**GITHUB**

**What is git?**

Git is open source version control system. Git is smiler to other version control system Subversions, CVS and Mercurial. Developers create something (an app, for example) they make constant changes to the code, releasing new versions up to and after first officials (non-beta) release.

Version control systems keep these revisions straight, sorting the modifications in a central repository. This allow 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 changes, download them and contribute.

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).

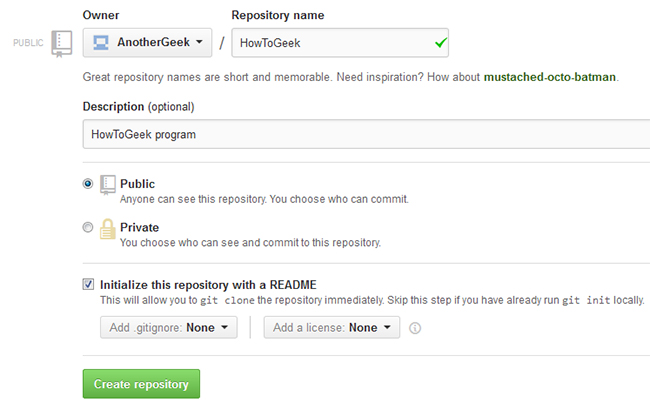
**The “Hub” in GitHub**

Git is a command-line tool, but the center around which all things involving Git revolve is the hub GitHub.com where developers store their projects and network with like-minded people.

### **Repository**

A repository (usually abbreviated to “repo”) is a location where all the files for a particular project are stored. Each project has its own repo, and you can access it with a unique URL.

A **repository** can be a place where multiple databases or files are located for distribution over a network, or a **repository** can be a location that is directly accessible to the user without having to travel across a network. Also see data warehouse and data mining.



### **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’ve forked a repository, made 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. 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 main project’s maintainer to communicate.

### **Social networking**

The social networking aspect of GitHub is probably its most powerful feature, allowing projects to grow more than just about any of the other features offered. Each user on GitHub has their own profile that acts like a resume of sorts, showing your past work and contributions to other projects via pull requests.

Project revisions can 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 convince them that they can be trusted and their contribution is legit.

### **Changelogs**

When multiple people collaborate on a project, it’s hard to keep track revisions—who changed what, when, and where those files are stored. GitHub takes care of this problem by keeping track of all the changes that have been pushed to the repository.

**Why GitHub**

Following features shows why we use GitHub

* Code Review
* Project Management
* Integrations
* Actions
* Package registry
* Team management
* Social coding
* Documentations
* Code hosting

**Code Review**

**Write better code**

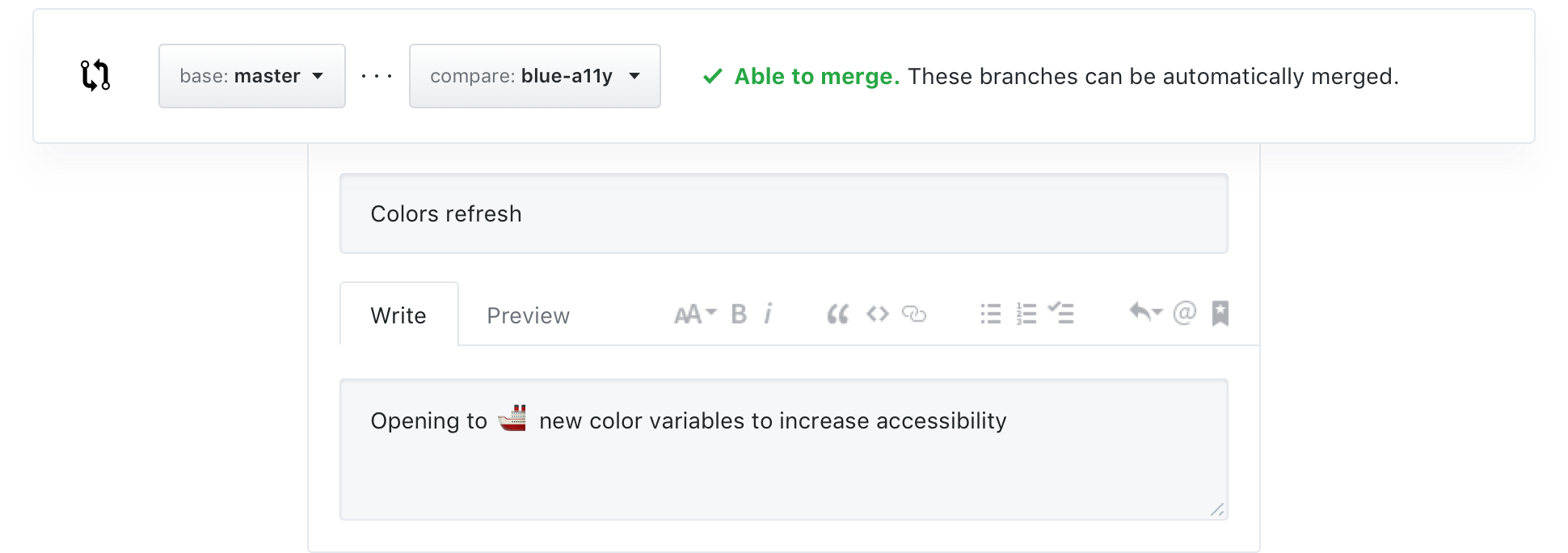
On GitHub, lightweight code review tools are built into every pull request. Your team can create review processes that improve the quality of your code and fit neatly into your workflow.

## Start with a pull request

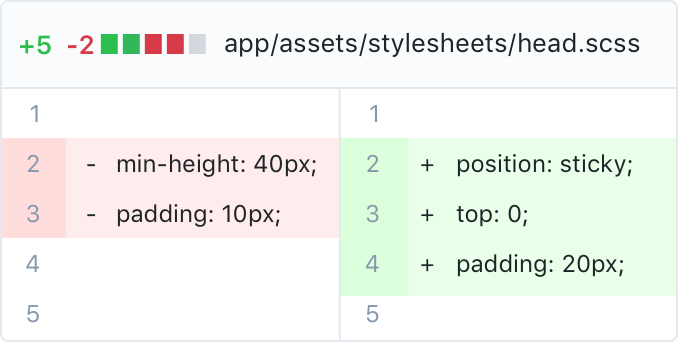
## Pull requests are fundamental to how teams review and improve code on GitHub. Evolve projects, propose new features, and discuss implementation details before changing your source code

### **Make a change**

Start a new feature or propose a change to existing code with a pull request—a base for your team to coordinate details and refine your changes.

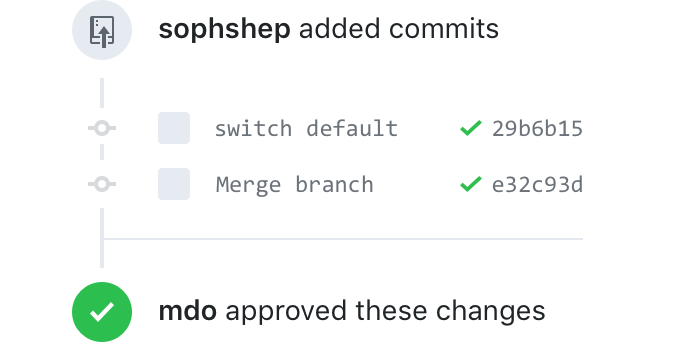


### **See every update**



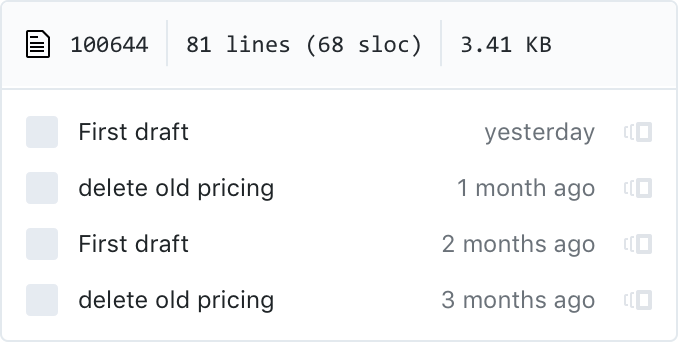
### **Diffs**

Preview changes in context with your code to see what is being proposed. Side-by-side Diffs highlight added, edited, and deleted code right next to the original file, so you can easily spot changes.



### **History**

Browse commits, comments, and references related to your pull request in a timeline-style interface. Your pull request will also highlight what’s changed since you last checked.

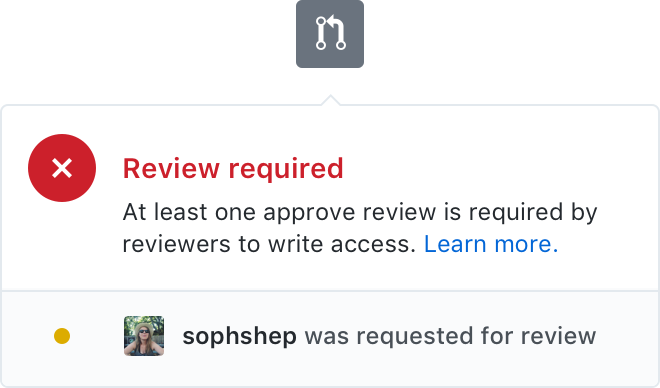


### **Blame**

See what a file looked like before a particular change. With [blame view](https://github.blog/2017-01-18-navigate-file-history-faster-with-improved-blame-view/), you can see how any portion of your file has evolved over time without viewing the file’s full history.

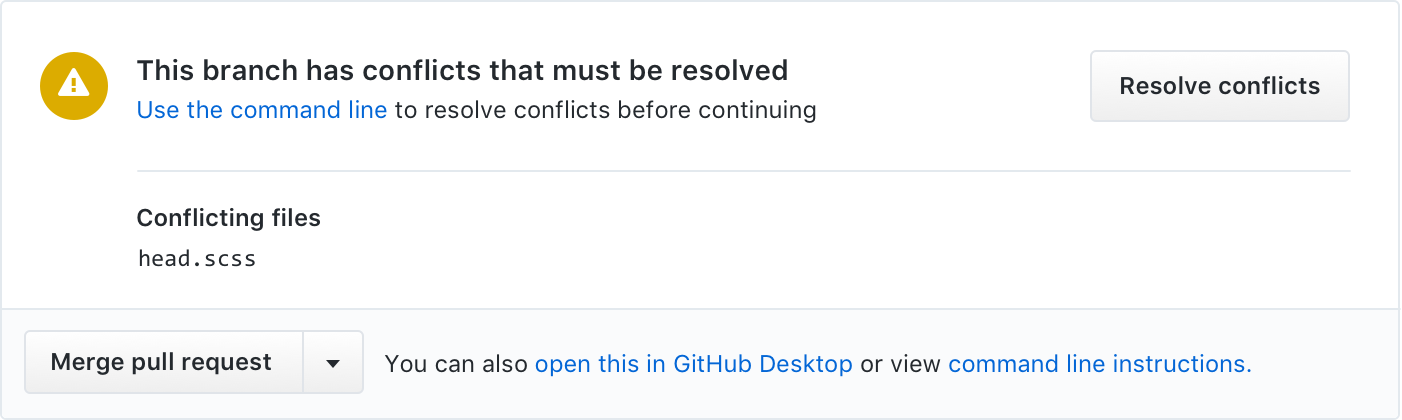
#### **Review requests**

If you’re on the other side of the code, requesting peer reviews is easy. Add users to your pull request, and they’ll receive a notification letting them know you need their feedback.



#### **Reviews**

You can’t always avoid conflict. Merge pull requests faster by resolving simple merge conflicts on GitHub—no command line necessary.



# **Project management**

On GitHub, project managers and developers coordinate, track, and update their work in one place, so projects stay transparent and on schedule.

# **Integration**

# **Built to grow**

Between applications and custom tools, there’s no limit to the ways you can build on GitHub. Fine tune your process as your team changes, update tools as technologies shift, and find new ways to work better.

## Discover new tools

## From continuous integration to code review, find tools for every step of the development process and start using them in minutes. Browse and buy apps from GitHub Marketplace with your GitHub account—no need for multiple accounts or payment methods.

# **Package registry**

With GitHub Package Registry you can safely publish and consume packages within your organization or with the entire world.

# **Documentations**

Quality documentation is a hallmark of any healthy software project. On GitHub, you can create well-maintained docs and make sure they receive the high level of care they deserve.

# **Code Hosting**

GitHub is one of the largest code hosts in the world with over 100 million\* projects. Private, public, or open source, all repositories are equipped with tools to help you host, version, and release code

## How to use Git Bash

## Download Git.

## Go to the location or folder where your project is located.

## Right click on that folder and select Git bash here.

## Git bash window will open.

## Initialize local directory as a git repository with command git init.

## Add files to your new local repository with command git add.

## Commit the files that you have staged in your local repository with command git commit –m “First Project”

## To check the status write command git status.

## Now we add URL for the remote repository where our local repository will be pushed, command we will write is git remote add origin *remote repository URL.*

## Push the changes in your local repository to Git hub. Command use git push --set-upstream origin master or git push –u origin master.

## Python is open source or not?

Python is developed under an OSI-approved open source license, making it freely usable and distributable, even for commercial use. Python has many libraries available for developers under open source licenses.

## Java is open source or not?

Today most major components of Java are available under open source licenses, and those which are not available under open licenses typically have drop-in replacements which are open.

## What is open source?

## The term "open source" refers to something people can modify and share because its design is publicly accessible.

## What is open source software?

Open source software is software with source code that anyone can inspect, modify, and enhance.

"Source code" is the part of software that most computer users don't ever see; it's the code computer programmers can manipulate to change how a piece of software—a "program" or "application"—works. Programmers who have access to a computer program's source code can improve that program by adding features to it or fixing parts that don't always work correctly.