**GitHub Basics**

Contents

[Repositories 2](#_Toc73353904)

[Branching 2](#_Toc73353905)

[Commit 3](#_Toc73353906)

[Pull Requests 3](#_Toc73353907)

[Merge Pull Requests 4](#_Toc73353908)

[Conclusion 4](#_Toc73353909)

**GitHub** is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere.

Some of the GitHub essentials are **Repositories**, **Branches**, **Commits** and **Pull Requests**

## Repositories

A repository is usually used to organize a single project. It can contain folders, files, images, videos, spreadsheets, and data sets. – anything your project needs

*(Recommended a README text to provide information about your project)*

**To create a new repository:**

* In the upper right corner, next to your avatar or identicon, click and then select New repository.
* Name your repository hello-world.
* Write a short description.
* Select Initialize this repository with a README.

## Branching

Branching is the way to work on different versions of a repository at one time. By default, your repository has one branch named “**main”** which is considered to be the definitive branch. We use branches to experiment and make edits before committing them to “**main”**

When you create a branch off the main branch, you are making a copy or snapshot, of “main” as it was that point of time. If someone else made changes to the **“main”** branch while you were working on your branch, you could pull in those updates.

This diagram shows:

* The **“main”** branch called **“feature”**
* The journey that **“feature”** takes before it is merged into **“main”**

**Timeline

Description automatically generated**

Here at GitHub, our developers, writers, and designers use branches for keeping bug fixes and feature work separate from our **“main”** (production) branch. When a change is ready, they merge into **“main”**

**To create a new branch:**

* Go to your new repository hello-world.
* Click the drop down at the top of the file list that says branch: main.
* Type a branch name, readme-edits, into the new branch text box.
* Select the blue Create branch box or hit “Enter” on your keyboard.

## Commit

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.

**Make and commit changes**

* Click the README.md file.
* Click the pencil icon in the upper right corner of the file view to edit.
* In the editor, write a bit about yourself.
* Write a commit message that describes your changes.
* Click Commit changes button.

These changes will be made to just the README file on your readme-edits branch, so now this branch contains content that’s different from **“main”**.

## Pull Requests

Now that you have changes in a branch off of main, you can open a pull request.

Pull Requests are the heart of collaboration on GitHub. When you open a pull request, you’re proposing your changes and requesting that someone review and pull in your contribution and merge them into their branch. Pull requests show diffs, or differences, of the content from both branches. The changes, additions, and subtractions are shown in green and red.

As soon as you make a commit, you can open a pull request and start a discussion, even before the code is finished.

By using GitHub’s @mention system in your pull request message, you can ask for feedback from specific people or teams, whether they’re down the hall or 10 time zones away.

You can even open pull requests in your own repository and merge them yourself. It’s a great way to learn the GitHub flow before working on larger projects.

## Merge Pull Requests

In this final step, it’s time to bring your changes together – merging your readme-edits branch into the main branch.

* Click the green Merge pull request button to merge the changes into main.
* Click Confirm merge.
* Go ahead and delete the branch, since its changes have been incorporated, with the Delete branch button in the purple box.

## Conclusion

Here’s what you have accomplished:

* Created an open-source repository
* Started and managed a new branch
* Changed a file and committed those changes to GitHub
* Opened and merged a Pull Request