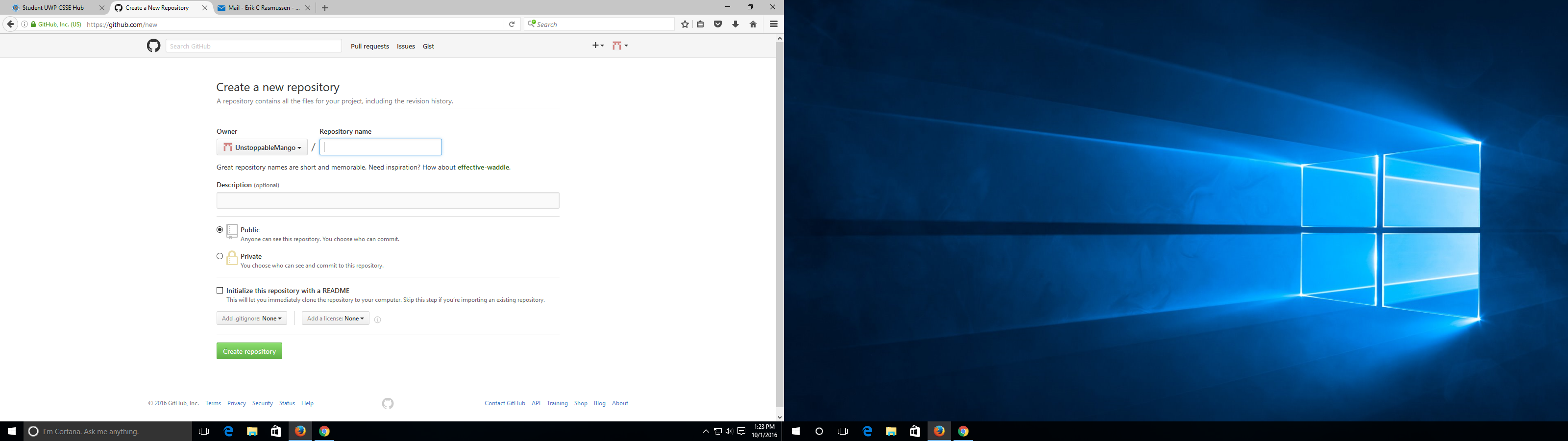
In order to create a new repository on github, first navigate to the webpage using a web browser. Then locate and click the button labeled “New repository”. In the next screen type the name of the repository in the field labeled “Repository name”. If desired, a description can be added to provide basic information about the repository. Once all the required fields have been filled, click the button labeled “Create repository”. A repository on github has now been created.

In order to create a new repository using the git CLI, first navigate to and open the git CLI. Next, type “git init” followed by any desired parameters. Typing “git init” alone will create a repository at the root folder location.

In order to clone the repository from github into a local repository, first navigate to github using a web browser. Sign in, and select the repository to be cloned. Locate the dropdown labeled “Clone or download” and select in. Within the dropdown, either select and copy the link directly, or click the button off to the right to copy the link to the clipboard. Open the git CLI and navigate to the desired repository location. Next, type “git clone “ and paste the link copied from github. The remote repository will now be cloned to the local directory.

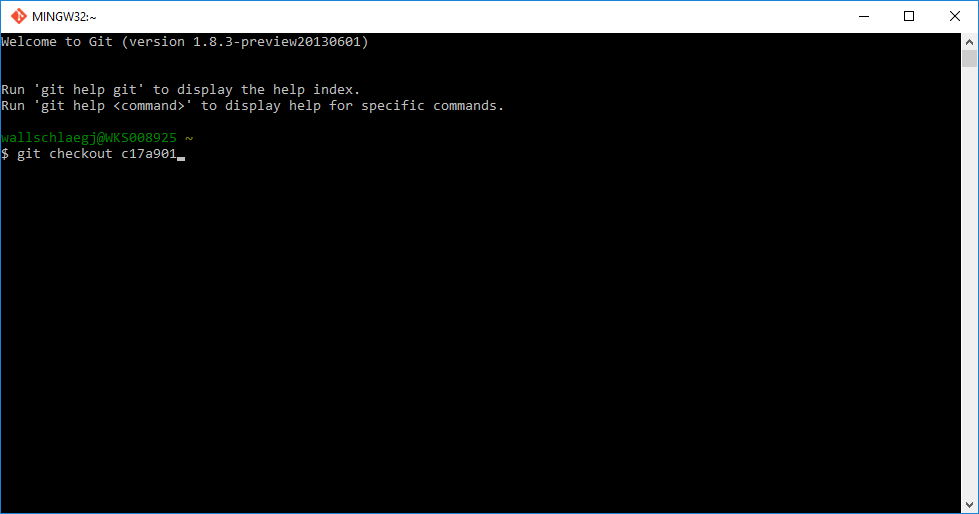


In order to create a new branch, you need to use the “git branch <name>” command and type the name of the new branch in <name>. Your new branch will be created. Then you need to use “git checkout <name>” in order to switch to that branch. You can now make modifications and edits to documents in the new branch.

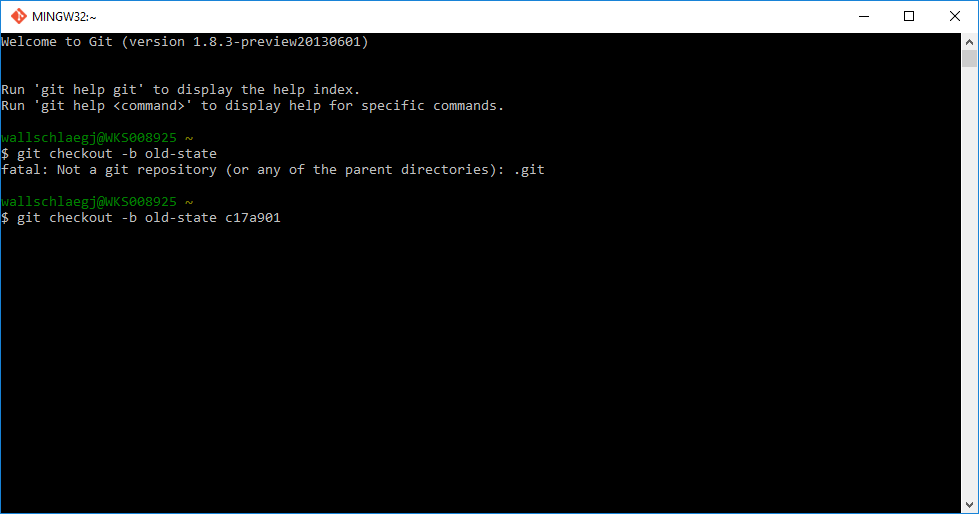
After you have finished your edits, you can merge the branch together with another one. First checkout to the desired base branch. Then you merge the branch with the base branch using “git merge <name>”. You then use “git push origin master” to push the changes to the remote repository.

With Git, reverting has a few different general meanings. You can either temporarily revert to a previous commit, permanently delete unpublished commits, or undo previous published commits with new commits. In all scenarios, the process is very easy and straight forward, and only involves a few commands once the git log has been accessed.

First, let’s say you want to temporarily revert to a previous commit. That is, temporarily go back to it, fool around, then come back to where you are, all you have to do is check out the desired commit using the following command:



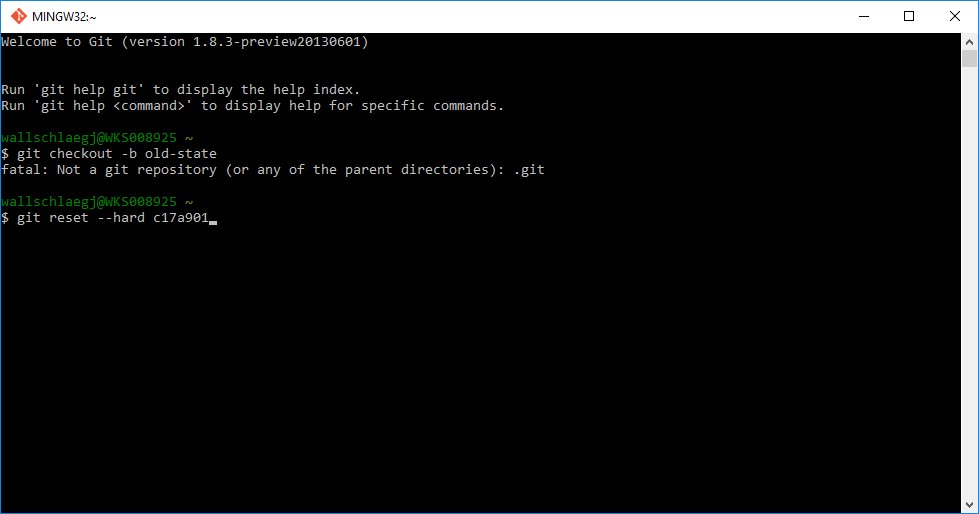
Or if you want to make commits while you're there, all that’s needed is to make a new branch at that commit.



To go back to where you were before reverting, just check out the branch you were on again.

If you want to get rid of everything you’ve done after the commit you’ve reverted to, there are the other two options; delete unpublished commits or overwrite previously published commits with new commits.

For unpublished commits, it is a simple case of using the reset command:



This will erase any uncommited work, so be careful when using this command.

On the other hand, if you've commited the work, you probably don't want to reset the branch, since that's effectively rewriting history. In that case, you could indeed revert the commits. With Git, revert has a very specific meaning: create a commit with the reverse patch to cancel it out. This way you don't rewrite any history.

