Git and GitHub for Group Projects

In this practice, you will be working in groups of 2-5 to practice using Git and GitHub to collaborate on a Python project. You will be working on a simple Python script that calculates the average of a list of numbers. Each group member will work on a different function of the script, and then you will merge your changes together to create a final version of the script.

Note: Please go through steps 1 to 9 each time with a different team member as the leader. Because the team lead responsibility differs and the commands they use are unique to them, pass the lead title to another team member so that everyone becomes a leader at least once.

Step 1: Clone the repository

1. Go to the GitHub and create a repository for your group.
2. Clone the repository to your local machine using the git clone command.

Step 2: Divide the tasks

1. As a group, divide the tasks amongst yourselves. Each person should work on a different function of the script, as outlined below:
    - Function 1: Write a function that takes a list of numbers as input and returns the sum of the numbers.
    - Function 2: Write a function that takes a list of numbers as input and returns the length of the list.
    - Function 3: Write a function that calculates the average of a list of numbers using the functions from step 1 and 2.

Step 3: Create a branch for each function

1. Each group member should create a new branch for the function they will be working on using the git branch command.
2. Switch to your branch using the git checkout command.

Step 4: Write the code

1. Write the code for your function in the Python script provided in the repository.
2. Save the changes and commit them using the git commit command.

Step 5: Merge the branches

1. After all group members have completed their functions, team lead should merge the branches into the main branch using the git merge command.

Step 6: Push the changes

1. Push the changes to the remote repository using the git push command.
2. Check the GitHub repository to ensure that the changes have been successfully pushed.

Step 7: Review and finalize

1. Team lead should review the final version of the Python script with your group members and ensure that it is working correctly.

Step 8: let's create a confilct

> Today we will take a look at two cases of confilct. We first create the confilcts cause nothing can teach us more than our own mistakes, then we try and resolve them.

Case 1: Have two members of your team, let's say A and B, change the same function in the same line, then commit the changes. One of them (A) should now push the changes they made. Now tell the second person (B) to try to pull the changes from remote repository. Congrats! We created a confilct. (if you have trouble making a confilct, call your mentor)

> **Note: if you did the first case successfully go to Step 9: Resolve, before doing the second case.**

Case 2: Have A and B change the same line of code again but this time tell both of them to push their versions of code at the same time.

Step 9: Resolve

Now try resolving the confilct with hints git provides and/or searching. In second case, you can have your leader , merge the pull requests from their github. After resolving the second case in previous step continue reading the questions.

Try thinking about these questions before reading the answers:

1. When is the first scenario likely to happen?
2. What could have gone wrong that the second case happened in the project?
3. Whose responsibility is it to solve these confilcts?

Answers:

1. First case is likely to happen when we didn't pulled the latest changes from the remote repo.
2. Second case, however, is more likely to happen when the team did not delegate the duties correctly and two people edited the same file.
3. First one you should solve on your own, always check for updates → `git fetch`/`git pull` (take a look at `git stash` if you changed a lot and can't pull without saving). In the second scenario the one to clean the mess may differ. But a percise delegation of tasks can prevent this kind of confilcts in most cases.

Good luck, and have fun collaborating on this project using Git and GitHub!