

Python Lab 2.1: Conditionals

In `lab2_1.py` in the text editor at top-right, write a few python commands to:

1. Ask the user to type their score in a test and store it in a variable
2. Test if the score entered by the user falls within the valid range from 0 to 100. If not display `Invalid score`.
3. If the user enters character/s instead of a number display `Wrong input`.
4. Using the following table to find and display the letter grade of the user score.

Low Range	High Range	Letter grade
0	59	F
60	69	D
70	79	C
80	89	B
90	100	A

► Spoiler warning

Spoiler text. Note that it's important to have a space after the summary tag. You should be able to write any markdown you want inside the `<details>` tag... just make sure you close `<details>` afterward.

```
console.log("I'm a code block!");
```

```
{% spoiler "Hint 1 : Validate range " %}
```

```
if score < 0 or score > 100 :  
    print('Invalid score')
```

```
{% endspoiler %}
```

```
{% spoiler "Hint 2 : Invalid Input " %}
```

use a try/except block after reading the score which will include the conversion to a floating point number... and exit the program if the input is invalid.

```
try:  
    score = float(score)  
except:
```

```
print('Wrong input')
quit()
```

{% endspoiler %}

{% spoiler "Hint 3 : Finding the letter grade " %}

```
if score < 60 :
    print('F')
elif score < 70 :
    print('D')
elif score < 80 :
    print('C')
elif score < 90 :
    print('B')
else:
    print('A')
```

{% endspoiler %}

{% next %}

Keep in mind that the sample solution is not the only solution. There are different ways to reach to the same result...

{% spoiler "Solution : Putting it all together..." %}

```
try:
    score = float(score)
except:
    print('Wrong input')
    quit()

if score < 0 or score > 100:
    print('Invalid score')
elif score < 60:
    print('F')
elif score < 70:
    print('D')
elif score < 80:
    print('C')
elif score < 90:
    print('B')
else:
    print('A')
```

{% endspoiler %}

{% next %}

Execute your program

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

```
python lab2_1.py
```

Use the following test data to make sure your program produces correct results.

TEST 1:

Enter your score: 35

F

TEST 2:

Enter your score: 102

Invalid score

TEST 3:

Enter your score: 60

D

{% next %}

Check Your Code

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

- ☒ a number out of the valid range
- ☒ a character or word as input
- ☒ a number below 60
- ☒ a number in every possible range 🎉

```
check50 mkotsovoulou/ods6001a/main/labs/lab2_1
```

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

```
style50 lab2_1.py
```

{% next %}

Submit your code

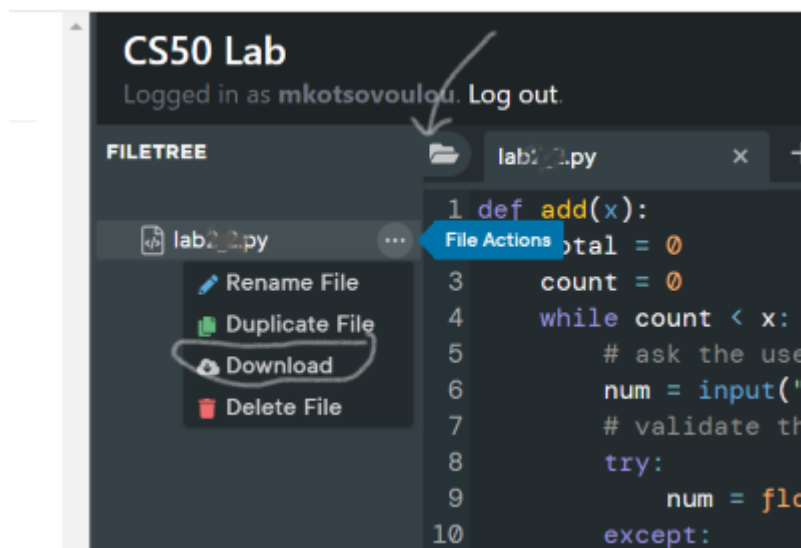
Execute the command below, logging in with your **GitHub username** and **Personal Access Token** when prompted. For security, you'll see asterisks (*) instead of the actual characters in your token.

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 mkotsoyoulou/ods6001a/main/labs/lab2_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.



Done!

