**Learn how to sign up for a GitHub account and share your code**

**What is GitHub?**

One of the best ways to share what you’ve been learning with other people is to put your code on GitHub. GitHub is both a website and a service that facilitates software development by allowing you to store your code in containers, called *repositories*, and by tracking changes made to your code. In addition, it offers a hosting service and tools to build, test, and deploy code.

GitHub uses [Git, a version-control development tool](https://en.wikipedia.org/wiki/Git), to manage your projects by tracking changes to files and allowing multiple people to work on the same project. Although both GitHub and Git have similar names, GitHub is a service while Git is a development tool that can be used by you outside of and without GitHub.

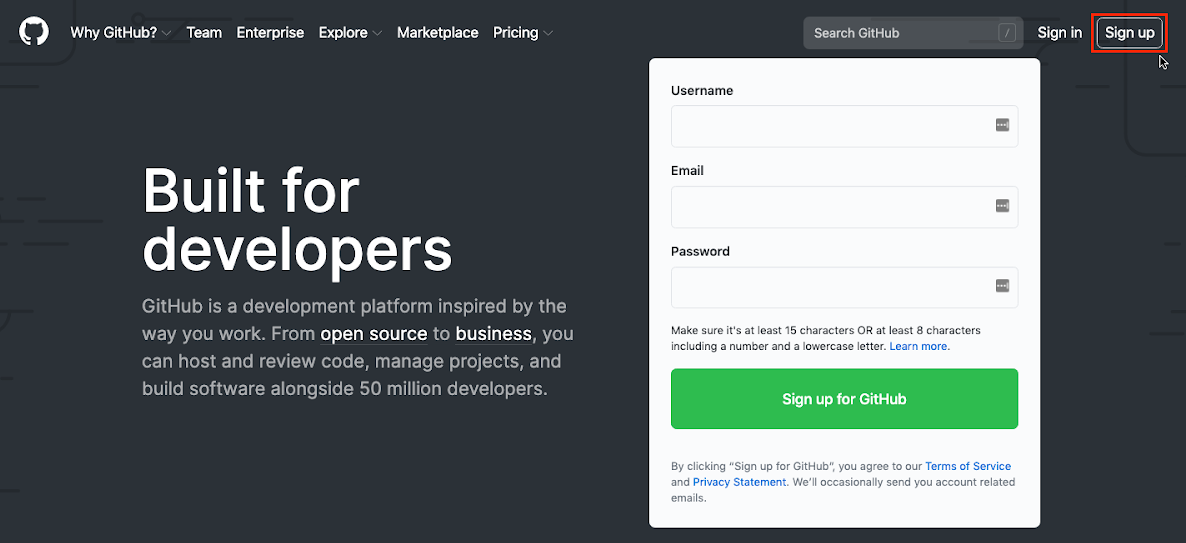
**Why is GitHub important?**

There are many reasons why knowing about GitHub is important for your personal growth as a developer. A large part of the appeal of Github is the access it grants developers to the massive community of developers around the world who openly share their code, projects, and software development tools with each other. Therefore, if you want to continue working on your Git skills, creating your programming portfolio, or finding work, GitHub can help.

**How to Sign Up for an Account**

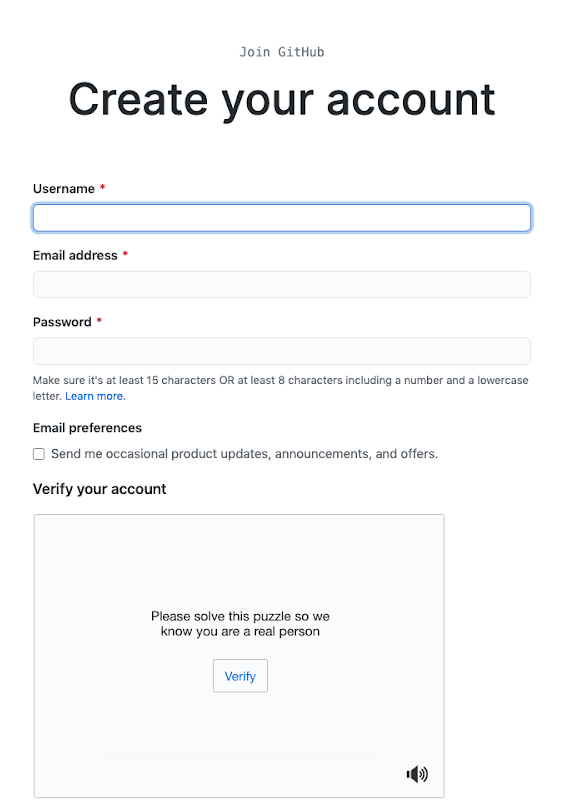
Now that you’re aware of GitHub’s benefits, you probably want to sign up for an account and try it out yourself. First navigate to the home page of the GitHub website, [https://github.com](https://github.com/).

In the upper-right corner click on the **Sign Up** button, as outlined in this screenshot:



**Create Your Account**

You’ll see a page with a form under the heading “Create your account”.



Fill in each field for username, email address, and password. Choosing a username and email are especially important! Be sure to read through the following tips.

**Username**

When choosing a username, it’s wise to choose one you wouldn’t mind future employers or colleagues seeing. A combination of your first and last name like **firstnamelastname** or using initials, like **i\_lastname**, are good because they make it easy to find you on GitHub or identify you when you make [pull requests](https://docs.github.com/en/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/about-pull-requests) or [reviews](https://docs.github.com/en/free-pro-team@latest/github/collaborating-with-issues-and-pull-requests/about-pull-request-reviews). Remember, you’re likely using this account to share or access code.

Also be aware that usernames are first come, first serve and may not be available if someone else already claimed the username. Additionally, usernames may only contain alphanumeric characters and hyphens are not allowed at the beginning or end.

**Email Address**

Like with usernames, pick an email that you’re comfortable sharing with peers and potential hires. Because of the way Git works, it’s important to note that your email can be exposed publicly when you make a pull request or merge in code to a repository, making it visible to anyone looking thr**GIT BRANCHING**

**Generalizations**

Let’s take a moment to review the main concepts and commands from the lesson before moving on.

* Git *branching* allows users to experiment with different versions of a project by checking out separate *branches* to work on.

The following commands are useful in the Git branch workflow.

* git branch: Lists all a Git project’s branches.
* git branch branch\_name: Creates a new branch.
* git checkout branch\_name: Used to switch from one branch to another.
* git merge branch\_name: Used to join file changes from one branch to another.
* git branch -d branch\_name: Deletes the branch specified.

ough your projects. When you sign up for a new GitHub account, your email address is hidden by default.

**F**Congratulations, you now know enough to start collaborating on Git projects! Let’s review.

* A remote is a Git repository that lives outside your Git project folder. Remotes can live on the web, on a shared network or even in a separate folder on your local computer.
* The Git Collaborative Workflow are steps that enable smooth project development when multiple collaborators are working on the same Git project.

We also learned the following commands

* git clone: Creates a local copy of a remote.
* git remote -v: Lists a Git project’s remotes.
* git fetch: Fetches work from the remote into the local copy.
* git merge origin/master: Merges origin/master into your local branch.
* git push origin <branch\_name>: Pushes a local branch to the origin remote.

Git projects are usually managed on Github, a website that hosts Git projects for millions of users. With Github you can access your projects from anywhere in the world by using the basic workflow you learned here.

**inish Creating Your Account**

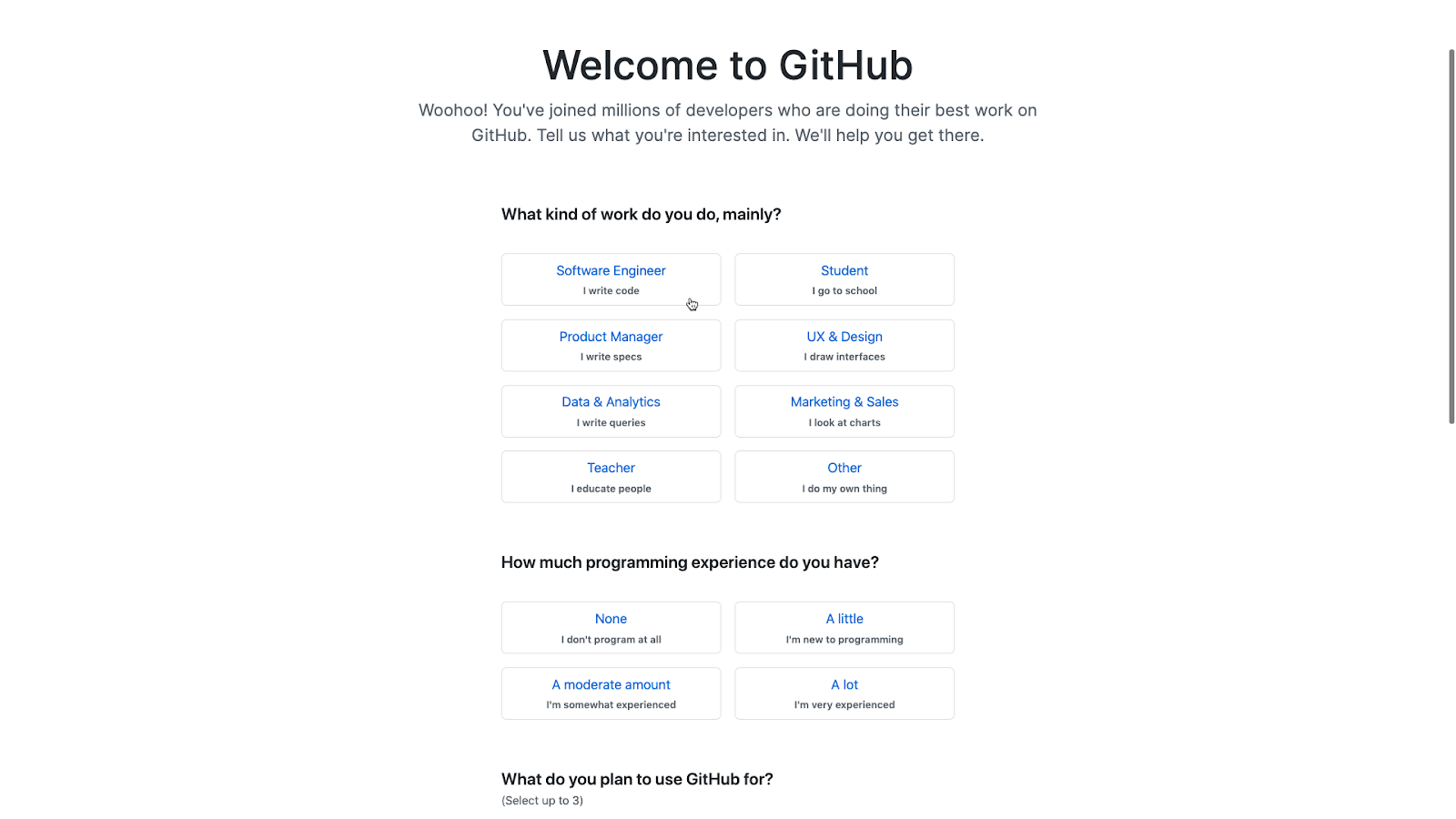
Lastly, fill out the password field. When you’re done filling out the various fields, verify your account. You will receive an email from GitHub prompting you to verify.

Once you see a green checkmark, click on the blue **Create an Account** button.

Go to your email account, find the email from GitHub, and click on the button inside it to finish creating your account.

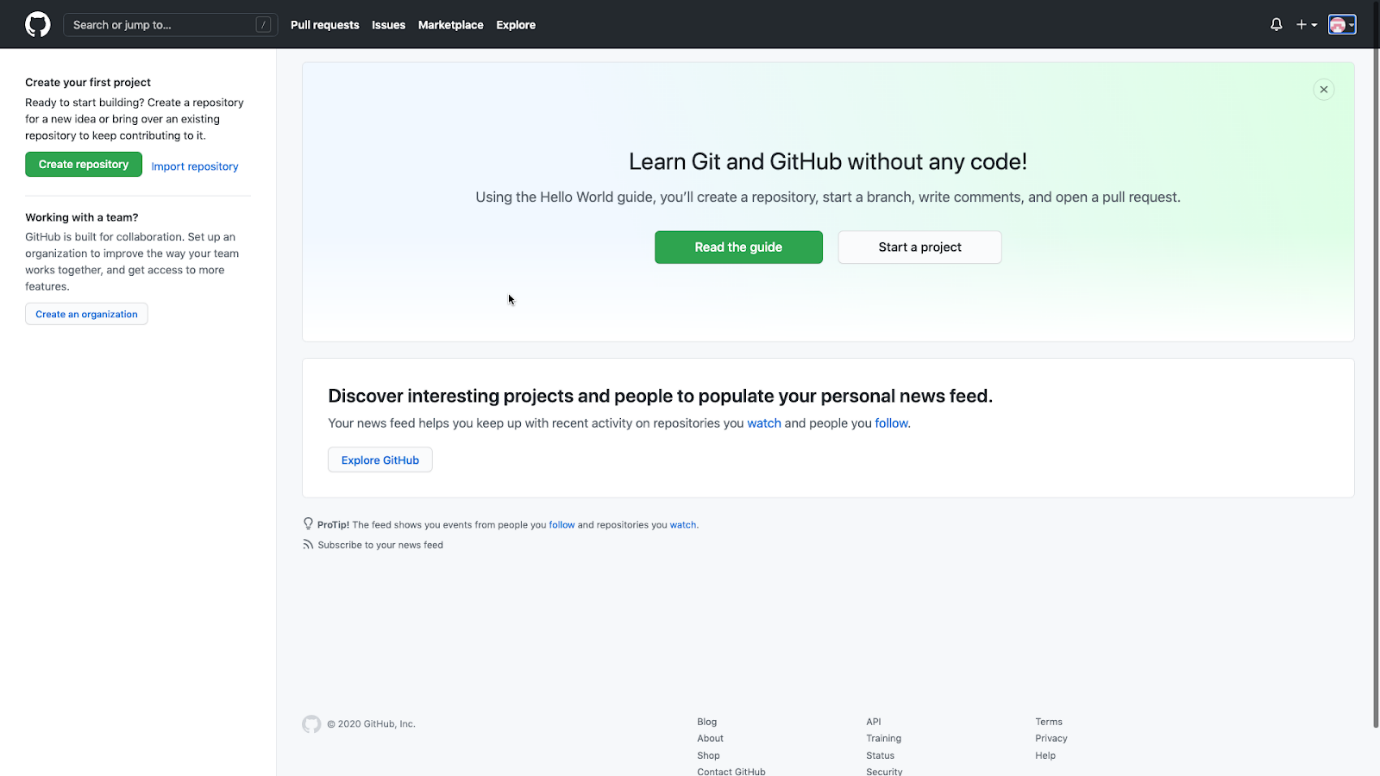
**Settings**

After successfully creating an account, you should see a page display asking you “What do you want to do first?” Go through the steps to complete your own set up process. You should now see a welcome page:



You can either answer the optional questions or move on by clicking on the **Complete setup** button to finish creating your account.

Your browser should display a personal dashboard with a section for your projects and some messages:



Now you have your own GitHub account! You can continue to customize your account by:

* [setting your email to private](https://github.blog/2017-04-11-private-emails-now-more-private/)
* [setting up two-factor authentication](https://github.blog/2013-09-03-two-factor-authentication/)

That’s it, you now have your very own GitHub account. 🎉

**Recap**

With more people working remotely and with teams distributed across different countries and timezones, GitHub and Git can be valuable tools for collaborating on projects. You can also use GitHub to work on any file-based project such as writing documentation.

Let’s review what you did in this article:

* Learned that GitHub is both a website and service for storing and sharing code
* Learned that Github uses Git to facilitate software development by tracking changes
* Created your own GitHub account
* Enabled security features like keeping your email private and turning on two-factor authentication

**Going Further**

Once you feel comfortable navigating GitHub consider doing the following:

* [Add information about yourself](https://docs.github.com/en/free-pro-team@latest/github/setting-up-and-managing-your-github-profile/personalizing-your-profile) in your profile, including an avatar, bio, location, etc.
* [Set your status](https://docs.github.com/en/free-pro-team@latest/github/setting-up-and-managing-your-github-profile/personalizing-your-profile#setting-a-status) in your profile to let people know what you’re doing.
* If you feel comfortable enough with Git, you can take a look at [this article on GitHub Pages](https://pages.github.com/), GitHub’s hosting service that allows you to create a personal website, “project site”, based on a repository to make your portfolio site.
* If you’re interested in paid accounts, GitHub has a [pricing page](https://github.com/pricing) with various types of accounts and features you can look at.

Show the world what you can do with your code!