Lab 5: Code Reviews

2024-10-04

Table of contents

Hands-on steps for the lab

1. Pair up in teams of 2 (or 3 if necessary).
2. One member of the team will create a new repository on https://github.berkeley.edu/ and invite the other team member(s) as collaborator(s) on the repo (Settings, Collaborators, Manage access, Add people).
3. Using the GitHub web interface, each team member will create a new file in the repo via "Create new file" under "Add file", and paste their code for presidential debates into it (problem 2 of PS3). Note that the files should have different names or be placed under different directories to avoid conflicts.
4. Commit your new file (from the GitHub web interface) as a pull request / separate branch. That is, select "Create a new branch for this commit and start a pull request" instead of "Commit directly to the main branch". Discuss the differences of committing to the main branch and opening a pull request with your partner(s) and when you would use one or the other.
5. Write a descriptive message (what is the file being added, why are you adding it, are there any details the reviewer should be aware of, etc.) to accompany your pull request.
6. Now each team member will go to the pull request made by another team member and look at the code changes.
7. Read each other's code carefully and leave some thoughtful comments.
8. The comments may be about implementation details, efficiency concerns, or style and readability.
9. Remember to be constructive and kind :)

Considerations

Each review is going to be unique, but there are a few common points that might be worth considering in general. We suggest students think about (and beyond) the following:

- How is the code organized into functions or class methods;
- Use of loops vs. map() vs vs. list comprehensions;

- What approach (if any) was taken to identify unexpected things in the text and correct them, so that the parsing into chunks and sentences could be improved;
- What checks/assertions (if any) were implemented as sanity checks of the results (not unit tests of functions, but rather checks at steps of the workflow to see that the results are as expected at given checkpoints, e.g., that the number of chunks is not extremely small or large);
- What other checks/assertions do you think could be considered.

Acknowledgements

This lab was originally authored by Ahmed Eldeeb and adapted for the Fall 2024 semester.