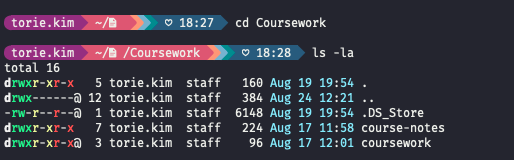
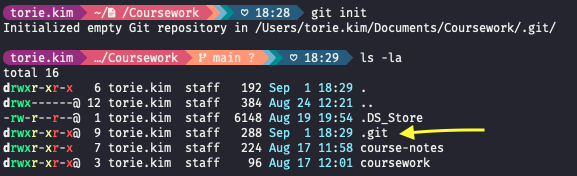
# How to make your Coursework directory into a git repository on Github

Why create a new repository for each workshop, when you can keep it all in one place? Follow the steps to make your local Coursework directory a git repository:

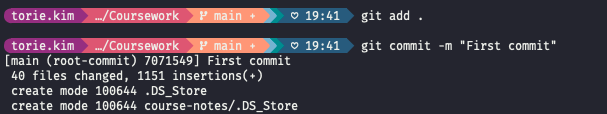
1. Check if your “Coursework” directory is a git project. First cd into your project directory. Then ls -la in CLI to see if **.git** is listed:



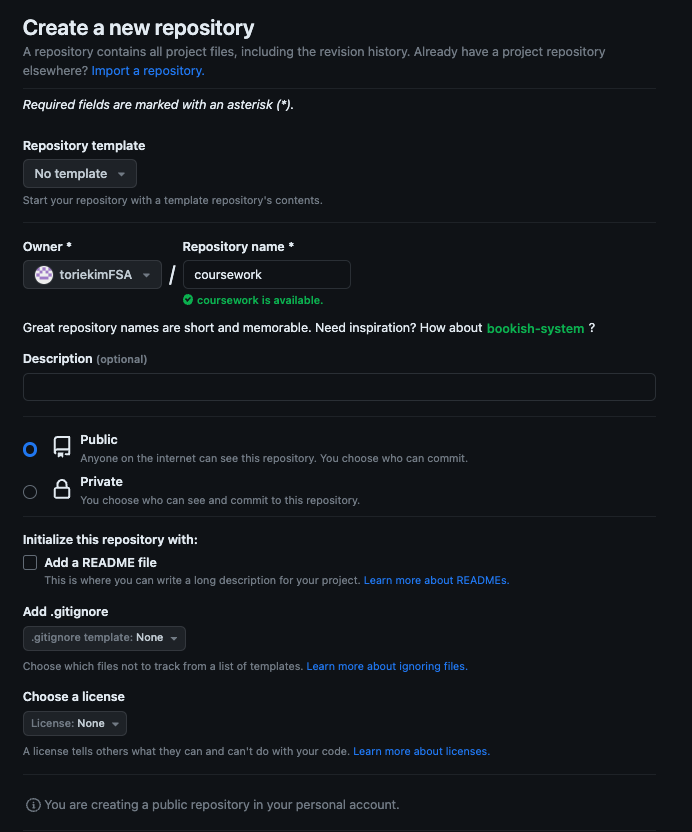
If it is, continue on to step 2. If not, git init to create a git project. Then ls -la to confirm that it has been created.



1. Add all files (git add .) and make your first commit (git commit -m <message>):



1. Create a new repository on GitHub called “coursework”



1. Follow steps under **Quick setup** > **…or push an existing repository from the command line** by copy/pasting the commands:

git remote add origin git@github.com:toriekimFSA/coursework.git

git branch -M main

git push -u origin main



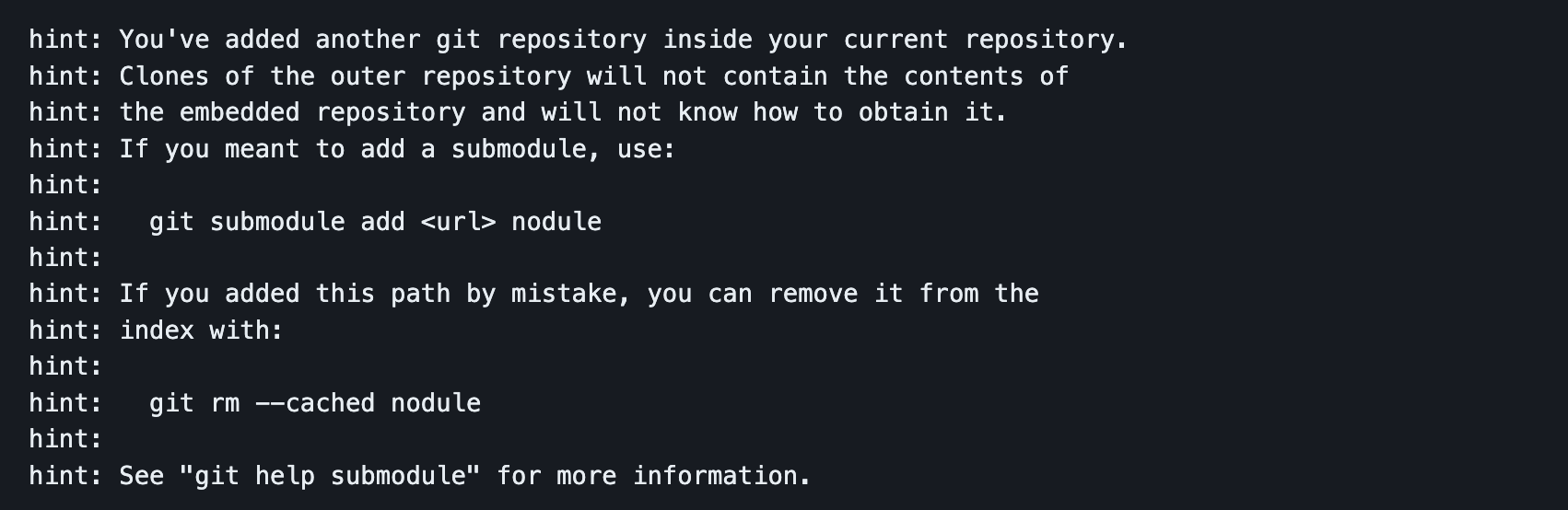
1. Refresh your page and you should now see your coursework repository!
2. Now, when you make changes or add to your project, you should add, commit, and push to your remote repository on Github!

Example workflow:

git add main.html   
git commit -m “add: block07 workshop html file”  
git push origin main

## Troubleshooting

If you’ve already created a git repository with one of your subdirectories in Coursework (i.e. block04 or github-workflow workshop is already a git repository within your Coursework directory), you may get a warning message like the following:



[This gist](https://gist.github.com/claraj/e5563befe6c2fb108ad0efb6de47f265) has a great overview of what’s happening and how to resolve the issue if you want just one git repo with everything in it. Essentially, you’ll have to delete the inner directory’s .git folder and then go to the parent directory and run the command git rm --cached <directory name>

If you want to keep the inner directory as a git repository, read and follow the directions in [this dev.to article](https://dev.to/jjokah/submodules-a-git-repo-inside-a-git-repo-36l9). The tldr; you can make it a submodule of the parent directory by running the command git submodule add <url> nodule, where **<url>** is the URL of the inner directory’s repo and **nodule** is the name given to the submodule. For example, for github-workflow it would look like:

git submodule add https://github.com/toriekimFSA/github-workflow github-workflow