

1. Create github account if you do not have one
2. Go to <https://github.com/CSC322-Grinnell/Treefluent>
3. If you do not have git, install git using these instructions:  
<https://www.linuxbabe.com/linux-server/install-git-version-control-on-linux-debianubuntu-fedora-archlinux-centos-2>
4. Before working, if you do not have a local branch, create one
  - a. To create a clone, first use cd to navigate to the folder you want to put the clone in
  - b. Then do git clone ssh:<ssh>
    - i. Find the ssh on git hub by clicking on code, then ssh then copying it
5. When you plan to work on the code first using your git terminal use cd to navigate to the folder that was created when you created a clone
6. Then do git pull
7. Make changes
8. Then do git add <name>
  - a. Name is all the files you changed. Make sure to do these steps at the end of every work session
9. Once you've added every file do git commit -m 'MESSAGE'
  - a. Use the message template below
10. Do git push
11. DONE!!! :)

### Commit messages:

Capitalized, short (50 chars or less) summary

More detailed explanatory text, *if* necessary. Wrap it to about 72 characters or so. In some contexts, the first line is treated as the subject of an email and the rest of the text as the body. The blank line separating the summary from the body is critical (unless you omit the body entirely); tools like rebase can get confused *if* you run the two together.

Write your commit message *in* the imperative: "Fix bug" and not "Fixed bug"

or "Fixes bug." This convention matches up with commit messages generated by commands like `git merge` and `git revert`.

Further paragraphs come after blank lines.

- Bullet points are okay, too

- Typically a hyphen or asterisk is used for the bullet, followed by a single space, with blank lines in between, but conventions vary here

- Use a hanging indent

If you use an issue tracker, add a reference(s) to them at the bottom,  
like so:

Resolves: #123