

Session 7A Version Control using Git and Working with GitHub

Exercise 2 – Git Branches and Merging

For this exercise, one person should share their screen and walk everyone through the exercise, so that everyone in the group follows along on their own machine. Make sure everyone is ok with all steps and take things as slowly as you need. Refer back to the lecture slides to help you.

1. Have a look at the full git log for *main*.
2. Using the repository you set up earlier, create a new branch called *dev*, and ensuring you have switched to the new branch
3. Use a *git restore* to restore code from an earlier commit
4. Make some changes to this code. Then save it, and stage and commit the changes.
5. Have a look at the full git log for *dev*.
6. Merge the changes you made on *dev* into your *main* branch. Make sure you are in the correct branch first before doing this.
7. Have another look at the log to view the commit history again
8. Merge *main* back into *dev*. Again, make sure you are in the correct branch before doing this. Confirm that both *dev* and *main* have the same HEAD.
9. Make a change to some of the lines of code of one of your code files. Then save, stage and commit the changes.
10. Switch back to *main* and make some **different** changes to the **same lines of code in the same file**. Save, stage and commit the changes.
11. Try merging *dev* into *main*. You should be informed that there is a conflict.
12. Make a decision on how to resolve the conflict. Save, stage and commit the changes.
13. Look at the log to confirm the new commit.
14. In the folder of your repository, create a new .csv file called *input_data.csv*, as well as a folder called *raw_data*, which itself contains another new .csv file called *raw.csv*.
15. Check the status of your repository and confirm that the new files are acknowledged but not being tracked.
16. Create, stage and commit a .gitignore file that tells Git to ignore the *input_data.csv* file in your repository, as well as anything in the folder *raw_data*.
17. Check the status of your repository again and confirm that the new files are no longer being acknowledged (a “clean working tree”)
18. Create, stage and commit a fictitious *readme.md* file for your repository
19. Add a new function to the code in of your code files in your *dev* branch – save, stage and commit the changes. Then create a new (third) branch from *dev*. You should observe the code and commit histories are taken from *dev* and not from *main*.
20. Play around with making changes in your third branch, merging back from your third branch into *dev*, and then in turn into *main*.