hw02

July 18, 2023

1 Metadata

```
Course:   DS 5100
Term:     Summer 2023 Residential
Module:   M02 Homework
Author:   R.C. Alvarado
Date:     7 July 2023
```

2 Student Info

- Name: Lindley Slipetz
- Net ID: ddj6tu
- URL of this file in GitHub: https://github.com/sliplr19/DS5100-ddj6tu/blob/main/lessons/M02/hw02.ipynb

3 Instructions

In your **private course repo on Rivanna**, write a Jupyter notebook running Python that performs the numbered tasks below. For each task, create a code block to perform the task.

Save your notebook in the `M02` directory as `hw02.ipynb`.

Add and commit these files to your repo.

Then push your commits to your repo on GitHib.

Be sure to fill out the **Student Info** block above.

To submit your homework, save the notebook as a PDF and upload it to GradeScope, following the instructions.

4 Data

```
Table 1: GRADES

name    grade
Jon     95
Mike    84
Jaime   99
```

Table 2: TOUCHDOWNS

name	touchdowns
Alex	2
Patrick	4
Tom	1
Joe	3
Alex	1

5 Tasks

5.1 Task 1

Using the data in Table 1, create a dictionary called `gradebook` where the keys contain the names and the values are the associated grades. Print the dictionary. (1 PT)

[2]:
```
gradebook = {
'Jon':     95,
'Mike':   84,
'Jaime':   99
}
```

5.2 Task 2

Index into the gradebook to print Mike's grade. Do NOT use the `get()` method for this. (1 PT)

[4]:
```
gradebook['Mike']
```

[4]:
```
84
```

5.3 Task 3

Attempt to index into gradebook to print Jeff's grade. Show the result. Do NOT use the `get()` method for this. (1 PT)

[5]:
```
gradebook['Jeff']
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Cell In[5], line 1
----> 1 gradebook['Jeff']

KeyError: 'Jeff'
```

5.4 Task 4

Using Table 2, build a list from the names called `names` and print it. (1 PT)

[15]:
```
names = ["Alex", "Patrick", "Tom", "Joe", "Alex"]
```

[16]:
```
print(names)
```

```
['Alex', 'Patrick', 'Tom', 'Joe', 'Alex']
```

5.5 Task 5

Sort the list in ascending order and print it. (1 PT)

[21]:
```
print(sorted(names))
```

```
['Alex', 'Alex', 'Joe', 'Patrick', 'Tom']
```

5.6 Task 6

Build a set from the names in Table 2 and print it. (1 PT)

[23]:
```
nameset = set(["Alex", "Patrick", "Tom", "Joe", "Alex"])
print(nameset)
```

```
{'Patrick', 'Alex', 'Tom', 'Joe'}
```

5.7 Task 7

Build a dictionary from the touchdowns data, calling it `td`, and print it. Use lists to store the values. Remember that dictionary keys must be unique. (1 PT)

[26]:
```
td = {"Alex1":    2,
"Patrick":  4,
"Tom":      1,
"Joe":      3,
"Alex2":    1}
tdvalues = list(td.values())
```

[26]:
```
[2, 4, 1, 3, 1]
```

5.8 Task 8

Compute the sum of Alex's touchdowns using the appropriate built-in function.

[27]:
```
Alexsum = td["Alex1"] + td["Alex2"]
```

[27]:
```
3
```

5.9 Task 9

Get the keys from `td` and save them as a sorted list `list1`. Then get a set from `names` and save them as a sorted list called `list2`. Compare them with a boolean operator to see if they are equal. (2 PTS)

[28]:
```
list1 = sorted(list(td.keys()))
list2 = sorted(list(nameset))
list1 == list2
```

[28]:
```
False
```

[ ]:
```

```