Introductory Git Guide

Common Commands

Note: These command examples use [brackets] to denote placeholders. You will fill in the placeholders, and not use brackets in the final command.

Description	Command
Clone a repository from a remote	`git clone [url]`
Check the status of a git repository	`git status`
Stage a particular file for commit	`git add [filename]`
Stage all changes in the current directory for commit	`git add .`
Commit the staged changes (makes a timestamped commit/checkpoint)	`git commit -m "Your message here"`
Push local commits to a remote repository	`git push`
Push local commits to a remote repository, saving the specifics of where that code should end up	`git push -u [remote name] [remote branch (master)]`
Retrieve the latest changes from a remote (teams, multiple computers)	`git pull`
Create a new git repository in the current working directory	`git init`
Add a new remote	`git remote add [name] [url]`
Remove a remote	`git remote rm [name]`
View list of remotes and their urls	`git remote -v`
Set the name to include in all future commits (initial setup)	`git config --global user.name "Your Name Here"`
Set the email address to include in all future commits (initial setup)	`git config --global user.email "yourEmailHere@something.com"`

Starting a Project

Technique 1

You can use this technique when you are starting a project from scratch.

1. Create a new repository on github.com. If you don't have a README file already, you can check the box to have one created for you.
2. Copy the clone (https) url from the repository page on github.
3. In your terminal, make sure your working directory is your Source folder (or whichever folder contains all the projects you will work on in this class)
    a. DO NOT create a project folder yourself. The cloning process creates a folder.
4. Use the clone command from the table above, making sure to paste the link copied in step 2.
5. The repository should clone, and you should now have a new folder inside your Source folder named after the repository on github.com. You will need to change your working directory to this folder before running any git commands on it.

Technique 2

You can use this technique always, but it is especially useful when you already have a project folder on your computer you would like to send up to github.

1. Create a new repository on github.com, making sure that NO boxes are checked for creating a README, .gitignore, etc.
2. Copy the clone (https) url from the repository page on github.
3. In your terminal, make sure your working directory is your project folder
4. Use git init to initialize git inside the project folder, so that you can use git commands
5. Use the `git remote add` command from the table above to add a new remote named `origin`. For the remote url, you should paste the https url copied in step 2.
6. At this point, all the git commands should work. When it comes time to push for the first time, you will need to use: `git push -u origin master` to properly set the destination for pushed code. After doing this once, you will only need to use `git push` (without the -u part) in the future.

Working on a Project

- More commits are better than fewer commits
- Once you have made a logical change, you should commit the files involved in that change and use a helpful message as the commit message
    - For example, you may have a staff page. Adding a new staff member may involve adding a new image file to your project, adding some html, and adding some css. At a minimum, that change encompasses 3 files. You might consider staging those 3 files and making a single commit with the message "Added Jane Doe to the staff page"
    - If you're just getting started, you can commit just a basic unfinished file. It doesn't have to be perfect!
- When you are ready, you can push commits to the remote. This is important because it lets us see your progress, and keeps your code safe online in the event something were to happen to your computer
    - You don't have to push after every commit. Some people push before taking a break, or if they need to show someone their code.

Additional Topics

If you are enrolled in a course that includes a more advanced git module, you will eventually learn these concepts:

- branches
- merging
- merge conflicts