Python Lab 3.1: String manipulation - Functions - Loops

> The purpose of this practice is to help you apply the concepts discussed up to **week 3**:
>
> - obtain user input
> - define functions that accept parameters and return values
> - call functions
> - perform string manipulation / iteration
> - use conditional expressions

In `lab3_1.py` in the text editor at top-right, write a program which will:

1. Define the avg_digits() function:
    - The function should accept one string parameter, and return the `average` of the digits in the string.
    - sum and countdigits are initialized to 0
    - The main part of the function should repeat the steps for each character in the string:
        - if the character is a digit:
            - add the number to the sum
            - increment the countdigits variable to keep track of how many digits exist
2. The main program should:
    - Read a sentence or a string from the user
    - Call the avg_digits() function by passing `str1` as an argument. Assign the value returned from the function to an `average` variable.
    - Display the `average` using only 2 decimal points.

Some Technical Details: A Basic Python main()

In some Python scripts, you may see a function definition and a conditional statement that looks like the example below:

```
def main():
    print("Hello World!")

if __name__ == "__main__":
    main()
```

In this code, there is a function called main() that prints the phrase Hello World! when the Python interpreter executes it. There is also a conditional (or if) statement that checks the value of **name** and compares it to the string "**main**". When the if statement evaluates to True, the Python interpreter executes main().

In this program write your code to accept user input in def main():

{% next %}

You can use a for loop

▶ Hint 1 : Iterate through every character in a string

```
for char in input_str:
    ....
    ....
```

Use char.isgigit() function. This function return true of the character is a digit and false other wise...

▶ Hint 2 : Check if the character is a digit

```
def avg_digits(input_str):
    sum = 0
    countdigits=0
    # repeat the steps below for each character in the string
    for char in input_str:
        # is the character a digit?
        if char.isdigit():
             # increment the countdigits
             # add the number to the sum

    #return the average (sum/countdigita)
```

Use print(f" ") to format your output

▶ Hint 3 : how to print only two decimals

```
 print(f"{average:.2f}")
```

Execute your program

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

```
python lab3_1.py
```

Check that your code produces correct results.

For the sample string provided, the average is: 6.333333333333333 and when formatted using 2 decimals, the correct output should be 6.33

{% 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_1
```

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

```
style50 lab3_1.py
```

{% next %}

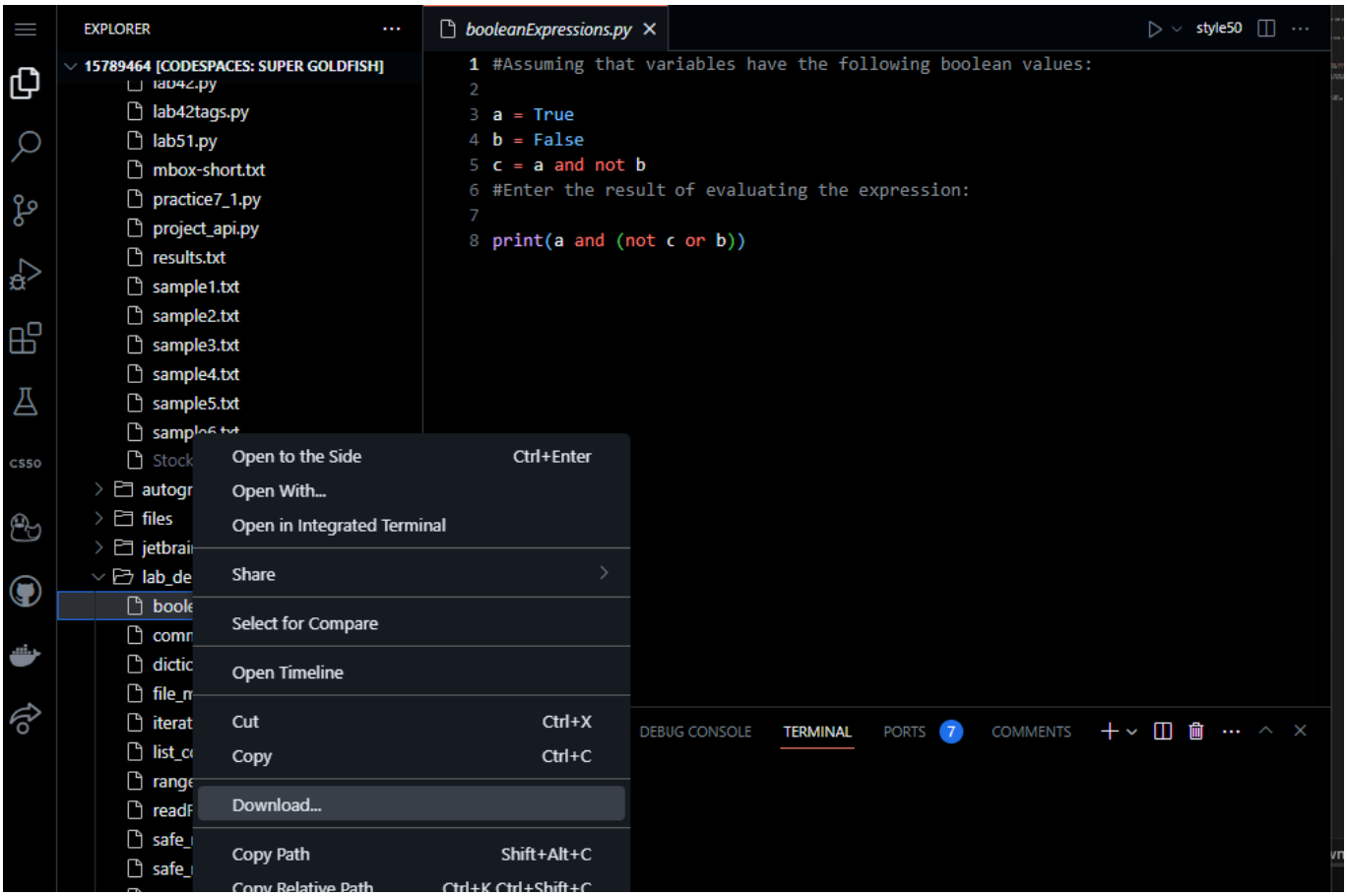
Submit your code

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_1
```

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!