

Installing and Using GitHub Desktop

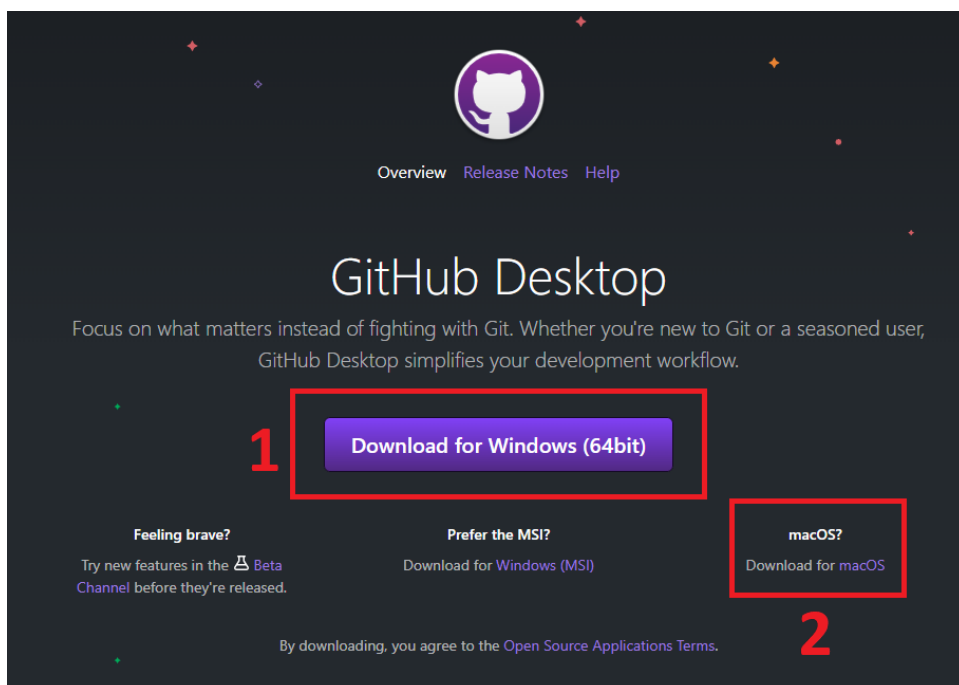
GitHub Desktop vs command line Git

GitHub desktop is an application with official support for Windows and MacOS, and community support for Linux. It can be used as an alternative to Git that is used in the command line, and will help you access repositories, editor and review changes to your code

Where to get GitHub Desktop

You can install GitHub Desktop either on Windows 7 64-bit or later and on macOS 10.12 or later. If you already have an GitHub account, you can connect it to your GitHub Desktop, from where you are able administrate your repositories and collaborate with others. For the download and installation of GitHub you can follow the next steps:

1. Visit the official [download page for GitHub Desktop](#).
2. Click on the button **Download for Windows (64-bit) (1)** if your operating system is Windows, or click on **macOS (2)** if you are using a macOS operating system.



3. Go to your computer's **Downloads** folder and double-click on the GitHub Desktop Setup file.
4. Follow the steps in the installation wizard. GitHub Desktop will launch automatically after the installation procedure is complete.

For Linux users, you can find community managed releases and instructions from ShiftKey: [shiftkey/desktop: Fork of GitHub Desktop to support various Linux distributions](#)

GitHub Desktop Accounts

GitHub Accounts

GitHub Desktop is directly integrated with GitHub, and users can sign in to their GitHub account from GitHub Desktop. This authentication procedure is completed in your computer's default browser.

With a GitHub account signed in to GitHub Desktop it is very easy to have access to the projects you are involved in, and clone repositories directly to your local machine, and start editing the code.

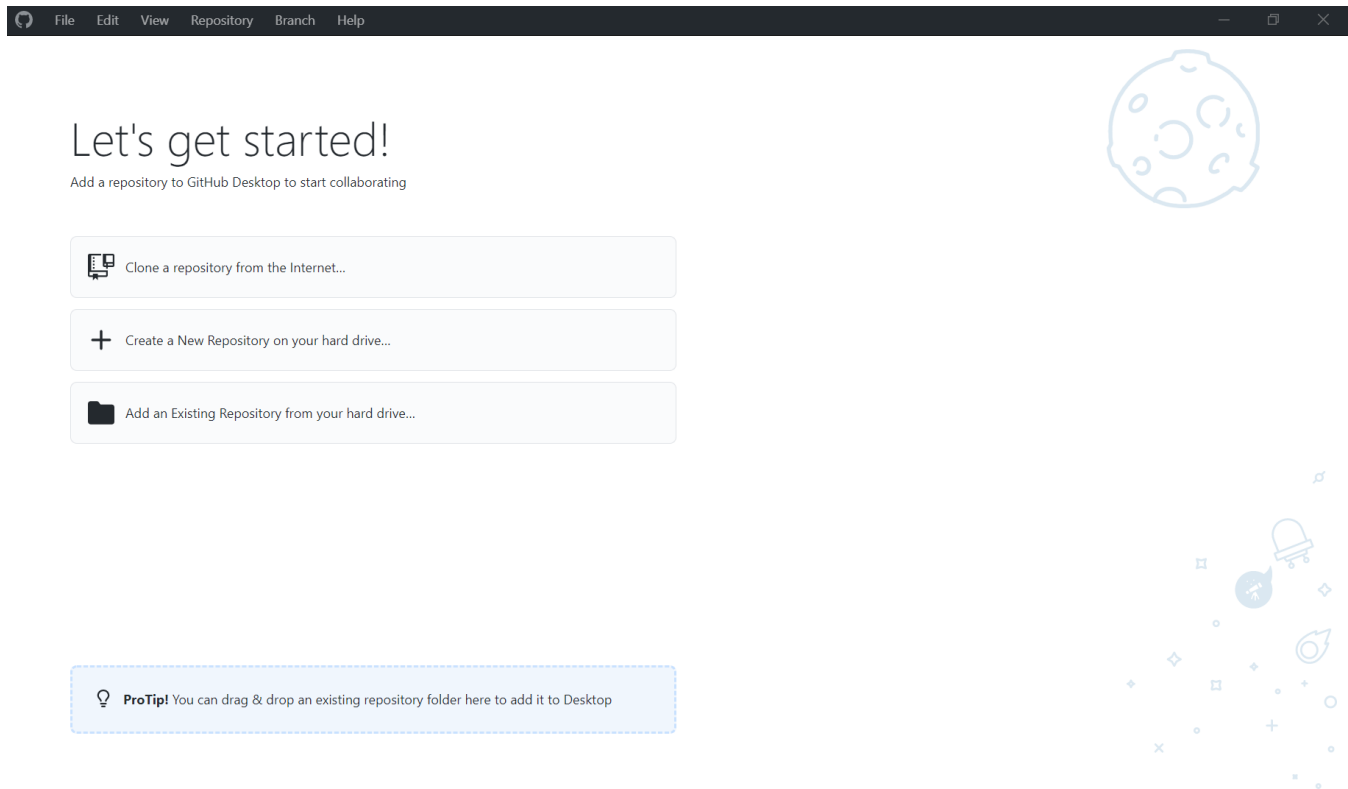
Other Accounts

For GitLab, BitBucket, Gitea, or other Git servers.

GitHub Desktop, despite the name, does not require the use of a GitHub repository, the software is Git compatible, and can be used with other Git servers, such as GitLab, BitBucket, or Gitea.

The GitHub Desktop Interface

After you have connected your GitHub account with GitHub Desktop you are ready to start using GitHub Desktop. In the following screen you can clone a repository from the internet, create a new repository on your computer or even add an existing one.



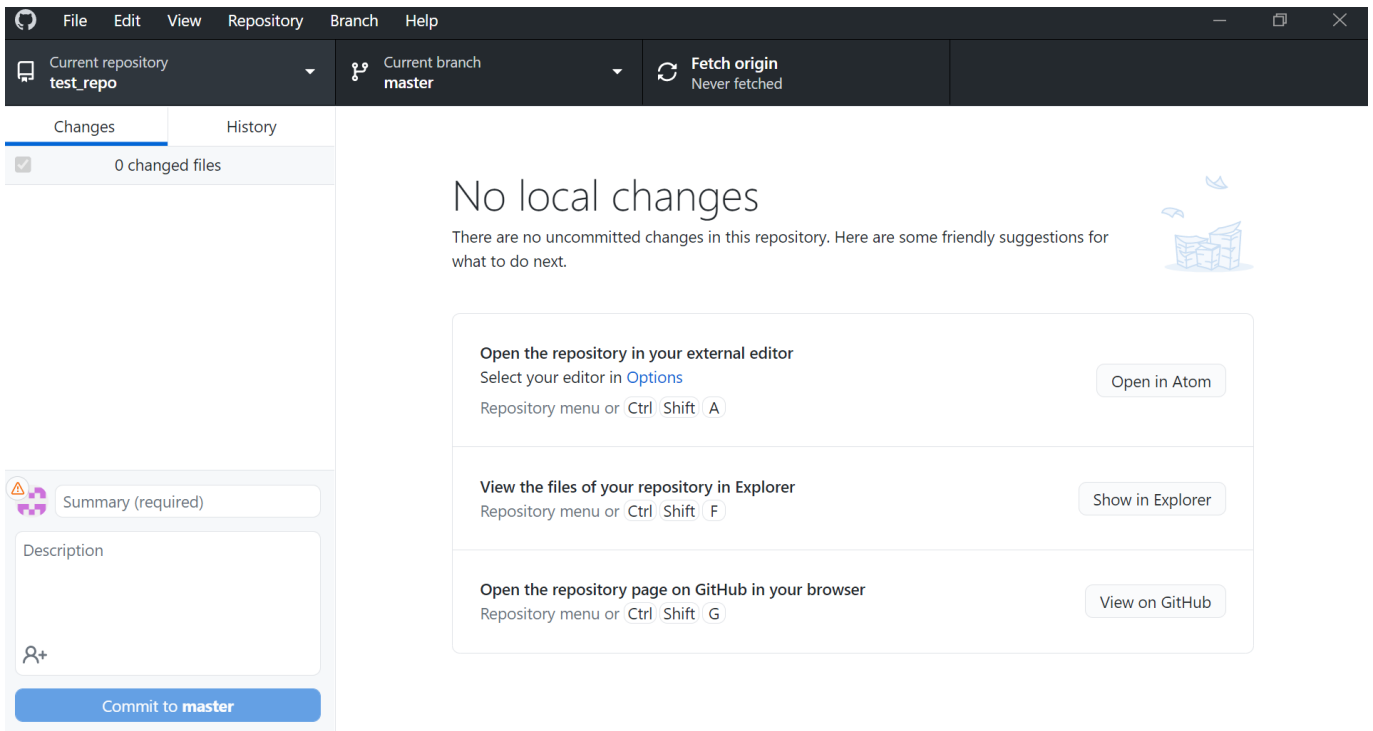
With the successful creation or cloning of a repository, you will be able to see the Repository view.

Repository View with No Code Changes

Below is what a repository will look like without any code changes.

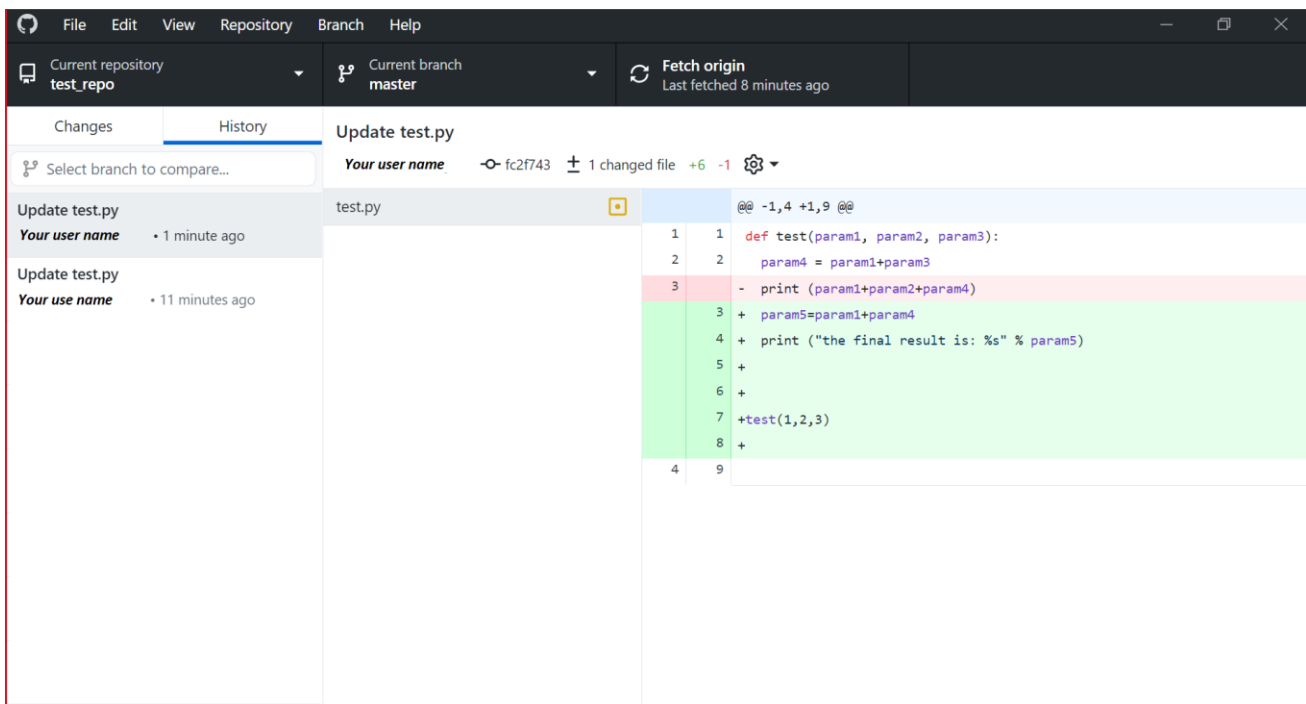
In the main area you have options to open the repository in your code editor, view the repository and its files on your filesystem, or to open the repository on GitHub.

Along the top bar under the menu, you can find the current repository you are working in, the branch of the repository, and a button to fetch origin. The fetch origin button is important, if multiple people are working on a code base or you are working on multiple computers, click this button every so often to get the most up to date version of the code from the repository.



Repository View with Code Changes

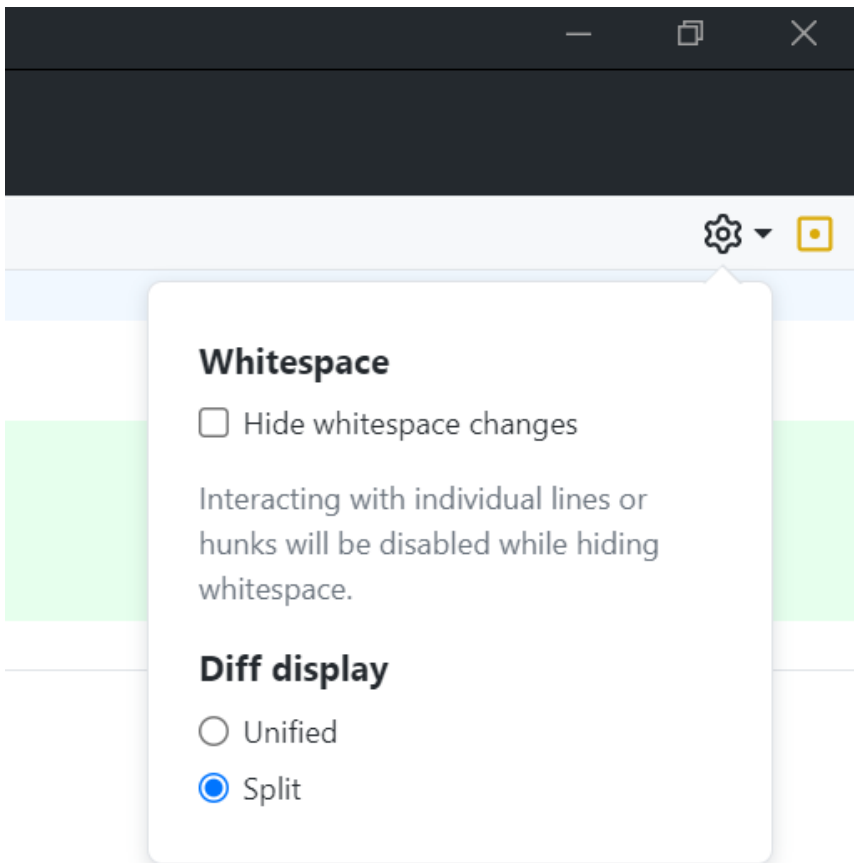
When you make changes to your code and save those changes to your local files, GitHub Desktop will convert to the repository view with changes, and that can be seen below.



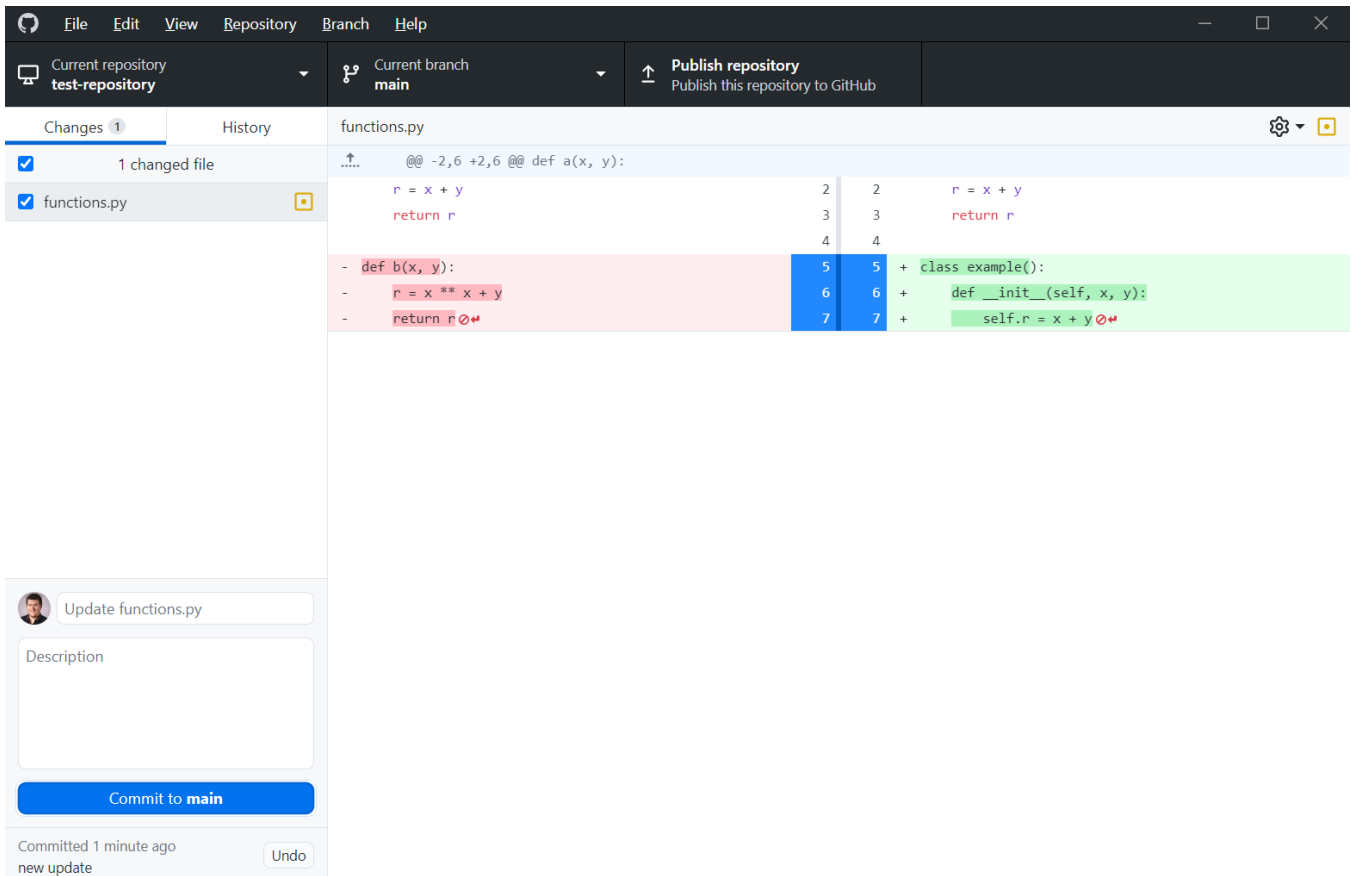
Adding Split View to Code Changes

When editing code it can be helpful to see the changes that are made in you code, and GitHub Desktop has a split view mode so you can see where the changes are line by line.

To change to split view, in the repository view with changes, click on the gear icon in the top left. Under Diff display, select Split.



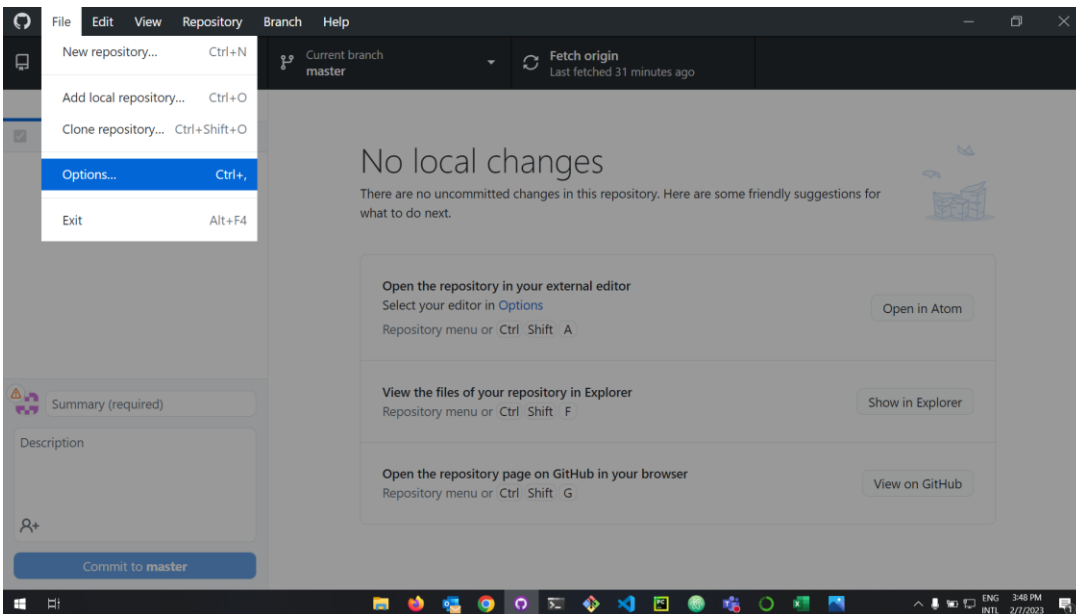
Then you should see the new changes in your code like in the screenshot below:



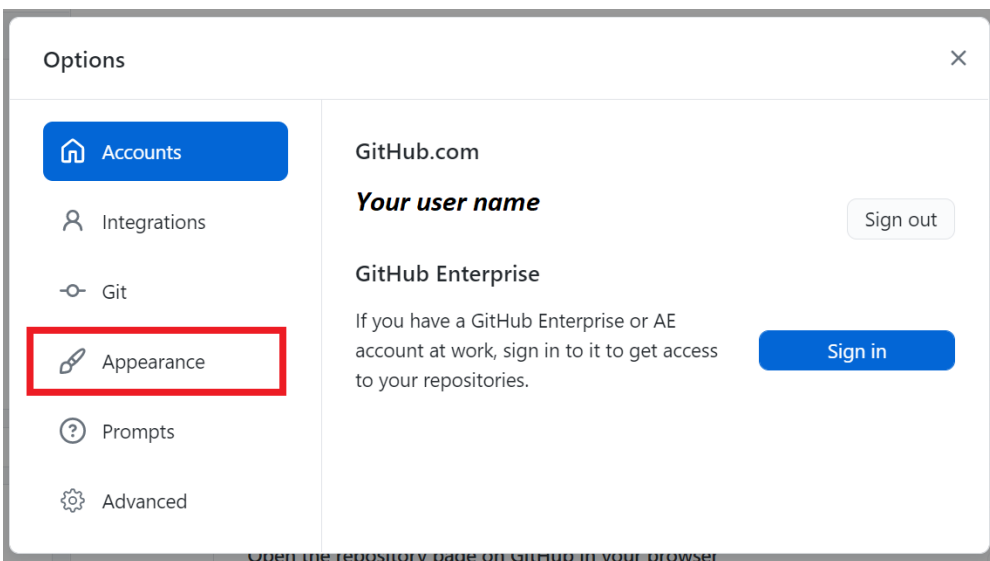
Dark Mode

If you wish, you can change the appearance of your GitHub Desktop interface according to your preference, with the following steps:

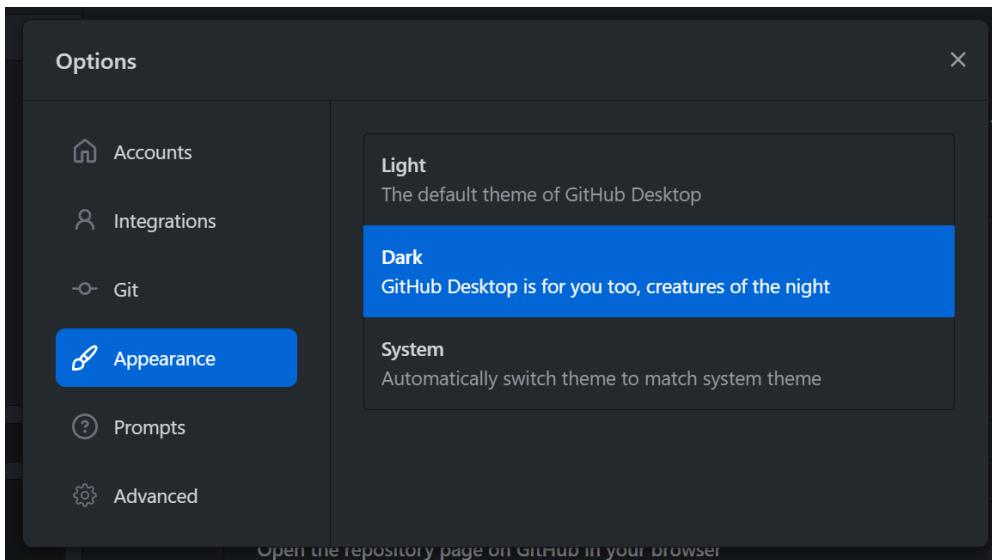
On the left side of the top bar of the interface, click on the **File** option. Then select **Options**.



The **Options** window will pop out. Click on the **Appearance** option.



In the **Appearance** view you can select **Dark**



This will change your GitHub Desktop theme to Dark Mode.

Changing the default Code Editor

From the repository page, there is the helpful “Open in {Code Editor}” button at the top, and GitHub can automatically open your repository in many different code editors.

For a list of supported code editors, see the [GitHub Desktop help page on Code Editors](#). We suggest VSCode/VSCodium, Sublime, JetBrains, or RStudio.

To change the default editor, follow these steps:

1. Go to File > Options
2. Click on Integrations on the left
3. Change the default editor with the External editor selection box. GitHub Desktop will display the available code editors currently installed on your machine.

