GitHub Help Document

This document is to serve as a reference to using GitHub for academic collaboration purposes.

Table of Contents

[2 Introduction 2](#_Toc31738007)

[2.1 What is GitHub 2](#_Toc31738008)

[2.2 Setup 2](#_Toc31738009)

[2.2.1 Software 2](#_Toc31738010)

[2.2.2 Project Collaborators (Teachers/Administors) 3](#_Toc31738011)

[2.2.3 Students 4](#_Toc31738012)

# Introduction

## What is GitHub

GitHub is software that has been developed for primarily programmers. Whenever more than one programmer must collaborate on a project, there has always been the issue of sharing code from a project with others. GitHub allows for others to replicate a programming environment while maintaining a certain level of integrity to the layout of the paths and the current state of the project.

What does this mean for us as programmers? Let’s take the example of having two programmers working on a project together. Whenever an update to code is made by one programmer, there is an issue of getting the updated code to his or her partner. They could maintain a string of emails; but, this quickly gets cluttered and disorganized. Using GitHub, when an update is made to the code, the programmer could commit his changes to the GitHub project and the other programmer could pull this change from the project origin and have a copy of the most recent code directly to his or her system.

In an academic sense, this could be applied to a classroom setting. A professor could maintain administrative files in one folder and course materials such as code and power point presentations in another folder. If a student wanted a copy of these materials, the student could simply fork the branch to their own filesystem and have an exact copy of the entire project. This includes code, administrative documents, power points, and due dates.

## Setup

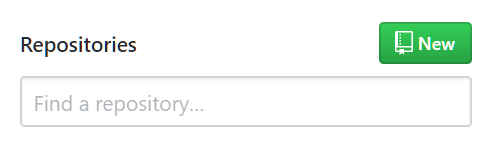
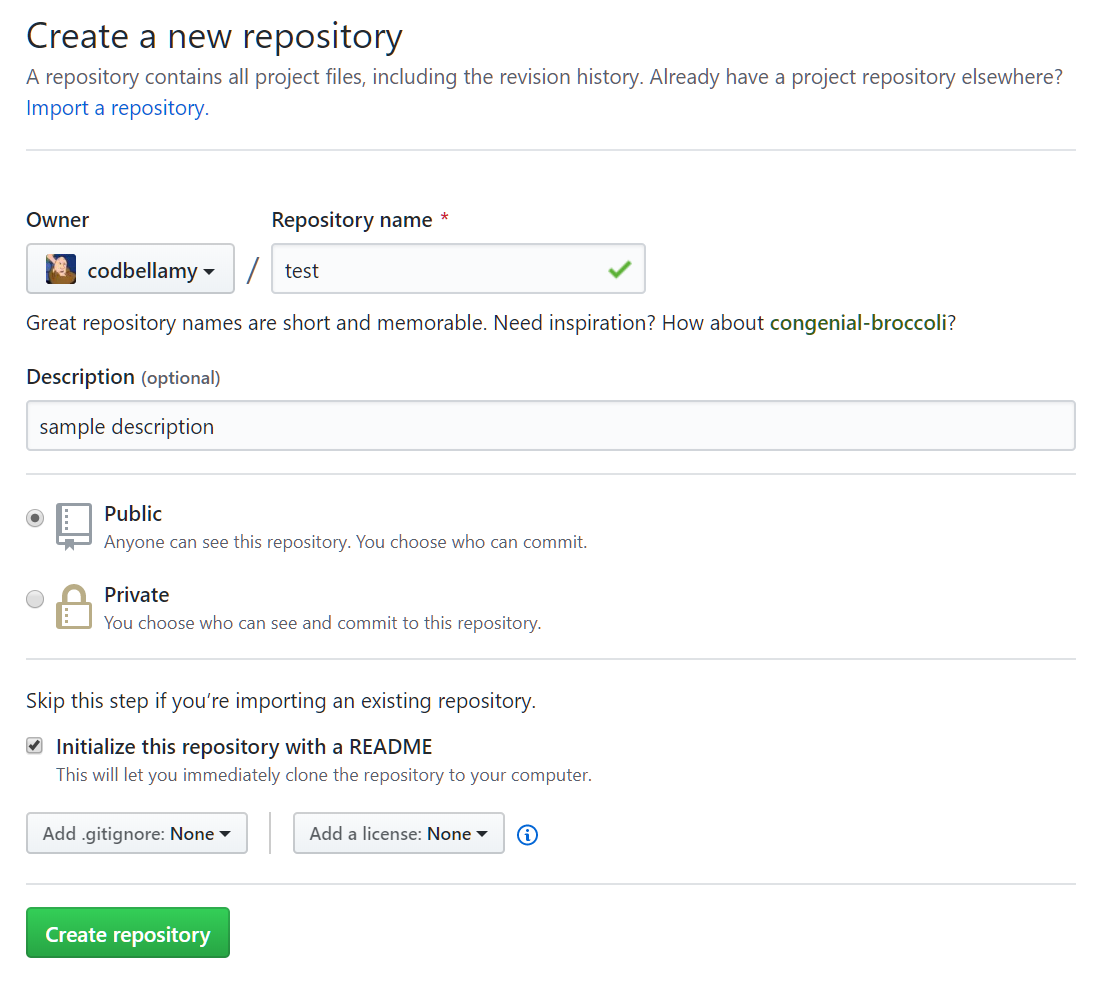
The following steps are to help you set up GitHub on your system. Although downloading the GitHub software is not necessary, it is highly recommended. Prior to completing any of these steps, it is mandatory for project collaborators to maintain an account. Standard users that will be cloning the project on their filesystems are not required to create an account. To create an account, go to <https://github.com/>.

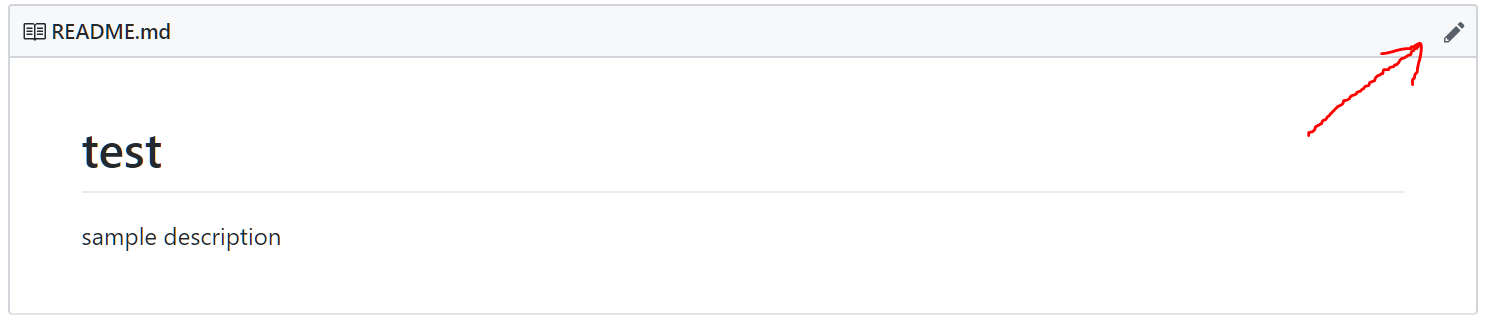
### Software

There is only one program that makes accessing, updating, and cloning projects marginally easier. This software is called “GitHub Desktop.” The software can be downloaded at <https://desktop.github.com/>. After installing GitHub Desktop, sign into your account if you are a project collaborator. All your project permissions will be translated to the software automatically. More in-depth tutorials on how to use the software will be covered in later sections.

### Project Collaborators (Teachers/Administors)

Project collaborators have the highest authority when it comes to managing the project or course files. Like D2L, only teachers and TAs should be added as project collaborators.

1. Sign into your account
2. At the top left of the screen by the “Repositories” header, click “New.” 
3. Create a repository name, enter a basic description of the project, and select “Initialize this repository with a README” 
4. Select “Create repository”

That’s it! You’ve successfully created your repository in GitHub. The README file is a text file that typically contains information about the project/course. You can edit this by clicking the pencil icon on the top-right of the dedicated README box. 

### Students

Students are considered standard users and will not have write permissions. Users can read non-private projects, clone projects to their desktops, and fork the project into a project of their own. Students that have the software can additionally clone the projects directly to their filesystems. A tutorial on how to clone projects will be covered in a later section.