Python Lab 3.2: Reading and Extracting Information from Files

> The purpose of this practice is to help you apply the concepts discussed up to **now**:
>
> - open a file
> - read the contents of the file
> - search inside strings for data
> - parse data

In `lab3_2.py` in the text editor at top-right, write a program which will: Search in a file named `datafile.txt` for all values in the `xmax` attribute, add them and display the sum. Open the datafile to observe the structure of the data...

The basic algorithm of the steps you have to perform is as follows:

1. Open the "datafile.txt"(included in your workspace)
2. Read the file contents for the file into a string variable
3. Split the file into 'lines' for processing
4. Loop through each line to search for `xmax`
    - if the line contains `xmax`
        - extract the number
        - add the number to the sum
5. Print the sum

{% next %}

To open the file and read the contents into a string variable use the open(filename) function. Remember that the open function returns a file handler... which you use to read()

▶ Hint 1 : Open and Read

```
fileh = open("datafile.txt")
data = fileh.read()
```

Of course you can also use readlines() instead of read() which will read and split at the same time... but in this example, we will split the string in the next step...

use the method splitlines() on the data and assign it to a lines collection

▶ Hint 2 : Split data into lines

```
lines = data.splitlines()
```

The next step is to write a loop to find the lines with `xmax` and print them

▶ Hint 3 : Loop, Search and print

```
 for line in lines:
    if "xmax" in line:
        print(line)
```

At this point you can run your code and see the output...

Your output should look like the following:

```
        "xmax": 451,
        "xmax": 732,
        "xmax": 984,
        "xmax": 399,
        ...
```

Now you need to think of a way to slice this string from the position after the colon : and before the comma , Think how you can find the position of the colon, and store it in a variable. then think how you can find the position of the comma, and store it in a variable. ☐ method to extract the number, but strip also the white spaces. Before adding the extracted number to the sum, convert it to an integer.

▶ Solution

```
fileh = open("datafile.txt")
data = fileh.read()
print(type(data))
lines = data.splitlines()
sum=0
for line in lines:
    if "xmax" in line:
        colon = line.find(':')
        comma = line.find(',')
        number = int(line[colon+1:comma].strip())
        sum += number
print(sum)
```

Execute your program

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

```
python lab3_2.py
```

Check that your code produces correct results.

For the sample datafile provides the sum of all xmax should be 13487.

{% 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/lab3_2
```

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

```
style50 lab3_2.py
```

{% next %}

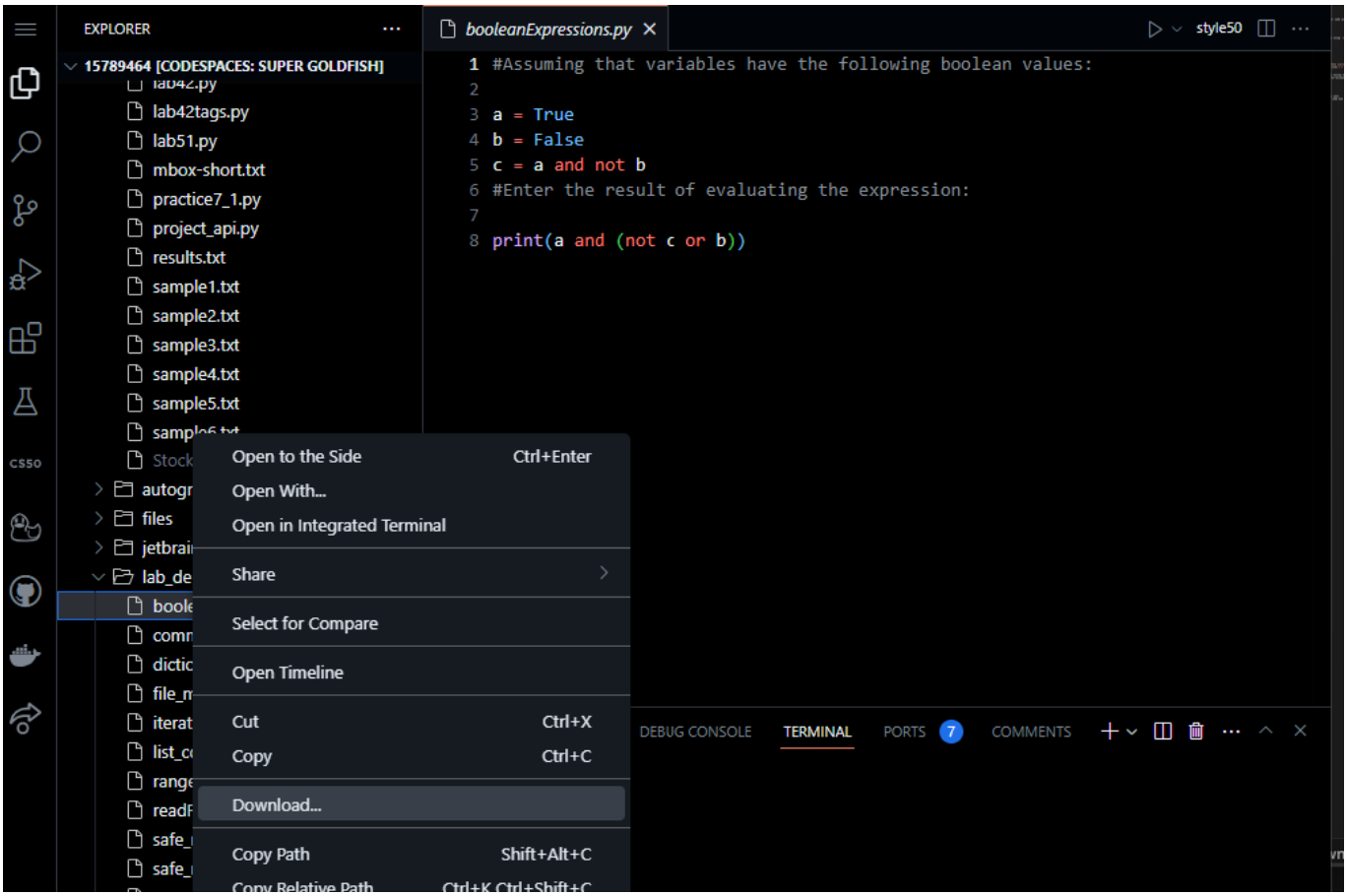
Submit your code

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/lab3_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!