|  |
| --- |
|  |
| GitHub |
|  |

**Inhaltsverzeichnis**

**Es wurden keine Einträge für das Inhaltsverzeichnis gefunden.**

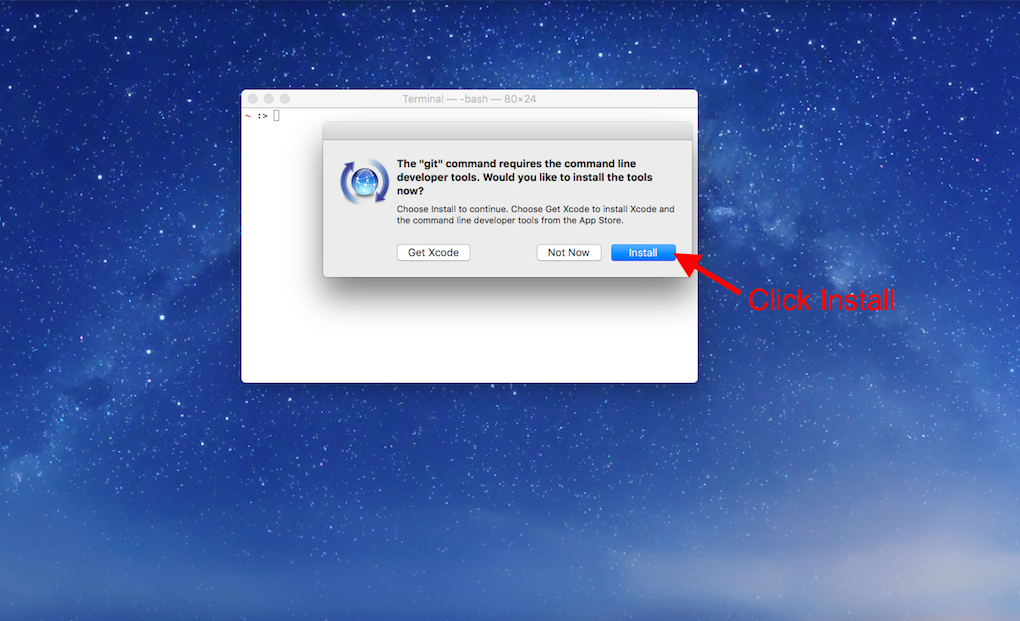
# Launch

### Mac users:

1. Launch the **Terminal** application. You can find it in **/Applications/Utilities/**. You can also use the **Spotlight** search tool (the little magnifying glass in the top right of your screen) to search for **Terminal**. Once **Spotlight** locates it, click on the result that says **Terminal**.

2. When **Terminal** opens, type in git and press enter.

3. If you don't already have Git installed, a dialog will appear saying that "The 'git' command requires the command line developer tools. Would you like to install the tools now?" Click "Install".



4. When the download finishes, the installer will go away on its own signifying that Git is now installed! Click "Done" to finish the installation process.

5. Navigate to GitHub's articles on setting up your [Git username](https://help.github.com/articles/set-up-git/) and [email](https://help.github.com/articles/setting-your-email-in-git/) and follow the instructions for each using Terminal.

6. GitHub offers two authentication options, HTTPS and SSH, to keep your work secure. This is a security measure which prevents anyone who isn't authorized from making changes to your GitHub repository. In this article, we will use HTTPS. Navigate to GitHub's article on [caching your password](https://help.github.com/articles/caching-your-github-password-in-git/) and follow the instructions to configure your computer to be able to use HTTPS.

Now skip down to the "Try it Out!" section below.

# Try It Out!

Now you have everything you need to practice your Git skills on your local computer. Take a moment to run the commands below to initialize a Git repository. We will use this Git repository again later in this tutorial so make sure you complete these steps exactly as described.

1. mkdir git\_practice to make a new directory to practice.
2. cd git\_practice to make the new directory your working directory.
3. git init to turn the current, empty directory into a fresh Git repository.
4. echo "Hello Git and GitHub" >> README.txt to create a new README file (more on this later) with some sample text.
5. git add README.txt to add the new file to the Git staging area.
6. git commit -m "First commit" to make your first commit with the new README file.

## Your First Remote Repository on GitHub

Finally, we'll create a repository on GitHub and then link it to a local repository on your computer. This allows you to backup your work constantly and safely, so you never need to worry about losing your work again!

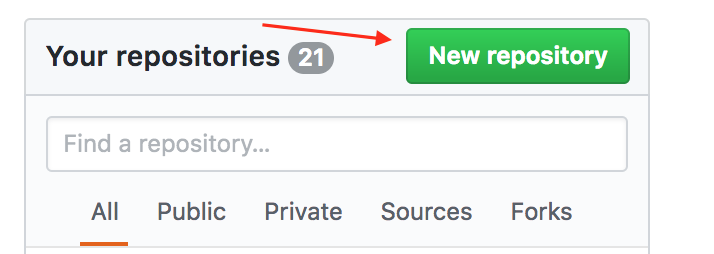
Now, let's connect our local Git repository to GitHub.

### Instructions

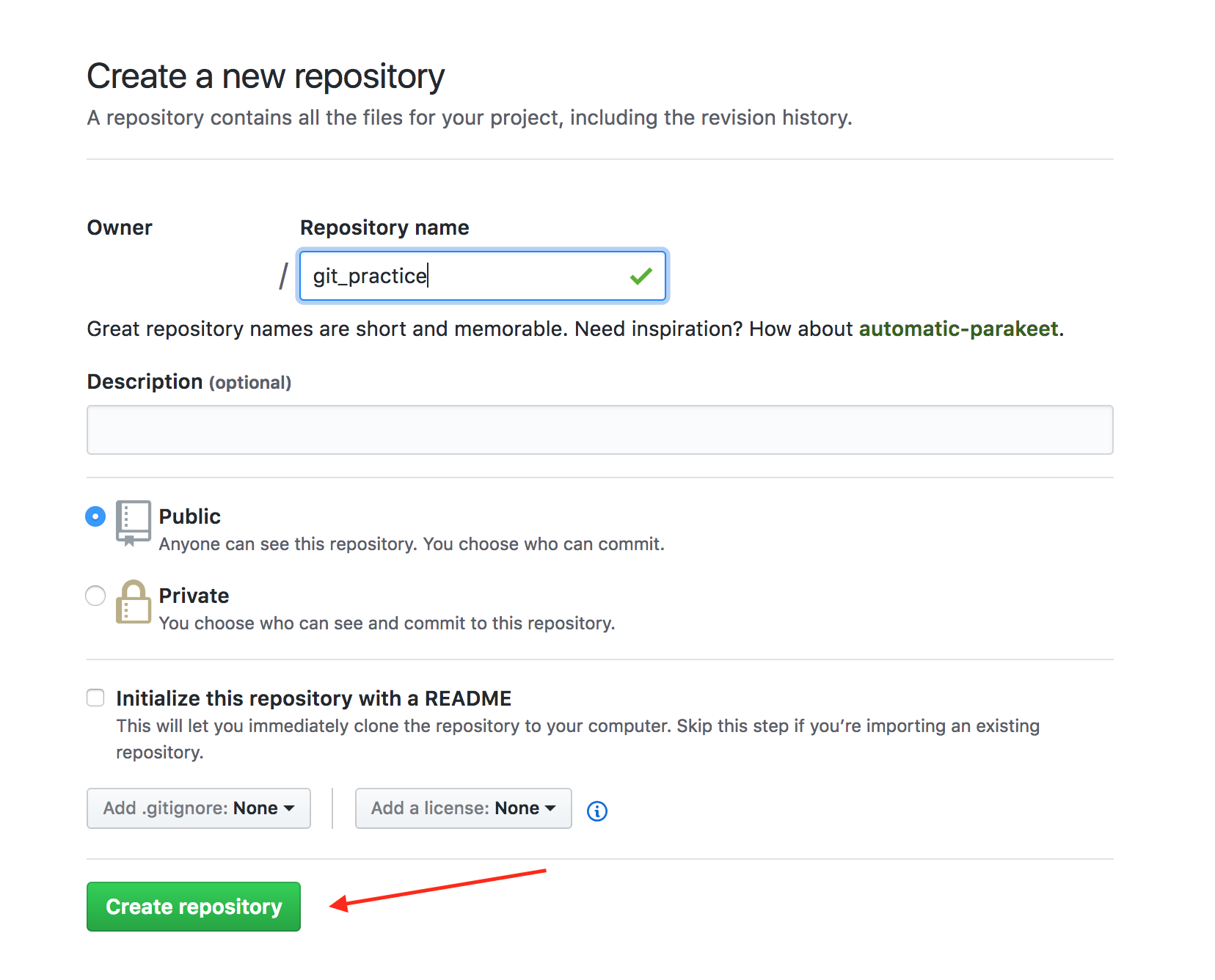
1. In your Command Line Interface, make sure your current working directory is your new Git repository. Navigate there if not.

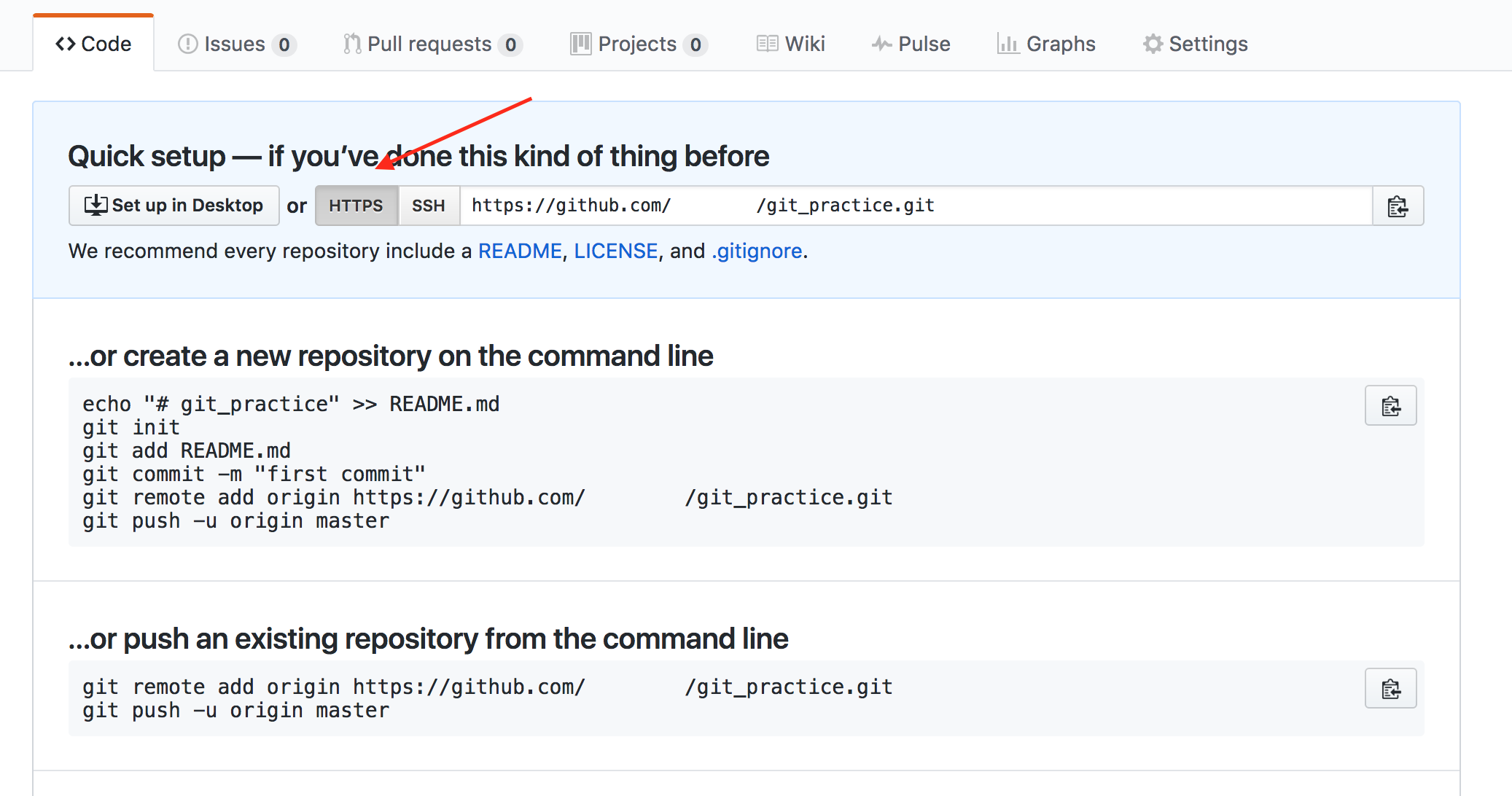
2. Check the status of which files and folders are new or have been edited. There should be no files modified.

$ git status

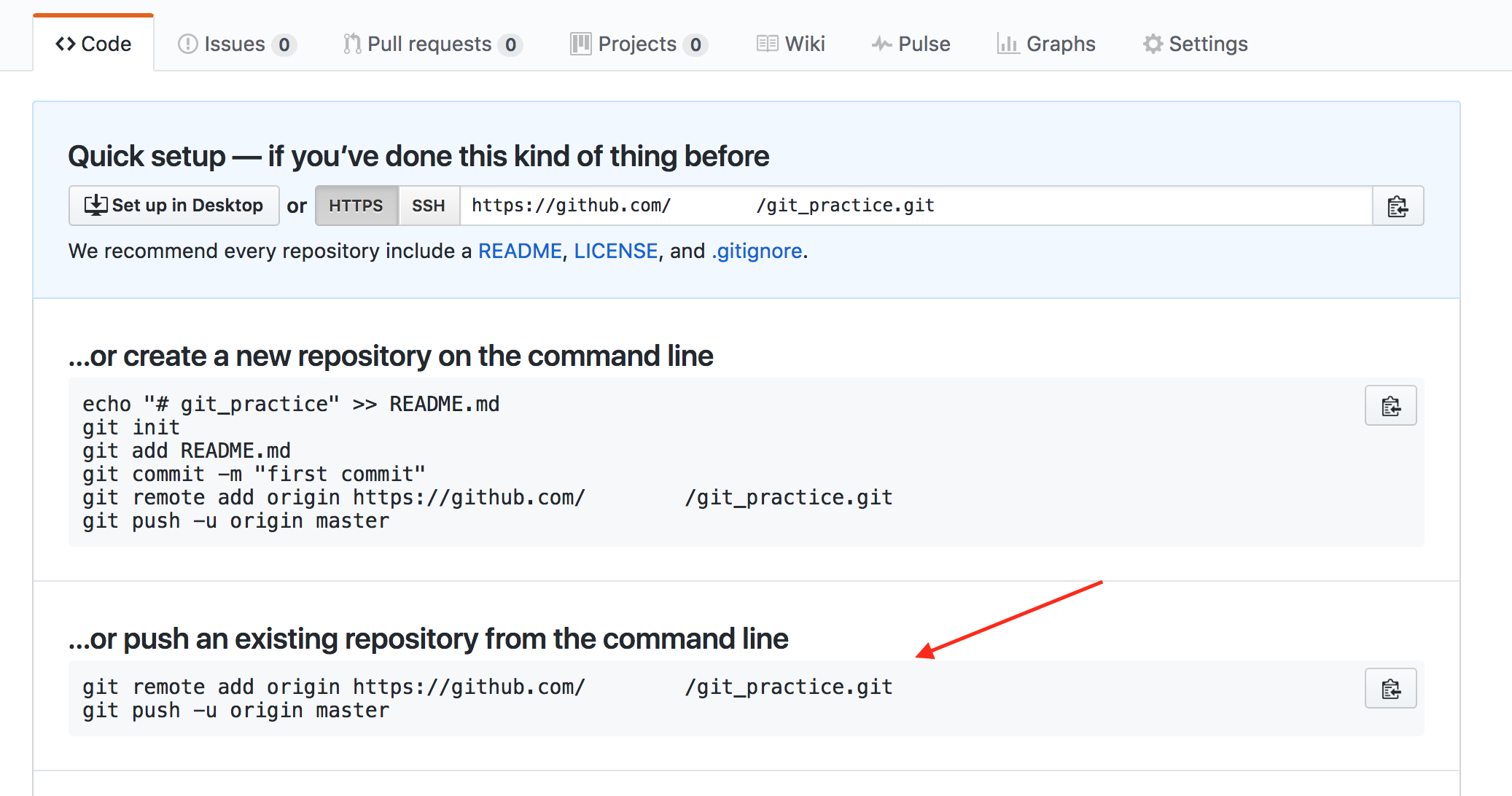
3. On GitHub, create a new repository by clicking the **New repository** button on the home page.

4. On the new repository page, give your repository a name. It's not necessary, but it would be convenient to name it the same as the directory, **git\_practice**. After naming the repository, click **Create repository**.



5. After creating a repository, GitHub displays the repository page. At the top of the page, make sure "HTTPS" is selected.

6.The repository is empty, so it's time to connect it to your existing work. Copy the Git commands on the GitHub page, under the title "...or push an existing repository from the command line", and paste them into your Command Line Interface. Running these commands will add a remote repository, and then push your local repository to the remote repository.

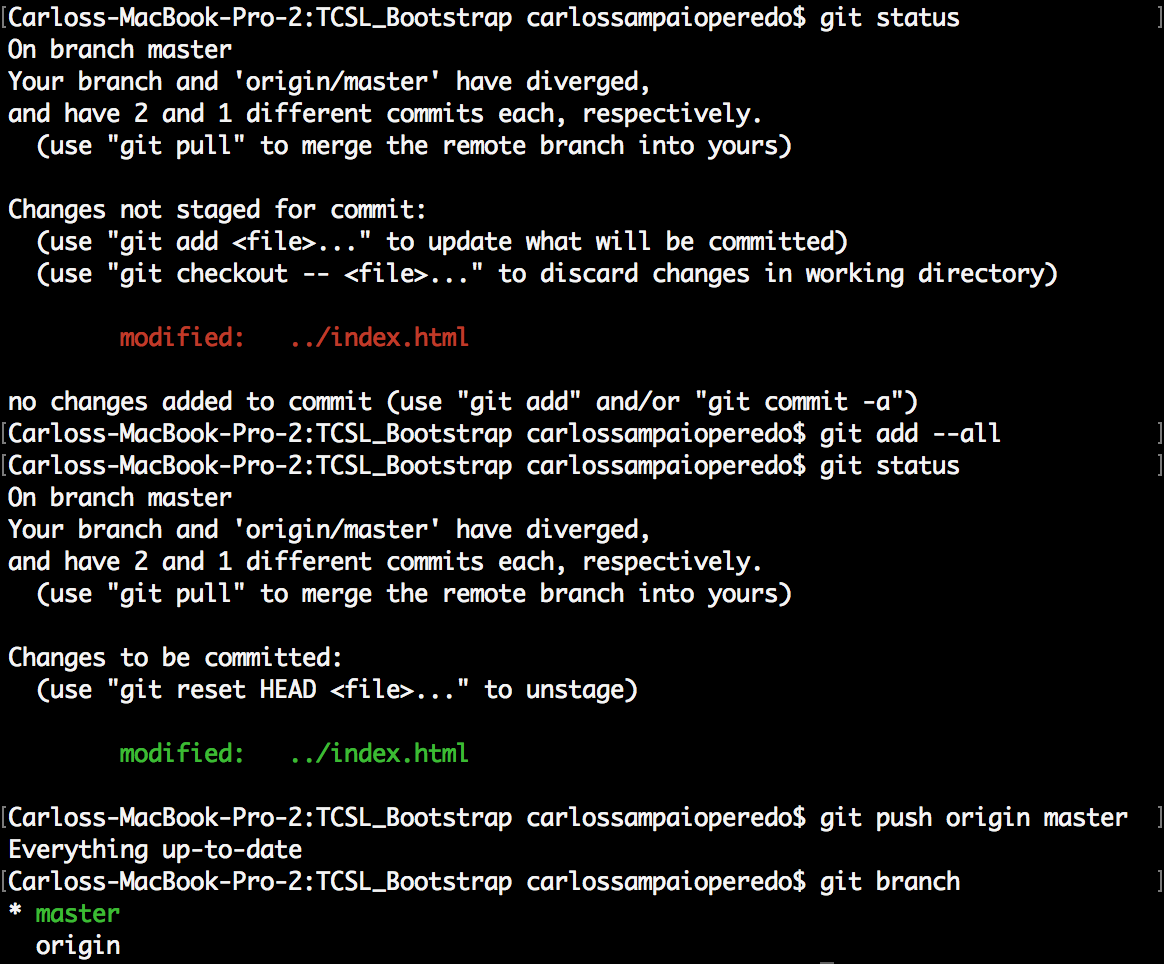
When asked for a username and password, type in your GitHub username and password and press enter after each. Don't be alarmed if you can't see the characters you are typing, they are intentionally hidden as a security measure.

**Note:** If you set up two-factor authentication with GitHub (don't worry if you didn't), follow [these instructions](https://help.github.com/articles/creating-a-personal-access-token-for-the-command-line/) to generate an OAuth token to be used instead of your password in bash. By default, GitHub does not set up two-factor authentication. If you are not familiar with two-factor authentication, you don't have to generate an OAuth token.

7. Once your Command Line Interface reports that the push is complete, refresh the page on GitHub. You should now see the text you wrote earlier in the README file, "Hello Git and GitHub."

GitHub automatically displays the contents of a file named **README.txt** if it exists in the repository. The README file is the perfect place to write a description of your project.

There you have it! Your first GitHub repository, linked to your local Git repository. You've taken some huge leaps, so be proud! Now you can use your knowledge of Git to track progress on your local computer, and push that progress to GitHub whenever you want. You can rest easy knowing that each step of your progress is safely stored in GitHub.



cd <zum Pfad>

git add –all

git push origin master

git commit -m “comment”

git push -u origin master

