**GitHub**

GitHub is a Web-based revision control hosting service for software development and code sharing. GitHub offers free public repositories and collaborations for open-source software creation and charges small and large businesses for private repositories and collaboration. From a social perspective, GitHub allows users to change, adapt and improve software — each project is called a "fork" — in its public repositories. Users can work together in teams or alone. Public users create profiles which display repositories and public activity and help programmers find projects.

GitHub was started in 2008 and based on Git, an open-source code management system built created by [Linus Torvalds](http://searchenterpriselinux.techtarget.com/definition/Linus-Torvalds) to make software builds faster. After a security breach on the standard Linux [kernel](http://searchenterpriselinux.techtarget.com/definition/kernel) distribution site at kernel.org in 2011, Torvalds released the 3.0 Linux kernel on GitHub. Other groups with GitHub public repositories include Red Hat's [JBoss](http://whatis.techtarget.com/definition/JBoss) software, Twitter, Microsoft Windows [Azure](http://searchcloudcomputing.techtarget.com/definition/Windows-Azure), the [Open Compute](http://searchdatacenter.techtarget.com/definition/open-compute-open-compute-project) project and [Mozilla](http://searchsoa.techtarget.com/definition/Mozilla).

### The “Git” in GitHub

To understand GitHub, you must first have an understanding of Git. Git is an open-source version control system that was started by Linus Trovalds – the same person who created Linux. Git is similar to other version control systems – [Subversion](http://www.howtogeek.com/66731/version-tracking-with-subversion-svn-for-beginners/), CVS, and Mercurial to name a few.

**Version control systems**

So, Git is a “version control system,” what’s that mean? When developers are creating something (an application, for example), they are making constant changes to the code and releasing new versions, up to and after the first official (non-beta) release.

Version control systems keep these revisions straight, and store the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute.

Similarly, people who have nothing to do with the development of a project can still download the files and use them. Most Linux users should be familiar with this process, as using Git, Subversion, or some other similar method is pretty common for downloading needed files, especially in preparation for compiling a program from source code (a rather common practice for Linux geeks).

In case you are wondering why Git is the preferred version control system of most developers, it has multiple advantages over the other systems available, including a more efficient way to store file changes and ensuring file integrity. If you’re interested in knowing the details, check out [this page](http://git-scm.com/book/en/Getting-Started-Git-Basics) to read a thorough explanation on how Git works.

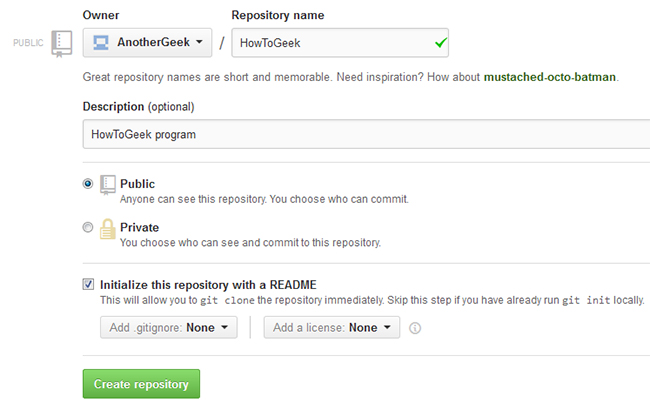
### The “Hub” in GitHub

We’ve established that Git is a version control system, similar but better than the many alternatives available. So, what makes GitHub so special? Git is a command-line tool, but the center around which all things involving Git revolve – effectively, the Hub, is GitHub.com, where developers can store their projects and network with likeminded people.

Let’s go over a few of the main reasons that geeks like to use GitHub, and learn some terminology along the way.

**Repository**

A repository is a location where all the files for a particular project are stored, usually abbreviated to “repo.” Each project will have its own repo, and can be accessed by a unique URL.



**Forking a repo**

“Forking” is when you create a new project based off of another project that already exists. This is an amazing feature that vastly encourages the further development of programs and other projects. If you find a project on GitHub that you’d like to contribute to, you can fork the repo, make the changes you’d like, and release the revised project as a new repo. If the original repository that you forked to create your new project gets updated, you can easily add those updates to your current fork.

**Pull requests**

You fork a repository, make a great revision to the project, and want it to be recognized by the original developers, maybe even included in the official project/repository. You can do so by creating a pull request, so the authors of the original repository can see your work, and then choose whether or not to accept it into the official project. Whenever you issue a pull request, GitHub provides a perfect medium for you and the project’s maintainer to communicate.

**Social networking**

The social networking aspect of GitHub is probably its most powerful feature, and is what allows projects to grow more than anything else. Each user on GitHub has their own profile, which can act like a resume of sorts, showing your past work and contributions to other projects via pull requests.

Project revisions are able to be discussed publicly, so a mass of experts can contribute knowledge and collaborate to advance a project forward. Before the advent of GitHub, developers interested in contributing to a project would usually need to find some means of contacting the authors, probably by email, and then have to convince them that their contribution is legit and they can be trusted.

**Changelogs**

When multiple people are collaborating on a project, it’s really hard to keep track of who changed what, and to keep track of the revisions that took place. GitHub takes care of this problem by keeping track of all the changes that have been pushed to the repository.

### GitHub isn’t just for developers

All this talk about how GitHub is ideal for programmers may have you believing that they are the only ones who will find it useful. Although it’s a lot less common, GitHub can actually be used for any types of files – so if you have a team that is constantly making changes to a word document, you can actually use GitHub as your version control system.  This practice isn’t common as there are better alternatives, but keep it in mind.

#### Scope

GitHub is mostly used for code.

In addition to [source code](https://en.wikipedia.org/wiki/Source_code), GitHub supports the following formats and features:

* Documentation, including automatically-rendered [README](https://en.wikipedia.org/wiki/README) files in a variety of [Markdown](https://en.wikipedia.org/wiki/Markdown)-like file formats (see [README files on GitHub](https://en.wikipedia.org/wiki/README#On_GitHub))
* [Issue tracking](https://en.wikipedia.org/wiki/Issue_tracking) (including feature requests) with labels, milestones, assignees and a search engine.
* [Wikis](https://en.wikipedia.org/wiki/Wikis)
* Pull requests with code review and comments.
* Commits history.
* Graphs: pulse, contributors, commits, code frequency, punch card, network, members.
* Integrations Directory
* Unified and split diffs.
* Email notifications.
* Option to subscribe someone to notifications by @ mentioning them.
* Emojis
* Small [Websites](https://en.wikipedia.org/wiki/Website) can be hosted from public repositories on GitHub. The URL format is http://*username*.github.io.
* Nested [task-lists](https://en.wikipedia.org/wiki/To_do_list) within files
* Visualization of [geospatial](https://en.wikipedia.org/wiki/Geospatial_analysis) data\* 3D render files that can be previewed using a new integrated STL file viewer that displays the files on a 3D canvas.[[16]](https://en.wikipedia.org/wiki/GitHub#cite_note-3d-16) The viewer is powered by [WebGL](https://en.wikipedia.org/wiki/WebGL) and [Three.js](https://en.wikipedia.org/wiki/Three.js).
* [Photoshop](https://en.wikipedia.org/wiki/Photoshop)'s native PSD format can be previewed and compared to previous versions of the same file.

#### Licensing of repositories

GitHub's Terms of Service do not require public software projects hosted on GitHub to meet the [Open Source Definition](https://en.wikipedia.org/wiki/Open_Source_Definition). For that reason, it is advisable for users and developers intending to use a piece of software found on GitHub to read the [software license](https://en.wikipedia.org/wiki/Software_license) in the repository (usually found in a top-level file called "LICENSE", "LICENSE.txt", or similar) to determine if it meets their needs.

### GitHub Enterprise

GitHub Enterprise is similar to GitHub's public service but is designed for use by large-scale enterprise software development teams where the enterprise wishes to host their repositories behind a corporate [firewall](https://en.wikipedia.org/wiki/Firewall_%28computing%29).

### Education program

GitHub launched a new program called the GitHub Student Developer Pack to give students free access to popular development tools and services. GitHub partnered with [Bitnami](https://en.wikipedia.org/wiki/Bitnami), [Crowdflower](https://en.wikipedia.org/wiki/Crowdflower), [DigitalOcean](https://en.wikipedia.org/wiki/DigitalOcean), [DNSimple](https://en.wikipedia.org/wiki/DNSimple), [HackHands](https://en.wikipedia.org/wiki/HackHands), [Namecheap](https://en.wikipedia.org/wiki/Namecheap), Orchestrate, Screenhero, [SendGrid](https://en.wikipedia.org/wiki/SendGrid), [Stripe](https://en.wikipedia.org/wiki/Stripe_%28company%29), [Travis CI](https://en.wikipedia.org/wiki/Travis_CI) and [Unreal Engine](https://en.wikipedia.org/wiki/Unreal_Engine) to launch the program.