

Getting started with GitHub

A beginner's guide.
(There's no code in this slide deck!)



Presented by Quinn Supplee
<https://github.com/quinnns>

What is GitHub?

- ☐ GitHub is a code hosting platform for version control and collaboration.
- ☐ It lets you collaborate with others down the hall, or across the world.
- ☐ But you can use it for *a lot more* than just software code.
- ☐ Many technical recruiters and hiring managers are in the habit of reviewing applicants' GitHub profiles when making a hiring decision.

Git vs GitHub

- ☐ Git, itself, is an Open Source tool originally created by Linus Torvalds (creator of Linux).
- ☐ GitHub is a commercial (for profit) company based in San Francisco, whose goal is to make using Git as easy as possible.
- ☐ GitHub layers a lot of special sauce and extra features on top of Git, but you can use Git on your own, or use a different 3rd party service such as GitLab or BitBucket.
- ☐ You can also install and configure your own Git server if you want full control.

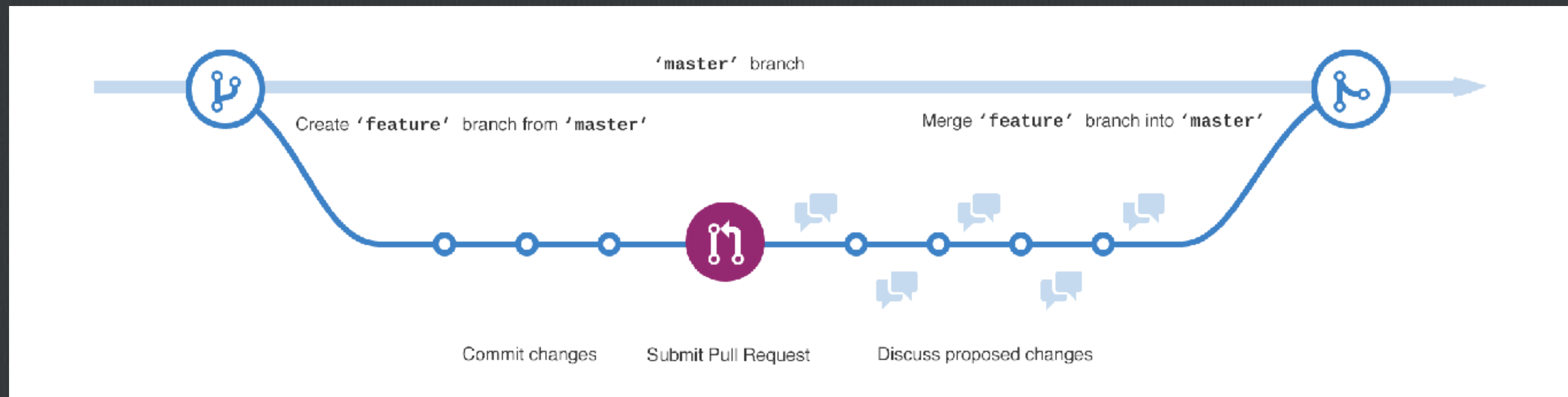
A few basic terms and concepts

- ❑ Every project is hosted in a “repository”.
- ❑ Repositories can be *public* (anyone can view, download, and potentially contribute to your code) or *private* (only you, and people you invite, can view or interact with your code).
- ❑ Public repositories are *free*, but private repositories require a *paid* GitHub account (currently \$7/mo).
- ❑ Git automatically creates a permanent history of all changes you’ve ever made to your project, and you can easily *go back in time* to retrieve older versions of your code if you mess something up.

Git is distributed

- ☐ One of Git's best features is that it is distributed.
- ☐ That is, every single "working copy" on any user's computer, stores the entire project history. There's not necessarily a "central server" the way there is with older version control systems such as CVS or SVN.
- ☐ Although, to be precise, GitHub is a central server and if you use GitHub exclusively, you give up some of that distributed nature.
- ☐ GitHub has had some downtime (who hasn't), but they're generally quite reliable.

Branches and merging



- ❑ Git allows you to create as many “branches” as you want. You can use branches for experimenting with new ideas, adding a feature, fixing a bug, or anything else that you want to work on separately without stepping on your existing code.

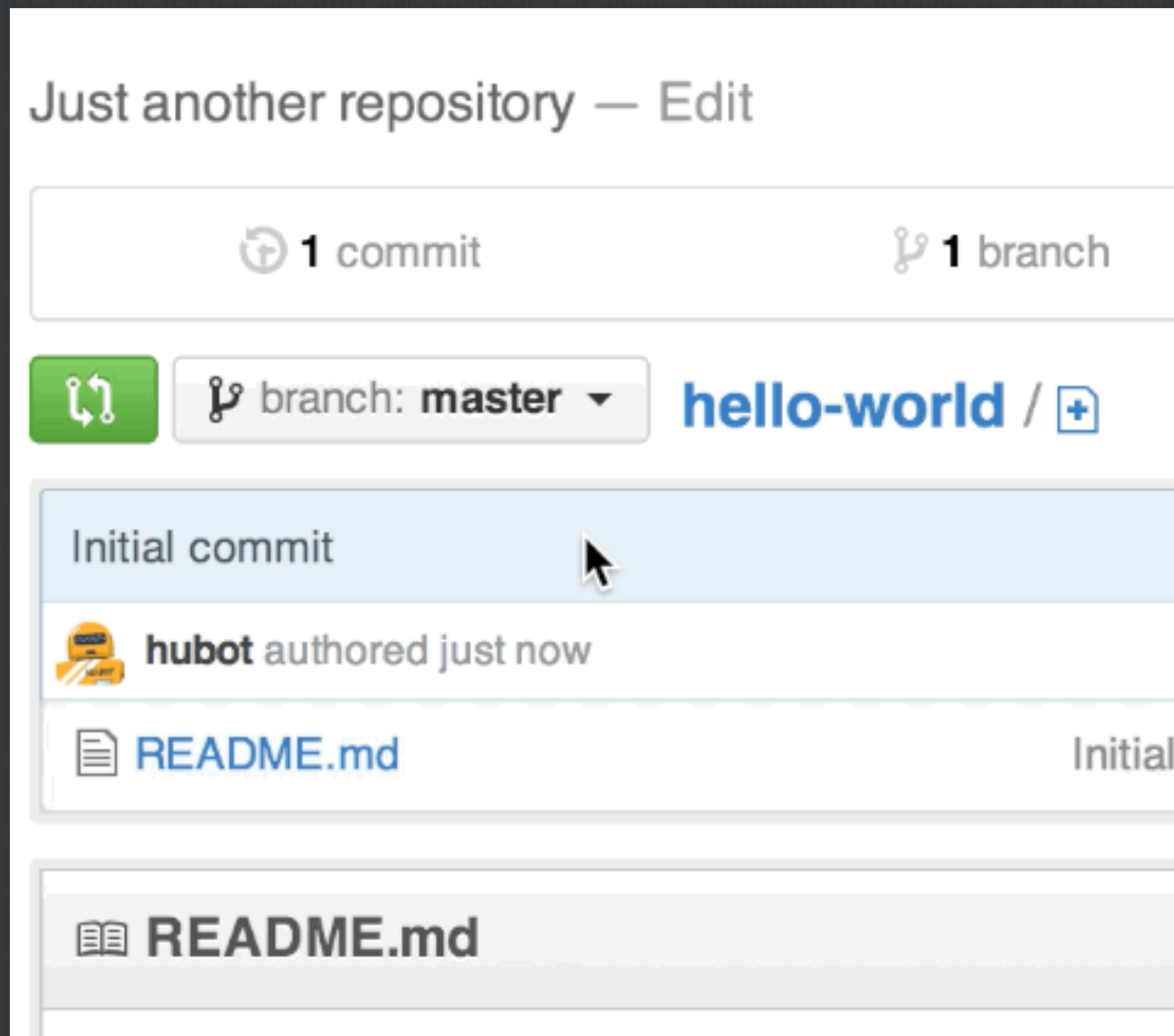
The work tree and staging area

- As you create or modify files, they will automatically be added to your *work tree*. You can easily add these files to your *staging area*, which is a place to assemble your code before adding it to a *commit*.

Make and commit changes

- ☐ On GitHub, saved changes are called *commits*. Each commit has an associated *commit message*, which is a description explaining why a particular change was made.
- ☐ Commit messages capture the history of your changes, so other contributors can understand what you've done and why. (Or, maybe just for you to remember what you did 6 months ago.)
- ☐ You can always review all commits to see what has been done in the past, and even roll back to any commit at any time if necessary.

Create a new branch



- ☐ You can create as many branches as you need, as long as they each have a unique name.
- ☐ Some branches might exist only on your computer; others might be “pushed” to GitHub and be available to all collaborators.

Merging

- ☐ Once you have finished your changes in your new branch, you can *merge* the changes back into your main (*master*) branch.
- ☐ Git will automatically try to merge your changes into your repository. If files conflict, it will give you an error and will show you where the conflicts are so you can manually resolve the changes.
- ☐ The branch you were previously working on still exists and you can go back to it at any time.
- ☐ You can use the built in “git diff” tool to view the differences between files.

Pull requests

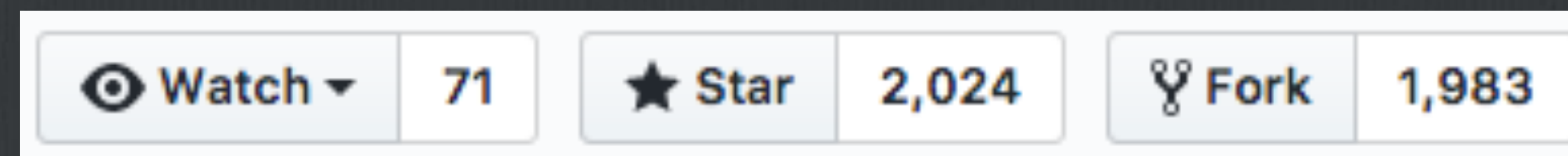
- ❑ A “pull request” is the main way to collaborate on GitHub.
- ❑ Basically, you’re *requesting* to make some changes to a repository, and asking someone (a *project maintainer*) to approve those changes and *pull* your code into the repository.
- ❑ You can use GitHub’s *@mention* feature to ask for feedback or approval from an individual or a team. Just ask for help by referring to a user’s handle.
- ❑ You can even make pull requests on your own repositories and then approve them yourself. This is a great way to get comfortable with pull requests.

Forking projects



- ☐ Sometimes you want to make some changes to someone's code for your own use but don't want to open a pull request and ask for your changes to be merged into the main repository.
- ☐ To do this, you can easily *fork* the project. This creates a totally separate copy of the repository under your own account. You can then change it any way you want to (please respect any Open Source or other licenses that may be present).

Social features



- ☐ GitHub has a lot of features to enable and encourage collaborative editing and social sharing.
- ☐ You can “follow” another user (it’s not at all creepy).
- ☐ You can “watch” a project and all updates to it will show up on your own dashboard.
- ☐ You can also “star” a project which will let the creator know you’re interested in it, but it won’t show all updates on your dashboard the watch “watch” does.
- ☐ The “Explore” page on GitHub shows interesting projects.

Issue tracking

- ❑ Every GitHub repository features an “*issue tracker*”.
- ❑ This is a place to report bugs, ask for enhancements, or just ask questions about a project.
- ❑ You can use this in your own projects as a quick and easy project management tool.
- ❑ You can invite your own clients to collaborate with you, even if they can’t code, and they can use the issue tracker to report bugs or just keep an eye on progress.
- ❑ Issues have some nice organizational features such as *milestones*, *labels*, and *references*.

GitHub Pages

- ❑ One of GitHub's coolest features is GitHub Pages. This tool lets you publish static (HTML, CSS, JavaScript) web pages for free. I maintain my own personal site on GitHub Pages.
- ❑ You can even use your own domain name with GitHub Pages.
- ❑ You can either write your HTML in the usual way, or use one of a number of Static Site Generators to do so.
- ❑ GitHub Pages natively supports a popular Static Site Generator called Jekyll.
<https://jekyllrb.com/>

Documenting your project

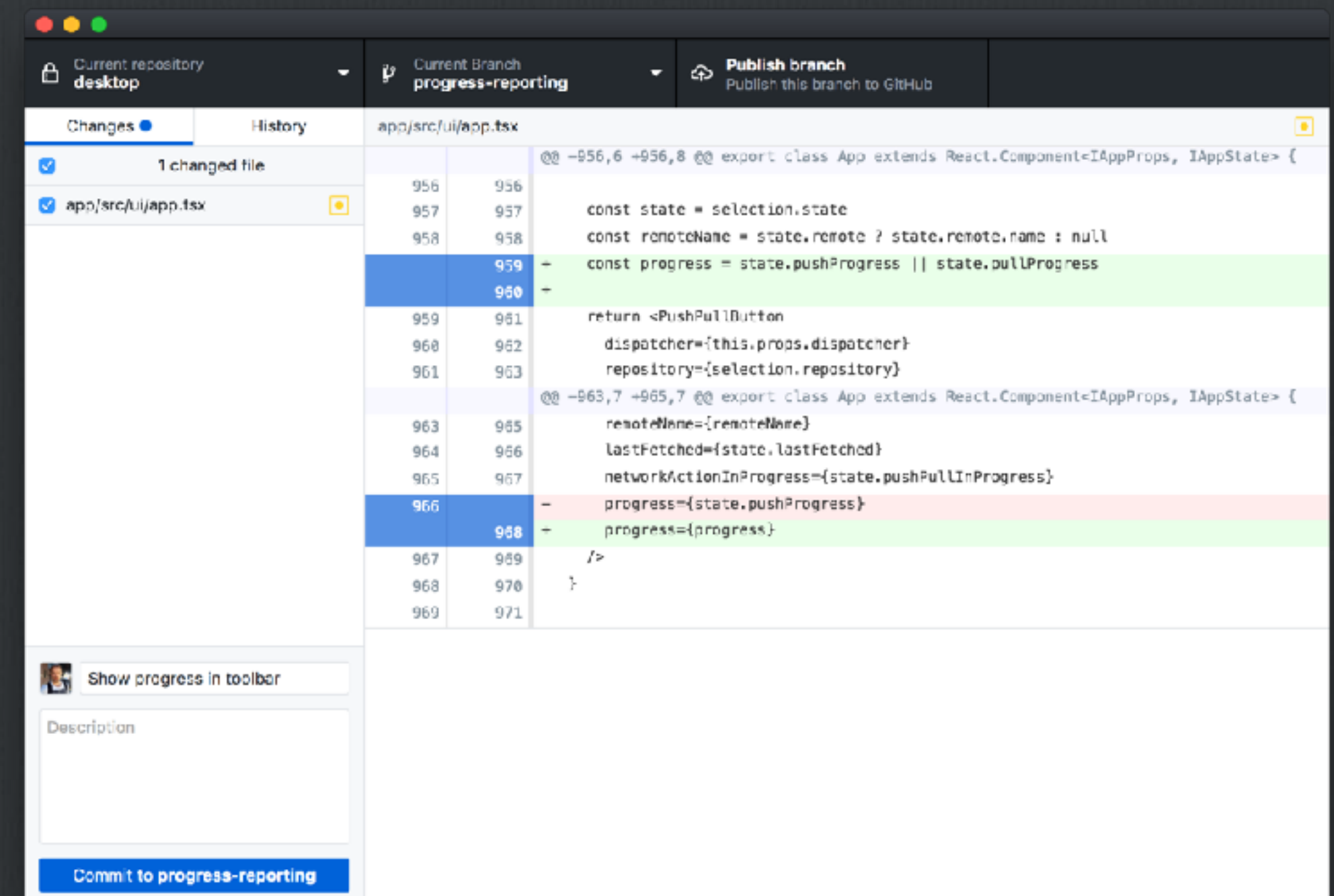
- ☐ Every GitHub project should have a *README* file.
- ☐ This file, as the name implies, is a quick overview of what the project is for and how to use it.
- ☐ If you need more comprehensive documentation, you can use the built-in *Wiki* feature to create a full-featured set of interlinked documentation pages.

Command-line tools

- ☐ The primary interface to Git (and GitHub) is via your command-line.
- ☐ MacOS and Linux have Git built-in.
- ☐ Windows users will need to install a tool called Git Bash
<https://git-for-windows.github.io/>

Graphical user interfaces

- There are a number of GUI interfaces for Git. Some popular ones that work with any git repository (not just GitHub) include *Tower*, *GitX*, *GitKraken*, *SourceTree*, and *Tortoise*.
- GitHub also offers a cross-platform app just for GitHub projects called *GitHub Desktop*. This is a great way to get started if you aren't yet comfortable working with the command line.
<https://desktop.github.com/>



Markdown

```
It's very easy to make some words **bold** and other words  
*italic* with Markdown. You can even  
[link to Google!](http://google.com)
```

It's very easy to make some words **bold** and other words *italic* with
Markdown. You can even [link to Google!](http://google.com)

- ❑ GitHub makes use of a simple text-based format called Markdown. You can use Markdown in *issue comments*, *READMEs*, and any file with the *.md* or *.markdown* file extension.
- ❑ You can use Markdown to quickly format text using a simple syntax, which will then be converted into HTML automatically.

Pricing

- ❑ GitHub is *free* for public repositories. You can make a free account and get started with no financial commitment.
- ❑ Personal accounts with private repositories start at \$7/month. This gets you unlimited private repositories.

Developer		Team	Business	
\$7		\$9	\$21	\$21*
per month		per user / month	per user / month	per user / month
Includes:		Includes:	Hosted on GitHub.com	GitHub Enterprise
Personal account		Organization account	Organization account	Multiple organizations
Unlimited public repositories		Unlimited public repositories	SAML single sign-on	SAML, LDAP, and CAS
Unlimited private repositories		Unlimited private repositories	Access provisioning	Access provisioning
Unlimited collaborators		Team and user permissions	24/5 support with 8-hour response time	24/7 support for urgent issues
Free for students as part of the Student Developer Pack .		Starting at \$25 / month which includes your first 5 users.	99.95% Uptime SLA	Advanced auditing
			Team sync (Coming 2017)	Host on your servers, AWS, Azure, or GCP

Storage limitations

- ☐ Every GitHub repository has a “soft limit” of 1GB. If you get near this limit, GitHub will send you an email.
- ☐ There’s also a hard limit of 100MB per file in the repository. If you try to push a file to your repository that is larger than 100MB, it will fail with an error message.
- ☐ If you need to store larger files such as binary assets, there’s a tool called Git Large File Storage (LFS) for this purpose. GitHub offers Git LFS as a paid add-on.

What next?

- ☐ Create your account at <https://www.github.com>
- ☐ Get basic tutorials at <https://guides.github.com>
- ☐ Download GitHub Desktop at <https://desktop.github.com>
- ☐ Refer to the official Git documentation at <https://git-scm.com>
- ☐ Check out Static Site Generation at <https://www.jekyllrb.com>
- ☐ Read about Large File Storage at <https://git-lfs.github.com>

Thank you!



Find me online at
www.quinnsupplee.com