GW ACM Git Cheat Sheet

Setting Up Git for the First Time

COMMAND	WHY?
git config --global user.name “my”github_username”	Set the name associated with your commits
git config --global user.email	Set the email associated with your commits (this should be the one you use for github)

Creating a Repository

COMMAND	WHY?
git init [project-name]	Initialize the folder to now be a git repository. If you want to create a NEW repo from scratch, use this command
git clone [url]	If you need to download a git repo (from somewhere like github) then use this command! Clones the repo from url to the current directory

Do Some Work!

COMMAND	WHY?
git add [file]	When you’ve made some changes to a file, add it to the stage for the next commit
git status	This tells you what changes you’ve made to the repo. Useful for figuring out what you’ve added to the stage
git reset [file]	If you want to take a file off the stage, without actually changing the contents, use this command.
git diff --staged	Shows the changes made to the files on the stage compared to the last commit
git commit -m “[commit message]”	This creates a commit (or snapshot) of the current status of all files on the stage. Commits are a history of all changes made to a project, so it is important to make a commit after every major change

GW ACM Git Cheat Sheet

Tell Git You Changed Files

COMMAND	WHY?
git rm [file]	Deletes the file from the directory, and tells github to delete the record of the file
git rm --cached [file]	Removes the file from git, but does NOT actually delete the file off your computer
git mv [file-original] [file-renamed]	Changes the filename or location, and tells git to track the move

Syncing With the Cloud

COMMAND	WHY?
git pull	"Pull" (download) any changes from github (or wherever you might have your repo online).
git push	"Push" changes to your remote source (e.g. github)

Branching

What is branching? It's for new features! Don't push everything to your master branch, only put code you're confident in master. Instead, have a branch for experimental features, then merge the branch with master when you're done!

COMMAND	WHY?
git branch	Lists all local branches in the current repo
git branch [branch-name]	Creates a new branch. As mentioned above, you should put a new major feature in a branch, so that you don't put broken/experimental code in your master branch
git checkout [branch-name]	Switch from the current branch to the specified branch. Will only work if there are no staged changes (so either run git commit or git stash)
git merge [branch]	Combines the specified branch's history into the current branch
git branch -d [branch]	Deletes the specified branch