

The Basics of GitHub

The goal of this module is to give you a brief introduction to GitHub and GitHub Classroom.

Git and GitHub

- **Git** is a **distributed Version Control System (VCS)**, which means it is a useful tool for easily tracking changes to your code, collaborating, and sharing. With Git you can track the changes you make to your project so you always have a record of what you've worked on and can easily revert back to an older version if need be. It also makes working with others easier—groups of people can work together on the same project and merge their changes into one final source!
- **GitHub** is a way to use the same power of Git all online with an easy-to-use interface. It's used across the software world and beyond to collaborate and maintain the history of projects.

Understanding the GitHub flow

The GitHub flow is a lightweight workflow that allows you to experiment and collaborate on your projects easily, without the risk of losing your previous work.

- **Repositories**
A repository is where your project work happens--think of it as your project folder. It contains all of your project's files and revision history. You can work within a repository alone or invite others to collaborate with you on those files.
- **Cloning**
When a repository is created with GitHub, it's stored remotely in the cloud. You can clone a repository to create a local copy on your computer and then use Git to sync the two. This makes it easier to fix issues, add or remove files, and push larger commits. You can also use the editing tool of your choice as opposed to the GitHub UI. Cloning a repository also pulls down all the repository data that GitHub has at that point in time, including all versions of every file and folder for the project! This can be helpful if you experiment with your project and then realize you liked a previous version more.

To learn more about cloning, read:

<https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/cloning-a-repository>.

- **Committing and pushing**
Committing and **pushing** are how you can add the changes you made on your local machine to the remote repository in GitHub. That way your instructor and/or

teammates can see your latest work when you're ready to share it. You can make a commit when you have made changes to your project that you want to "checkpoint." You can also add a helpful **commit** message to remind yourself or your teammates what work you did (e.g. "Added a README with information about our project").

Once you have a commit or multiple commits that you're ready to add to your repository, you can use the push command to add those changes to your remote repository. Committing and pushing may feel new at first, but we promise you'll get used to it

GitHub terms to know

- **Repositories**

We mentioned repositories already, they are where your project work happens, but let's talk a bit more about the details of them! As you work more on GitHub you will have many repositories which may feel confusing at first. Fortunately, your ["GitHub dashboard"] (<https://docs.github.com/en/github/setting-up-and-managing-your-github-user-account/about-your-personal-dashboard>) helps to easily navigate to your repositories and see useful information about them. Make sure you're logged in to see it!

Repositories also contain **READMEs**. You can add a README file to your repository to tell other people why your project is useful, what they can do with your project, and how they can use it.

To learn more about repositories read ["Creating, Cloning, and Archiving Repositories"] (<https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/about-repositories>)

and

["About README's"] (<https://docs.github.com/en/github/creating-cloning-and-archiving-repositories/about-readmes>).

- **Branches**

You can use branches on GitHub to isolate work that you do not want merged into your final project just yet. Branches allow you to develop features, fix bugs, or safely experiment with new ideas in a contained area of your repository. Typically, you might create a new branch from the default branch of your repository—main. This makes a new working copy of your repository for you to experiment with. Once your new changes have been reviewed by a teammate, or you are satisfied with them, you can merge your changes into the default branch of your repository.

To learn more about branching, read

[["About Branches"](#)]

(<https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-branches>).

- **Forks**

A fork is another way to copy a repository, but is usually used when you want to contribute to someone else's project. Forking a repository allows you to freely experiment with changes without affecting the original project and is very popular when contributing to open source software projects.

To learn more about forking, read

[["Fork a repo"](#)]

(<https://docs.github.com/en/github/getting-started-with-github/fork-a-repo>)

- **Pull requests**

When working with branches, you can use a pull request to tell others about the changes you want to make and ask for their feedback. Once a pull request is opened, you can discuss and review the potential changes with collaborators and add more changes if need be. You can add specific people as reviewers of your pull request which shows you want their feedback on your changes! Once a pull request is ready-to-go, it can be merged into your main branch.

To learn more about pull requests, read

[["About Pull Requests"](#)]

(<https://docs.github.com/en/github/collaborating-with-issues-and-pull-requests/about-pull-requests>).

- **Issues**

Issues are a way to track enhancements, tasks, or bugs for your work on GitHub. Issues are a great way to keep track of all the tasks you want to work on for your project and let others know what you plan to work on. You can also use issues to tell a favorite open source project about a bug you found or a feature you think would be great to add!

For larger projects, you can keep track of many issues on a project board. GitHub Projects help you organize and prioritize your work and you can read more about them in this

[["About Project boards document"](#)]

(<https://docs.github.com/en/github/managing-your-work-on-github/about-project-boards>). You likely won't need a project board for your assignments, but once you move on to even bigger projects, they're a great way to organize your team's work!

You can also link together pull requests and issues to show that a fix is in progress and to automatically close the issue when someone merges the pull request. To learn more about issues and linking them to your pull requests, read ["About Issues"](<https://docs.github.com/en/github/managing-your-work-on-github/about-issues>).

Using markdown on GitHub

You might have noticed already, but you can add some fun styling to your issues, pull requests, and files. ["Markdown"](<https://guides.github.com/features/mastering-markdown/>) is an easy way to style your issues, pull requests, and files with some simple syntax. This can be helpful to organize your information and make it easier for others to read. You can also drop in gifs and images to help convey your point!

To learn more about using GitHub's flavor of markdown, read ["Basic Writing and Formatting Syntax"](<https://docs.github.com/en/github/writing-on-github/basic-writing-and-formatting-syntax>).

Engaging with the GitHub community

The GitHub community is vast. There are many types of people who use GitHub in their day to day—students like you, professional developers, hobbyists working on open source projects, and explorers who are just jumping into the world of software development on their own. There are many ways you can interact with the larger GitHub community, but here are three places where you can start.

Starring repositories

If you find a repository interesting or you want to keep track of it, star it! When you star a repository it's also used as a signal to surface better recommendations on github.com/explore. If you'd like to get back to your starred repositories you can do so via your user profile. To learn more about starring repositories, read ["Saving Repositories with Stars"](<https://docs.github.com/en/github/getting-started-with-github/saving-repositories-with-stars>).

Following users

You can follow people on GitHub to receive notifications about their activity and discover projects in their communities. When you follow a user, their public GitHub activity will show up on your dashboard so you can see all the cool things they are working on. To learn more about following users, read ["Following People"](<https://docs.github.com/en/github/getting-started-with-github/following-people>).

Browsing GitHub Explore

GitHub Explore is a great place to do just that explore: You can find new projects, events, and developers to interact with.

You can check out the GitHub Explore website at <https://github.com/explore>. The more you interact with GitHub the more tailored your Explore view will be.

Resources

- [A short video explaining what GitHub is]
(<https://www.youtube.com/watch?v=w3jLJU7DT5E&feature=youtu.be>)
- [Git and GitHub learning resources] (<https://docs.github.com/en/github/getting-started-with-github/git-and-github-learning-resources>)
- [Understanding the GitHub flow] (<https://guides.github.com/introduction/flow/>)
- [How to use GitHub branches]
(<https://www.youtube.com/watch?v=H5GJfcp3p4Q&feature=youtu.be>)
- [Interactive Git training materials]
(https://githubtraining.github.io/training-manual/#/01_getting_ready_for_class)
- [GitHub's Learning Lab]
(<https://lab.github.com/>)
- [Education community forum]
(<https://education.github.community/>)
- [GitHub community forum]
(<https://github.community/>)