

Submitting Short Problems to Gradescope

Step 1: Downloading the Shorts from the class website

1. Under the “Assignments” tab from your class website, look under the “Additional resources” column for the short problem template zip file.

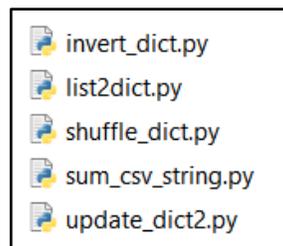
Additional resources

Test cases: | [assg01-long.zip](#)

Test cases: | [assg02-long.zip](#) Shorts files: | [shorts2_template.zip](#)

Test cases: | [assg03-long.zip](#) Shorts files: | [shorts3_template.zip](#)

2. Download this zip file and extract the contents. Inside the folder, you should see various python files—specifically one file for each short problem, with the same name as the short problem.



3. Open up VS Code or another IDE, and then add the folder with the short templates to your workspace. In VS Code, you would hit “File” > “Add Folder to Workspace” and then locate the folder you have downloaded.
4. Then, following the short problem instructions on the class website, you would fill out the rest of the starter code.

```
1  def sum_csv_string(csv_string):
2      # your code here
3
```

Step 2: Testing and Submitting on Gradescope

1. Locate the assignments in Gradescope labeled “PA-##-Shorts Autograder”, and click on the one for the assignment you are trying to submit to.

(BETA TEST): PA-03-Shorts Autograder	
	FEB 10, 2025 9:59 AM FEB 14, 2025 9:59 AM
(BETA TEST): PA-02-Shorts Autograder	
	FEB 10, 2025 9:59 AM FEB 14, 2025 9:59 AM

2. Submit the python files you have modified to run the autograder. If you are just trying to test specific short problems, you do not have to submit all of the short problem files for every submission—just the ones that you want to test. Make sure that your final submission **does** include all of the files though.

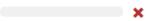
Submit Programming Assignment

Upload all files for your submission

Submission Method

Upload GitHub Bitbucket

Add files via Drag & Drop or [Browse Files](#).

Name	Size	Progress	x
invert_dict.py	55 b		x
list2dict.py	46 b		x
shuffle_dict.py	51 b		x
sum_csv_string.py	55 b		x
update_dict2.py	67 b		x

3. Once submitted, the autograder will run and automatically display which test cases you are passing or failing. If you are failing a test case, a message will be displayed showing you the inputs, expected outputs, and your output. If you are using something restricted for the short problem, the test case will fail and you will be told which content is banned. If your code results in a runtime error, the error message will display instead.

1) Sum-CSV-String-Test1 (0/0.8333333333333334)
Test Failed: 572 != 0 : ----- Input: '572' Expected output: 572 ----- Your output: 0
2) Sum-CSV-String-Test2 (0.8333333333333334/0.8333333333333334)

If you have any comments, concerns, or general feedback about the autograder, feel free to let us know!