

Suppose you want to change someone else's project or use someone else's project as a starting point for your own idea. If you are not a collaborator, you cannot modify files in the repo of the project.

In this case, GitHub makes it possible for you to become a contributor. You need to **fork** this repo, modify its content, and propose your changes via **pull request**. If the owners of the project like your changes, they might put them into the original repository.

## **Forking a repo**

A **fork** is a copy of a repository. It allows you to freely experiment with it without affecting the original project. To make a fork, you must log into GitHub.


Please, open [our educational repository](#) and click **Fork** in the top-right corner of the page. You will be prompted to modify the repository name and the description when creating the fork. Once you are done, go ahead and click **Create fork**.

## Create a new fork

A *fork* is a copy of a repository. Forking a repository allows you to freely experiment with changes without affecting the original project. [View existing forks.](#)

Required fields are marked with an asterisk (\*).

Owner \*      Repository name \*

 the-nulldev ▾ / intro-to-github

✔ intro-to-github is available.

By default, forks are named the same as their upstream repository. You can customize the name to distinguish it further.

Description (optional)


This is a demo repository to practice using GitHub.

☒ Copy the `master` branch only  
Contribute back to hyperskill/intro-to-github by adding your own branch. [Learn more.](#)


 You are creating a fork in your personal account.

Create fork

You will be automatically redirected to your fork of the original repository. You can also find it on the **Your repositories** page (click on your profile in the top-right corner of the page).

Find a repository...    Type ▾    Language ▾    Sort ▾     New

---

**intro-to-github** Public     Star ▾

Forked from hyperskill/intro-to-github

This is a demo repository to practice using GitHub.

---

Updated 4 hours ago

A fork is just a regular repository hosted on GitHub like the original one. You can add new files and modify them or remove existing ones in your fork.

## Branching

Before we start making changes to the content of the repo, we need to learn an important concept called **branching**. It provides a way to work on different versions of a repository at the same time.

By default, any repository has a branch called **main** that represents the actual version of a project. If you want to change some content in a repo, first, you need to create a new branch based on **main** to get an isolated snapshot of the project. Usually, a branch represents a feature or a fix that is being developed at the moment. It allows you to protect **main** from possible unwanted changes and to make a rollback. You can also create a new branch based on another branch (non-main) that is active.

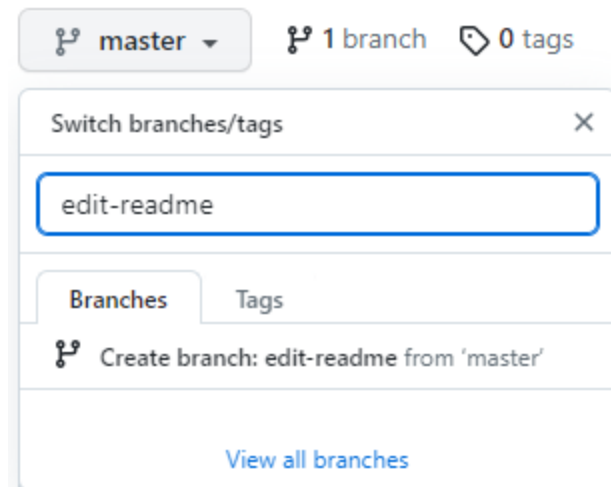
**Branching** especially helps organize team development processes. Several programmers work on different features at the same time, each in their own branch. After the job is done, the code is merged together.

Now it may not be easy for you to understand branching, so just follow our recommendation and use special branches rather than work on the **main** directly. If you want to read more about branches, check out [a short article https://docs.github.com/en/get-started/using-github/github-flow](https://docs.github.com/en/get-started/using-github/github-flow)

written by GitHub.

## Modifying content in a special branch

Before we start modifying the content, let's create a new branch. Click on the **main** branch, this will open a drop-down menu. Enter a unique descriptive name for your branch like **edit-readme** and then press **Enter** or click **Create branch: name**.



Let's modify the **README.md** file, which we can find in the new branch. Open this file and click the edit button (pencil) or click the same button below the repo. Make some changes; for example, write some information you learned about forks or branches.

To apply these changes, scroll to the **Commit changes** section below, write a meaningful message about your changes, and then click **Commit changes**. It will put these changes in your repository in the **edit-readme** branch. The original repo and the **main** branch of your fork will not undergo any changes.

Commit changes

Commit message

Update README.md

Extended description

Add an optional extended description..

☒ Commit directly to the master branch

☐ Create a new branch for this commit and start a pull request

[Learn more about pull requests](#)

Cancel Commit changes

Then check your changes in the **README.md** file.

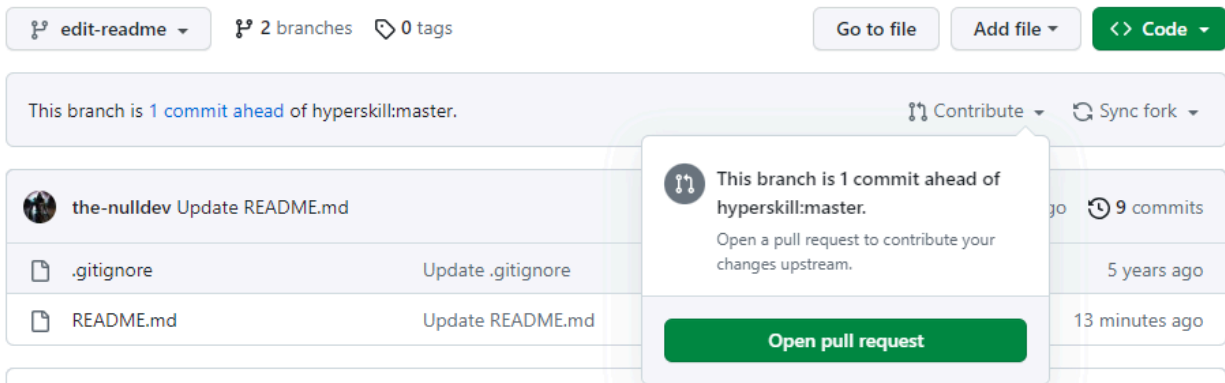
You may perform several commits in the same branch with different changes to better understand this process. You can also create or upload a new file in this repo if you want.

Further on we will learn how to modify the content of some repo using IntelliJ IDEA and Terminal instead of a web browser.

## Pull request

After you change the content of a repo, you can create a **pull request (PR)** to propose your changes to the original repository. Ensure that you are in the new branch where you made your changes. There will be a message showing you

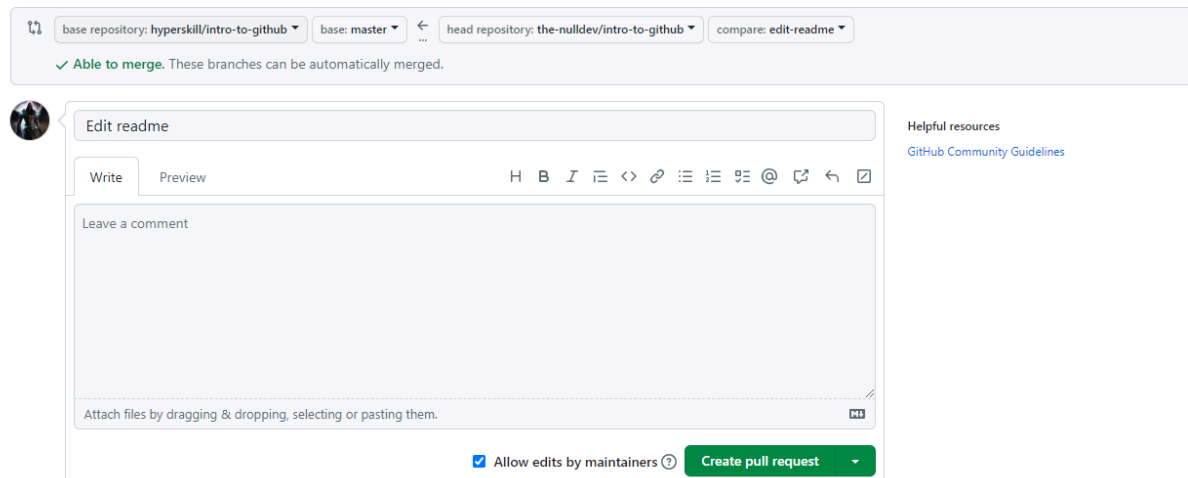
the number of commits that have not been merged to the main branch. Click the **Contribute** dropdown and then select **Open pull request**. There is also a **Sync fork** button that allows you to fetch changes from the main repository that you created a fork from.



Then you will see a special page for creating a **pull request**.

### Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



You can see the green text **Able to merge** which means that your version of the project does not conflict with the original version in the repo. In some cases, the text might be different; that would mean that the original repository has some recent changes, but you still can create a PR. Later we will learn how to synchronize your fork with the original repository.

Make sure that you've specified the base fork as the original repo (**hyperskill/intro-to-github**) with the **main** branch, and the head fork is your repo with a non-main branch.

After providing a title and optionally, a comment, click **Create pull request**. You will see the PR that you've created in the repo of this project in the **Pull requests** section. Your pull request will be reviewed, and if the changes you proposed are accepted, they will be merged with the main branch. The status of the PR will then change to **closed**. You can also close the pull request before it is accepted in case you change your mind after opening a pull request. To avoid having too many branches in your repository, you can delete this branch after the merge.

We will not respond to your requests in this repository, don't be surprised :)

Finally

,

you can also create a completely new repository for a new project without forking. To do this, go to your repository and click New. Enter a descriptive name for your repository. You will have full access to this repository as its owner, so you do not need to fork it to make changes. If you want, you can practice with your own repository.

## Conclusion

Let's quickly recap what we did here in general. First, we copied the repo, in other words, we created a fork of that repo. Then in our fork, we created a new branch. Next, we made changes to the file inside our new branch. Finally, we offered our modifications to the original repository by making a pull request of our fork with the new branch.

Suppose, you have found an interesting project on GitHub. You are not a collaborator on this project.

You forked this repo. Which operations are available for you?

Select one or more options from the list

- ☒ Create an issue in fork.
- ☒ Add new files in fork.
- ☒ Create a pull request to propose changes in the original repo.
- ☒ Change a file in fork (e.g. README.md).

Continue

---