

Before starting the final project, let's have a quick refresher on how to do [code collaboration with Git](#). Please find a partner to do this together else it will defeat the purpose of this whole exercise.

Choose one person to be Coder A and another to be Coder B. Then perform the following:

1. Coder A creates a Rails app and initializes a git repository. Call it any name you want. Do the necessary (`bundle install`, `bundle exec rake db:create`). Set up a `welcome#index` view and route as well.
2. Add, commit and push your app to Github.
3. Coder A adds Coder B as a collaborator (go to "Settings" and look for "Collaborators & Team").
4. Once Coder B accepts the invitation, Coder B will clone the repository (you don't have to fork it anymore since you are working on the same).
5. Both Coder A and Coder B now each creates a new branch. For example, Coder A can do `git checkout -b feature_a` and Coder B can do `git checkout -b feature_b`.
6. In each of your branch, do whatever you like. For example, Coder A can create a model called "articles" and do something there, whilst Coder B can create a model called "comments" and do something else there.
7. Once you guys are done, let's now take turns to merge each of your branches with master. Assuming that Coder A starts first. a. First, `git add` and `commit`. Do not push yet. b. Go to the `master` branch and do `git pull origin master`. c. Go back to your branch (`git checkout feature_a`). c. Push your branch into Github, i.e. `git push origin feature_a`.
8. Now go to Github. You will see that your branch has now appeared on Github. Click on "Compare & Pull Request" and follow the instructions there. Once you are done, congratulations, you have successfully merged your remote branch to the (remote) master branch!
9. Now go back to your local machine. Go to the `master` branch and do `git pull origin master` so that you have the updated repo in your local machine.
10. Coder B now does Steps 7 to 9.

You guys are done! Good job! Here are 2 more things for you to try out:

1. If you want to practice dealing with conflicts, both of you can also choose to work on the same file, in this case, maybe `welcome/index.html.erb`. Remember to each checkout a branch before working on the same file! It's always a BAD idea to work on the master branch.
2. You can also `git merge master` when wanting to merge your branch with the master branch. But note that this will only merge your branch with the local master branch, not the remote master branch. Go figure out with your partner on how to go about this :) If you made a mistake, it's okay. The more you practice, the more you get better at using git and will be more ready for your final project and future code collaboration work! :)