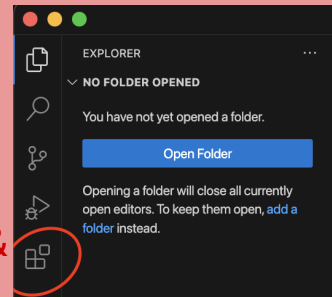


Python Grade Calculator

Output letter grades from percentages using Python!

PART 1: Set up Python & Visual Studio Code

1. This project will use a coding language called Python! You can download Python at <https://www.python.org/downloads/> and follow the installation steps once downloaded.
2. Visual Studio Code (VSC) is a code editor which we will use (if you want, you can use your own code editor). You can install VSC at <https://code.visualstudio.com/download> and follow the installation procedure.
3. Open Visual Studio Code. In the upper left corner, there should be 5 symbols, click the bottom symbol which looks like 4 squares (“extensions”).
4. Once you click on the extension tab, search up Python & install the 1st extension called “Python” published by Microsoft
5. Now you are ready to start creating your project! At the top of the page, click FILE > NEW FILE, it should prompt you to type in a name - type in `gradecalculator.py`
 - a. NOTE: You can replace “gradecalculator” with whatever filename you want, just make sure to add “.py” at the end.



PART 2: Start coding the basic Python

1. Create an empty array called grades - later, we will add all the percentages in it. In Python, an array is a list of strings, numbers, or other datatypes. Arrays are created by using brackets [].

```
grades = []
```

2. Under this line of code, create a variable which takes user input and asks the user how many grades they want to enter using input(). Make sure to convert this user input into integer format using int().

```
num_grades = int(input("How many grades do you want to enter? "))
```

3. Write a for loop which repeats for every grade that the user enters, and adds the grade to the array. A for loop repeats for every number/value in a specified range or amount. In this case, our for loop will repeat for every number within the range which the user inputted.

```
for num in range(num_grades):  
    grade = float(input("Enter 1 grade: "))  
    grades.append(grade)
```

4. Great! Now, above all of this code, we will define a function to calculate the grade. This function will take the grades as a parameter.

```
def calculate_grade(grades):
```

5. Inside the calculate_grade function, create a variable called average which is the sum of the grades / the amount of grades.

```
    average = sum(grades) / len(grades)
```

6. Below this line (still inside the function), we are going to write a series of if/else/elif statements for the program to calculate the letter grade. We want any average over 95 to be an A+, 90-95 to

be an A, 85-90 to be a B+, 80-85 to be a B, 75-80 is C+, 70-75 is C, 65-70 is D+, and anything below 65 to be a F. (you can change these numbers based on your grading standards)

```
if average >= 95:
    letter_grade = 'A+'
elif average >= 90:
    letter_grade = 'A'
elif average >= 85:
    letter_grade = 'B+'
elif average >= 80:
    letter_grade = 'B'
elif average >= 75:
    letter_grade = 'C+'
elif average >= 70:
    letter_grade = 'C'
elif average >= 65:
    letter_grade = 'D+'
elif average <= 65:
    letter_grade = 'F'
```

7. We want this function to output the average and letter grade. We can do this using **return**. This will return a tuple containing the average and letter_grade. A tuple in Python is used to store multiple items in one variable.

```
return average, letter_grade
```

8. Now go to the bottom of your program, after the previously written for loop. We will call the calculate_grade function taking the array grades as an argument. Then, we will assign the returned tuple to two different variables - average and letter_grade.

```
average, letter_grade = calculate_grade(grades)
```

9. Lastly, we want to output our code to the terminal. We can do this using **print** and string concatenation.

```
print("The average grade is: " + str(average))
print("The letter grade is: " + str(letter_grade))
```

10. Try running your code! You can do this by navigating to the correct folder in terminal using “cd” command, and then running the command “python3 filename.py”. The output should be functional now!

PART 3: FORMATTING!

1. Our next steps will be how to format the output to look nicer! Firstly, we will add an if statement in the case where the user doesn't provide grades as input. This will be inside the `calculate_grade` function at the top.

```
if not grades:  
    return None, "No grades provided"
```

2. Scroll to the bottom of your program (near the existing **print** commands). Above these, we will create another **print** command, acting as a header. We will format it using an f string, and centering this header within a 30 character wide field.

```
print(f"\n{'Grade Report':^30}")
```

3. Next we'll create a dashed line between the header and other **print** outputs. This can be done using the following line of code:

```
print(f"{'-' * 30}")
```

4. We will delete the 2 existing print commands that output the average and letter grade. Instead, we will replace them with formatted prints. The following lines of code will right-align the text and numbers with 20 spaces reserved for each value.

```
print(f"Number of Grades: {num_grades:>20}")  
print(f"Average Grade: {average:>20.2f}")  
print(f"Letter Grade: {letter_grade:>20}")
```

5. Try running the code again! This time, your code output will look a lot nicer and neater.