Python Lab 4.2: Use of Dictionaries

> The purpose of this practice is to help you apply the concepts discussed up to **now**:
>
> - add keys and values to dictionaries
> - increment values based on keys

In `lab4_2.py` in the text editor at top-right, write a program which will:

Write a program to read through the `2021-07-08_clean-hashtags.tsv` and figure out the tag containing keywords around the coronavirus such as 'virus','coronavirus', 'vaccine', 'covid' with the most mentions.

- The program looks for lines containing the #tag that includes the word 'virus' or 'coronavirus' or 'vaccine' or 'covid' and takes the second word of those lines as total mentions.
- The program creates a Python dictionary that maps each tag we are interested at with the number of its mentions.
- After the dictionary is produced, the program reads through the dictionary using a maximum loop to find the tag with the most mentions.

{% next %}

Your output should look like the following:

```
#covid19 34468
```

▶ Hint 1 : Read the file and add the data in the dictionary

```
fhand = open('2021-07-08_clean-hashtags.tsv', 'r')
virustags = {}

for line in fhand:
    if 'virus' not in line and 'vaccine' not in line and 'coronavirus' not in line
and 'covid' not in line:
         continue
    tag = line.split()[0]
    tagmentions = line.split()[1]
    virustags[tag] = virustags.get(tag, 0) + int(tagmentions)
```

A different way to search in the line

In the solution above we used multiple conditions to check if a line contains either of these words:

```
if 'virus' not in line and 'vaccine' not in line and 'coronavirus' not in line and
'covid' not in line:
         continue
```

Another way to perform this task is to declare a list with all the words you are interested in ... and in the loop check if none of the words is not found in the line, using the any keyword...

```
tags_to_search = ['virus', 'vaccine', 'corona', 'covid']
for line in fhand:
    if not any(tag in line for tag in tags_to_search):
        continue
```

▶ Hint 2 : Use a maximum type of loop

```
# initialize the most mentioned tag to None
max_tag = None

# initialize the most mentions to Zero
max_mentions = 0

# loop through every tag in the dictionary
for tag in virustags:
    # if the value (mentions) in the currect tag is greater that what was the
previous max_mentions
    if virustags[tag] > max_mentions:
        # keep this tag name to the max_tag variable
        max_tag = tag
        # set the max_mentions to this number
        max_mentions = virustags[tag]

# When the loop exits print the tag that you found!
print(max_tag, max_mentions)
```

Execute your program

Remember in order to execute your code you type in the terminal:

```
python lab4_2.py
```

Check that your code produces correct results.

For the sample datafile the output should be:

#covid19 34468

{% next %}

Check Your Code

Execute the below to evaluate the correctness of your code using `check50`, but be sure to test it yourself also.

```
check50 mkotsovoulou/ods6001a/main/labs/lab4_2
```

Execute the below to evaluate the style of your code using `style50`.

```
style50 lab4_2.py
```

{% next %}

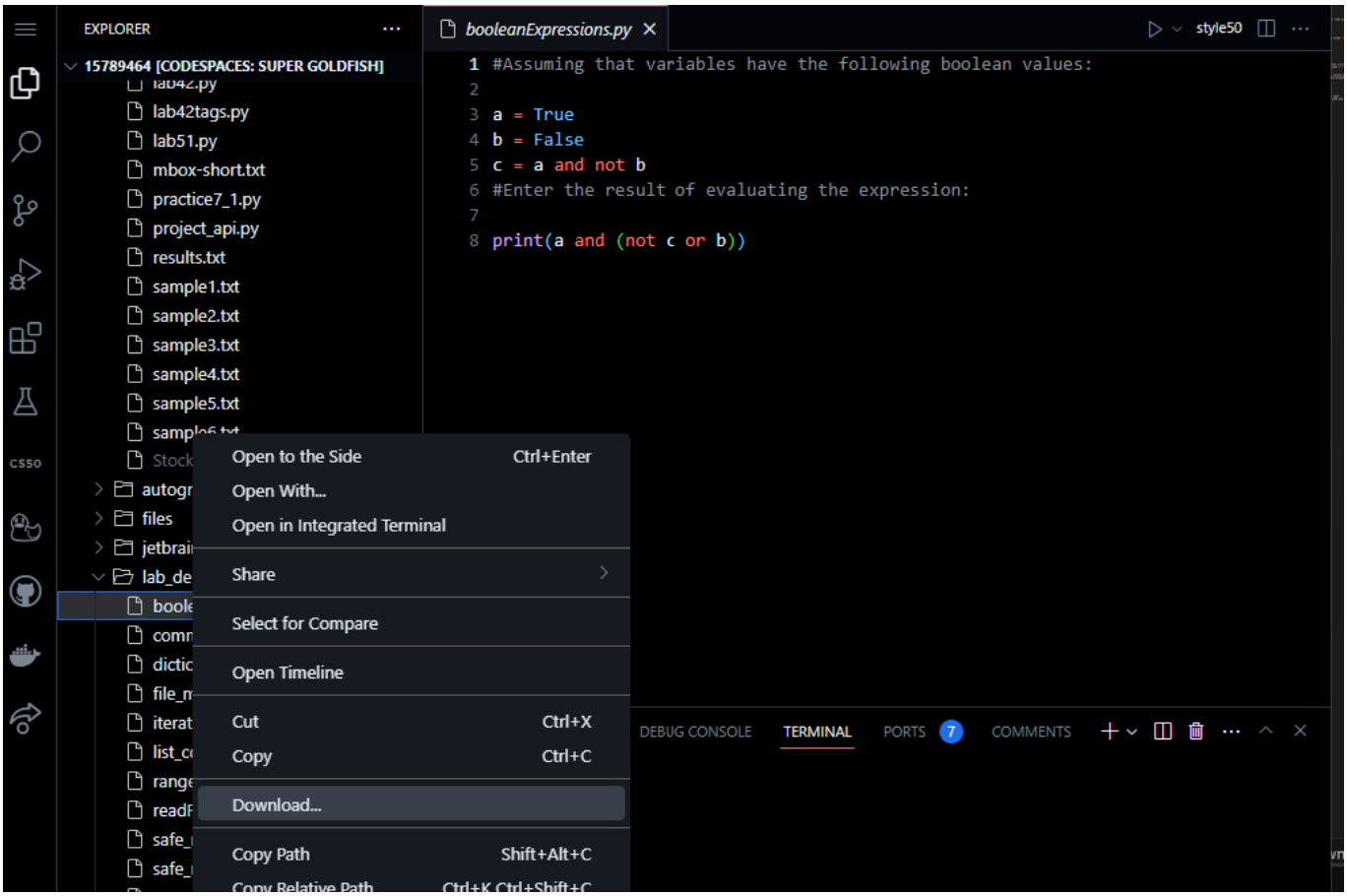
Submit your code

Execute the command below, logging in with your `GitHub username` and `Personal Access Token` when prompted. For security, you'll see asterisks (`*`) instead of the actual characters in your token.

If you do not have generated a Personal Access ToKen follow the instructions: https://docs.github.com/en/authentication/keeping-your-account-and-data-secure/creating-a-personal-access-token

```
submit50 mkotsovoulou/ods6001a/main/labs/lab4_2
```

You can re-submit your solution as many times as you want. When you are happy with your solution, download the code and upload it to Canvas.


EXPLORER
booleanExpressions.py
style50
15789464 [CODESPACES: SUPER GOLDFISH]
lab42tags.py
lab51.py
mbox-short.txt
practice7_1.py
project_api.py
results.txt
sample1.txt
sample2.txt
sample3.txt
sample4.txt
sample5.txt
files
jetbrai
lab_de
comn
dictic
file_n
iterat
list_c
range
readF
safe_
safe_
1 #Assuming that variables have the following boolean values:
2
3 a = True
4 b = False
5 c = a and not b
6 #Enter the result of evaluating the expression:
7
8 print(a and (not c or b))
Open to the Side Ctrl+Enter
Open With...
Open in Integrated Terminal
Share
Select for Compare
Open Timeline
Cut Ctrl+X
Copy Ctrl+C
Download...
Copy Path Shift+Alt+C
DEBUG CONSOLE
TERMINAL
PORTS 7
COMMENTS

Done!