

# 1 Dictionaries

## 1.1 Tasks to do

1. For all of the following tasks, we will have a dictionary with the following structure. We given an example first and then describe what it means.

```
1  d = {
2      "articles": [{
3          "slug": "how-to-train-your-mule",
4          "title": "How to train your mule",
5          "description": "Everwonderhow?",
6          "body": "It takes a Jacobian",
7          "tagList": ["mules", "training"],
8          "createdAt": "2016-02-18T03:22:56.637Z",
9          "updatedAt": "2016-02-18T03:48:35.824Z",
10         "favoritesCount": 0,
11         "author": {
12             "username": "jake",
13             "bio": "I work at statefarm",
14             "following": False
15         }
16     }, {
17         "slug": "and anotherarticle",
18         "body": "I'm getting bored",
19         "tagList": ["bored", "article"],
20         "createdAt": "2016-02-18T03:22:56.637Z",
21         "updatedAt": "2016-02-18T03:48:35.824Z",
22         "favoritesCount": 20,
23         "author": {
24             "username": "cap",
25             "following": True
26         }
27     }],
28     "tweets": [{
29         "body": "See my article on training mules.",
30         "author": {
31             "username": "jake"
32         }
33     }]
34 }
```

## 1.1 Tasks to do

2. Do not be overwhelmed by the dictionary. Let's explain this in detail. Above, we have a **dictionary of document types**. In `d`, we have only two document types: articles and tweets.

The value of each document type is a **list** which contains 0 or more documents. In this example, we have two articles (first one starting from line 2 and ending at line 16 and second one starting from line 16 and ending at line 27). We also have a single tweet (starting at line 28 and ending at line 33).

Each document within the list is represented as a dictionary and can have a lot of information. For example, the first article's title is *'How to TRAIN your mule'*. It also has an 'author' key – this is a special key since its value is also a dictionary. This inner dictionary will tell you the author's username and their bio, among other things. The author key is also present in documents of type tweet.

Read the description in detail and carefully to make sure you understand it fully. Then, read the description of tasks below. Do the tasks in order and they should be easier to complete.

- 3.(a) Write a function with the name `get_types`. This function will be given a dictionary following the above specifications and should simply return a list which has the types of the documents. For the above example, it should return `['tweets', 'articles']`. However, the order is not important here. It can also return `['articles', 'tweets']`.

Of course, your code should work for other dictionaries which follow the same semantics but might have different article types.

- (b) The second function should be called `get_types_counts`. It also takes in the dictionary as input and returns another dictionary. This returned key, value pair has the document types as keys and the number of their corresponding documents as their values. For the above example, the function should return `{ 'articles': 2, 'tweets': 1 }` since there are two articles and only one tweet. Obviously, the order is not important here.

- (c) Another function you need to write is called `get_author_count`. It also takes in the dictionary as above but also takes in an author's username as string. It returns the total number of documents authored by the person to whom this username belongs. Note that the document type is not important here – we are only interested in the total number of documents written by this person. For the above dictionary, if we pass in 'jake' as the username, the value returned should be 2 since the third document is written by someone else with the username 'cap'.

- (d) To get the final couple of tests to pass, you need to implement the following functionality: The function you defined in (c) above should be modified so that it can optionally take in a list of document types to limit the count to. For instance, we should be able to pass in a list of document types which should be counted for the given author. For example, if in the above example, we pass in `d, 'jake', ['articles']` as input, it should return 1 since jake has only written one article. His other document is a tweet which we are not interested in at the moment.

Make sure your changes in this fourth task do not break the tests for the previous task.