**Git and GitHub**

**Git**

Git is a *version management* tool. It tracks changes made in software projects, and allows you to reverse changes, create alternative versions and combine different versions. It is usually used with software code, but can be used for other documents as well.

This is a tool heavily used by professional software developers, and by large data science teams. Learning to use Git will pay off in the long run, and open additional career options.

Git is a *local tool*, it is installed on your computer. Git can be “activated” for any given directory, and it will track changes to files in that directory for you.

Git is a free, open source product. It was created in 2005 by Linus Torvalds, the main developer of Linux.

**GitHub**

Github is a website for file storage and sharing that incorporates the Git management system. It is somewhat like SharePoint, Google Drive or DropBox in that it allows users to share documents among a team, or to distribute them to people outside the team in a controlled way. The difference is that GitHub has Git built-in, so the version management of documents on GitHub (web storage) and your local version (your computer) is managed and controlled in a helpful and controlled way.

GitHub is now also used as way for programmers and data scientists to share portfolios of their work with one another and with potential employers, another important aspect of GitHub

**Some Basic Terms**

**Repository**-A repository is a storage space for code, you have probably already downloaded R or Python libraries from a repository (CRAN for R, Anaconda, or PyPy for Python). You can create repositories for your projects on GitHub, in this course I have repositories for each DSE5002 module. Repositories can be shared with others, or access can be limited, much like Google Drive or SharePoint.

Repositories in GitHub are always under Git control, so changes are automatically tracked.

**Fork-** To *fork a repository* is to make a copy of the entire repository that is still tracked but not automatically linked to the original repository anymore. It is “fork in the road”, so the project can diverge or go somewhere else.

When you fork someone else’s repository, you can modify your version without altering their version. For example, you will fork my repositories of assignments, then alter them and use Git to manage your work.

**Clone**-To make a copy of repository that is still connected to the original repository. You will typically want to start by creating a repository on GitHub and then **clone** it to your local computer. If you are working on multiple computer systems at once, like an office computer, a laptop and a home computer, you could clone the repository to each system so you can work from anyway and still keep all the changes straight. Team members could each clone a single GitHub repository and work on their pieces of a project independently.

**Commit Changes**- When you make changes to a file that is being managed by Git, you have to *commit* the changes to the file. This updates the Git tracking of the file and logs all the changes in your file.

**Push Changes**- If you are working with a local copy of a file on your computer that is being tracked by Git, and you want to add the changes (and commit them) to a GitHub repository, you *push* the changes from your local directory to GitHub.

**Pull Changes**-If there are recent changes to the files on a GitHub repository, and you want to be sure your local version matches the changes made on GitHub, you can *pull* the changes from GitHub down to your local computer.

*…More Git commands….*

Git has many more commands allowing you to accept or reject changes, remove or reverse changes, combine different forked versions and do a whole bunch of other things to the files in a repository. These make Git (or some other version control) pretty much essential for large software projects, or data projects for that matter, particularly when a large team of people need to collaborate.

We don’t need to know more about Git right now, we can get started with it.

Setting up to use Git and GitHub

1. Start by going to github.com and setting up an account

You will need a username and a password, and you will need to give it an e-mail account to manage it. I would suggest not using your merrimack.edu account, I would use a private, permanent e-mail account to do this. You don’t need to pay for an account, the free version will be fine.

You will need your username and password in the next step, be sure to write them down somewhere.

1. Now install GitHub Desktop

https://docs.github.com/en/desktop/installing-and-authenticating-to-github-desktop/installing-github-desktop

This is a desktop application that will help manage your local Git folders (on your computer) and help you to fork, clone, push and pull from GitHub.

Setting up GitHub Desktop will also install Git itself on your computer, so you can use Git from the command line as will as through the GitHub Desktop app.

Initially, it is easier to start with the App (GitHub Desktop). If you use Git a lot, you will probably find it faster to work with Git directly, using the command line interface.

1. Let’s fork a repository and set it up on your local machine
2. Create a folder somewhere on your local computer called Test\_Module\_2, ideally within in the directory your have created for DSE5002 (this course).
3. Go to your GitHub home page, make sure you are logged in to GitHub
4. On the search bar on the upper right hand side there is a search bar, to let you find other people’s repositories.

Search for my repository, DaveSheets-Merrimack

1. Click on my repository “DSE5002\_Module2”.

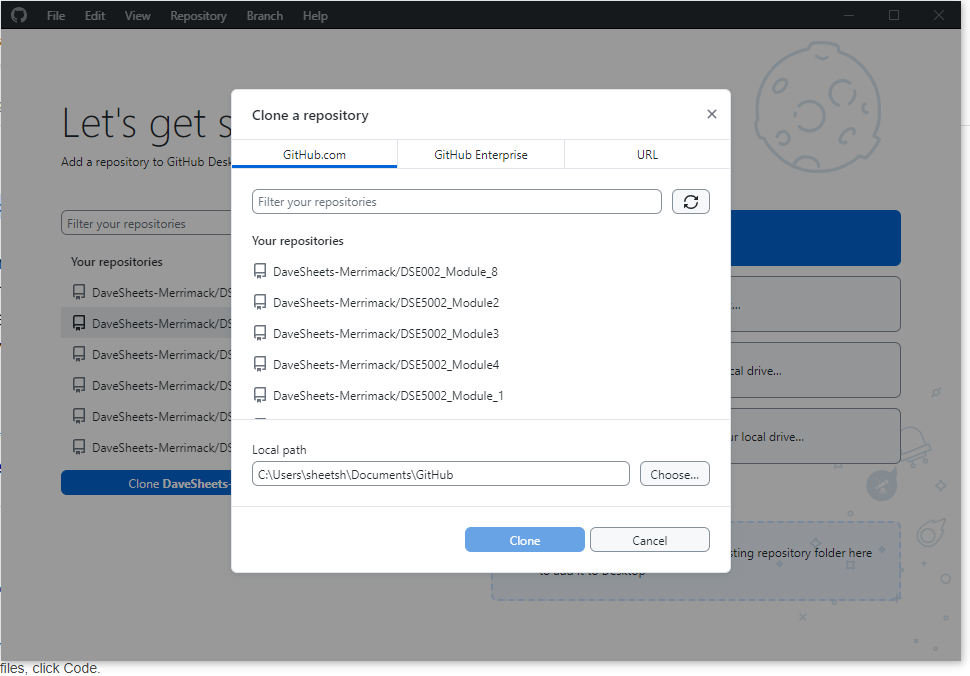
You will see a listing of the files in the repository and an option to fork your own copy.

Fork a copy.

1. Go back to your own home page on GitHub, you should now a copy of the DSE5002\_Module2 repository in your home page
2. Start your copy of GitHub Desktop

It should show you a list of your repositories on GitHub

Under “file” select “Clone a repository”, it will then open a window allowing you to clone the repository from GitHub to your local computer. Use this Window to clone the DSE5002\_Module2 repository onto your hard drive



Here is what the Clone repository option looks like on GitHub Desktop

You can now work on the material for Module\_2 on your local computer!

Later we will talk about pushing the content back to GitHub.