# What is GitHub

GitHub is a website for storing and editing code, it includes a lot of diverse functionality that allow for multiple programer to work on the same project. The biggest benefit is complete version control.

### Why our group uses GitHub because it ...

* simplifies collaborating on code, allowing multiple people to work on the same project
* makes it easy to give and receive feedback on your code
* makes your code available to the community at large, you can site GitHub repositories and allows you to share projects that haven’t been published in a journal with collaborators and others in the field. Your code is more likely to be used if it is easy to find.
* includes an easy to use “Issue Tracker” that lets you and anyone else highlight problems or planned changes to the code, we use these as checklists for projects and a way to report bugs.
* provides a platform to automatically test code gaurenteeing your code in the current form works the way you think it should.

# Create A GitHub Account

* Make an account on [github.com](https://github.com)
  + make sure to use your uci.edu e-mail
* Follow this link to get the [Student Developer Pack](https://education.github.com/pack)
  + This includes unlimited private repositories
* List your user name on the Research Group GitHub usernames note on Evernote
  + Tell D. Mobley you've done this so he gets you connected with the Mobley Lab group on GitHub

# Syncing Your Account

This section will have instructions for how to tell github your computer has permission to make changes on your account, you can do this for multiple computers and your green planet account. For now follow the instructions here: <https://help.github.com/articles/connecting-to-github-with-ssh/>

# Where do I go Next

This directory includes documentation from the group. It is mained by the current [github/infrastructure training person](https://docs.google.com/document/d/1Eg8RrzOkVbDpDjGlE6ttzCz8QpbMo_QFfazVoXa1hhU/edit?usp=sharing) (currently Caitlin). If you have questions about anything posted here talk to them. Feel free to update any of the writing if you find it to be more clear.

Your next step depends on how you’re going to start using Github:

* If you’re joining an existing project or one of the older graduate students already made a repository, check out [Cloning a Repository](https://docs.google.com/document/d/1kXob_nXz-LxOyJcHq-zgJnskma10SPXm7YCy1dhujq0/edit?usp=sharing) and then [Working in your repository](https://docs.google.com/document/d/1Ivk_IHnZd862YWdlppmqpU_0VRKY9FbbAOZ-XNC-TN8/edit?usp=sharing)
* If you are starting a new project, you’ll need to [Create a New Repository](https://docs.google.com/document/d/1ggRCqv0VcuQnd6Ct_icz6L5P_2XNL3DxNolfG3CppC0/edit?usp=sharing)

# Outside Resources

Git and GitHub are hugely popular tools, while our notes and instructions attempt to provide you with enough instruction to work within the group, you will inevitably have more questions. The Internet has a ton of resources, below is a short list of ones people in our group thought might be useful, if you find others please add them.

|  |  |  |
| --- | --- | --- |
| Link | Format | Why this resource/What is covered |
| [Git and Github for Poets](https://www.youtube.com/watch?v=BCQHnlnPusY&list=PLRqwX-V7Uu6ZF9C0YMKuns9sLDzK6zoiV) | YouTube Video | I have not watched more than 5 minutes of the first video, but this series was made to introduce someone completely outside the world of coding to Git/GitHub. It is an 11 video series that covers everything from why you would want to use GitHub to resolving conflicts when making changes |
| [GitHub Learning Lab](https://lab.github.com/) | Video and written tutorials | You have to sign up for GitHubLearning Lab with your github account, but it has tutorials for how to use git, github, and markdown |
| [Resources to Learn Git](http://try.github.io/) | Tutorial provided by GitHub to learn Git | These tutorials are intended for learning Git, the language you use to interact with github from your terminal |
| [How To Use Version Control](https://www.udacity.com/course/how-to-use-git-and-github--ud775) | Udacity course | This is a free course from Udacity on Git and GitHub, it looked potentially useful, but also like overkill. If following a class like structure is helpful for you this could be good, but I wouldn’t recommend spending too much |
| [Advanced Git Tips for Python Developers](https://realpython.com/advanced-git-for-pythonistas/) | Blog post | This seemed like a good resource if you’re looking to understand some of the finer points of Git and everything you can do with it. |

# Working on shared projects (an argument for GitHub):

* Many projects are shared among multiple people (which is part of the point of git and github)
* Git can help with collaboration management on shared projects and helps to prevent you making conflicting changes. It also helps with merging conflicts when they occur
* There are two main models for working on shared project:
  + The "branch-pull-merge" model, where you create a branch within the original repository (see "branching", below), work on it, and then create a "pull request" to have your changes included in the master branch when you are ready
    - We disfavor this model in favor of fork-pull-merge except when testing concerns dictate it or *when the project is small*
  + The "fork-pull-merge" model, where you create your own fork or copy of the repository (see "forking", below), work on it, and then create a "pull request" to have your changes included in the master branch when you are ready
    - In general we prefer this model except when testing concerns dictate the branch-pull-merge model

# Notifications

* **Don't want GitHub notifications in your e-mail?** Create a private slack channel with a custom e-mail address and set that as your GitHub secondary e-mail address to get your notifications in Slack.