostCommonCount()` so that will accept a list of names as a parameter, and return the count of the most common name in the list.

### Examples:

`mostCommonCount( ['marquard', 'zhen', 'csev', 'zhen', 'cwen', 'marquard', 'zhen', 'csev', 'cwen', 'zhen', 'csev', 'marquard', 'zhen'] )` returns 5

`mostCommonCount( ['Jacob', 'Michael', 'Joshua', 'Matthew', 'Emily', 'Madison', 'Emma', 'Olivia', 'Hannah'] )` returns 1

Provided code:

```
def mostCommonCount(namesList):
```