**GitHub** is a code hosting platform. It is a place where developers can store their projects and work together to develop new projects. It's easier to control program versions and collaborate this way. **GitHub** is based on a popular version control system called **Git**, and provides some additional features such as a web interface, collaboration tools, an issue tracker, project statistics, and more.

Note that Github is often used as a portfolio. You can publish your projects there and send a link to your Github profile to employers. Having projects in your Github profile will increase your chance of getting a job.

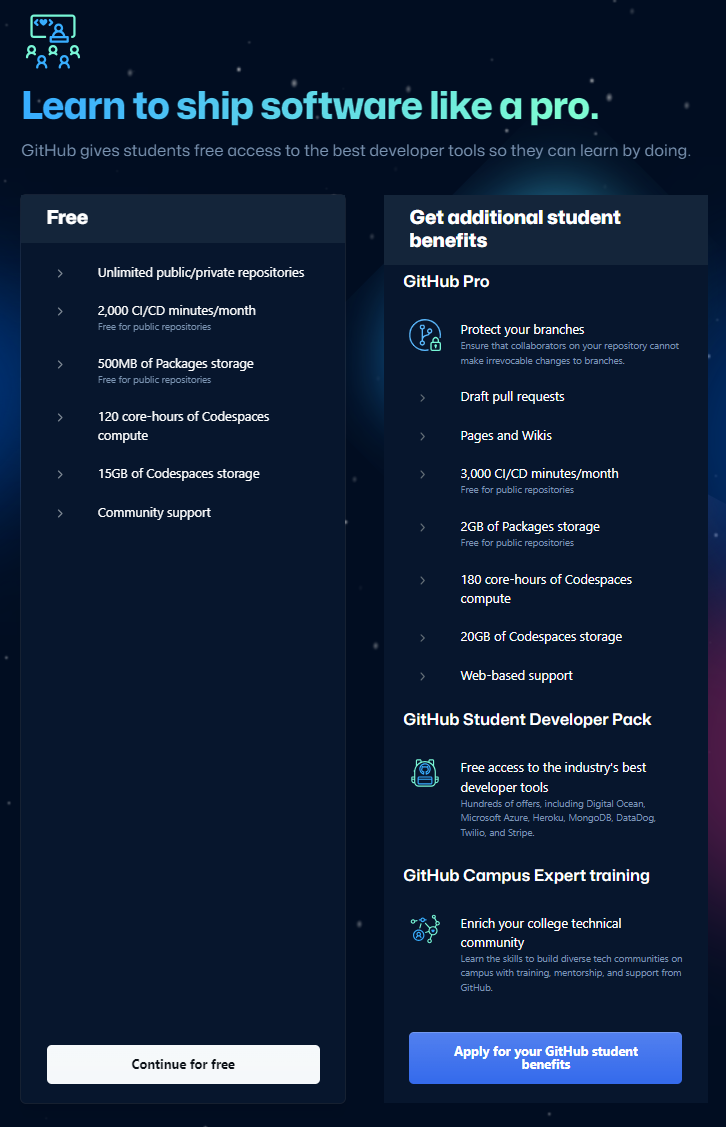
As of 2020, **GitHub** is the largest host of source code in the world. There are many popular open-source projects hosted on GitHub that you can contribute to.

In this lesson, you will learn how to start using GitHub through your web browser. We recommend you to repeat everything that we do here step by step.

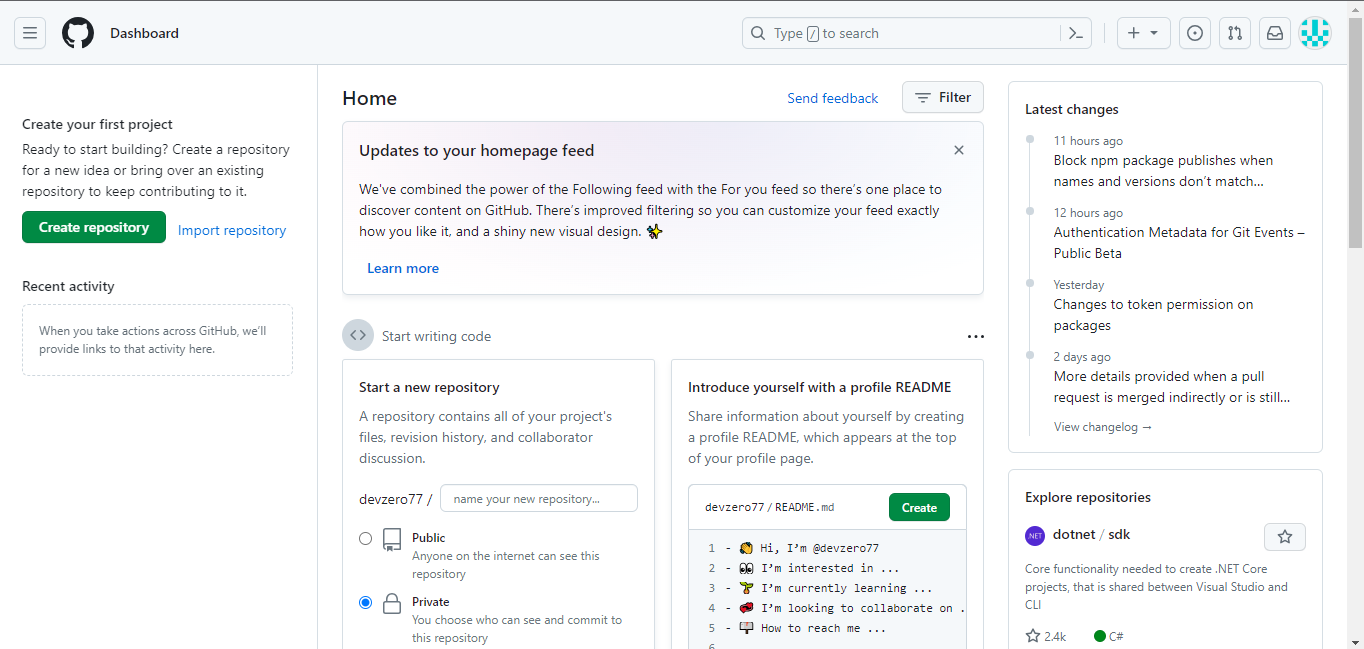
**Creating an account on GitHub**

To complete this lesson, first, you need to create a free account on [GitHub](https://github.com/) (or use an existing one). The profile is your public page on GitHub just like your social network profile. When you're looking for a job as a programmer, potential employers might check your GitHub profile and take it into account as they consider you for the job.

When you create an account, in the last step choose "Continue for free":



After you complete the registration, you will see a dashboard. You can choose to create a new repository or click on your profile icon at the top right corner to explore settings and other GitHub features. The dashboard looks like the one in the screenshot below:



You can open **your profile** and make changes (for example, you can add a bio or upload a picture).

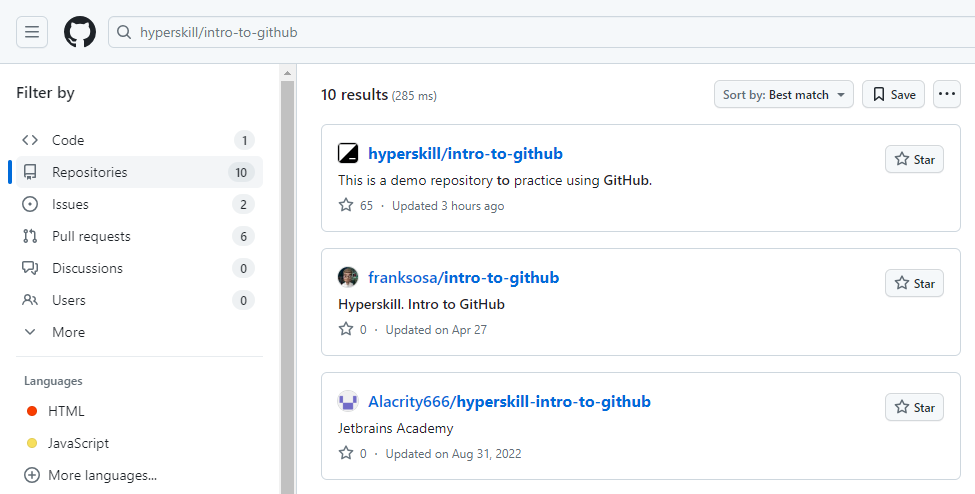
You can also use the exercises provided by the GitHub Bot to get started. Just look for the [Getting started](https://docs.github.com/get-started/quickstart/hello-world) exercise on the dashboard.

**Exploring a repository**

**GitHub** hosts millions of projects written in different programming languages. Each project is placed in its own container called a **repository** **(repo)** that can store code, configurations, datasets, pictures, and other files included in your project. Any changes to the files within a repo will be tracked via version control.

Just after you have gotten your account you do not have any repositories. In order to create one, click on your profile icon in the top right corner and select *Your repositories.* You will be redirected to your repositories tab. Here, you can create a new repository by clicking the *New* button

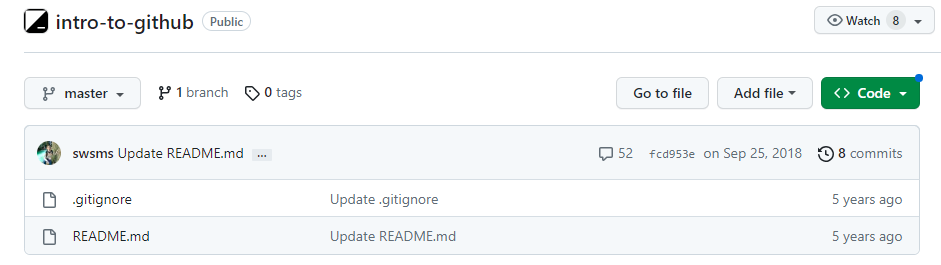
If you want to find some specific repository of a project, type its name or part of the name in the search box. You will see a list of suitable repositories.



Sometimes, the name of a repository might be too long or there may be lots of repositories with the same name. For example, try searching for **intro-to-github** without **hyperskill/**.

Usually, you have a link to a repository. Many open-source projects publish links on their personal websites, or your colleagues may give you a link to a repository of a project.

Here is the direct link to our existing [demo repository](https://github.com/hyperskill/intro-to-github/). Go ahead and open this repo and take a look at its internal structure.



From October 1, 2020, [the master branch on Github has been renamed to main](https://github.com/github/renaming). Now, in all repositories that will be created, the main branch will be called main, the old repositories created before October this year remain unchanged.

There are two files in the **Code** tab. **README.md** is a file that describes the project; every repository should have this file. GitHub finds this file and displays its content below the repo. The other file is **.gitignore**; it specifies which files and directories Git should ignore.

Although this repo is public, you cannot modify files in this repo directly, because you have read-only access to any public repository. To modify the files, you must be a **collaborator** on this project.

The **Pull requests** tab contains proposals for changes in the files that are in the repository (adding, deleting, or modifying files). The owners of the repo can review a request and approve your changes if they are good enough.

You can create an **Issue** or make a **Pull request (PR)** to contribute to the project, even if you are not a collaborator.

In our project, the **Wiki** tab is empty, but it is actually where the documentation of the project (how to use it, how it was designed, its core principles, and so on) should be stored. Whereas a **README.md** is to quickly let readers know what your project can do, wikis help provide additional information.

We don't discuss the **Projects** tab in this lesson.

In the **Insight** tab you can find stats and information about the repository.

You may also look at the**commit(s)** which represent changes in the content of the repo. Here is a [link](https://github.com/hyperskill/intro-to-github/commits/master) to see it directly. We will learn more about commits further on.

**Conclusion**

**GitHub** is a web platform where people can store their projects as Git-repos. Using their web interface, you can create an account, a repo and search for repos of other projects. Every repo on the platform has several tabs inside, such as **Code** with **README.md** file, **Pull requests** where you can offer changes in the files, the **Wiki** tab with project documentation and **Insights**with statistics about the repository.

Read more on this topic in [Bridging the Gap by Understanding DevOps](https://hyperskill.org/blog/post/bridging-the-gap-between-development-and-operations-by-understanding-devops) on Hyperskill Blog.

What is the name of the file in which directories and files that should **not** be pushed to the repository are written?

 .gitignore

From October 1, 2020, the master branch on GitHub has been renamed. What is the new name of this branch? main

**Operations with a project**

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

Which operations are available for you?

Correct options based on your non-collaborator status on GitHub:

✅ **Open files within the code section.**  
✅ **Create an issue.**

❌ **Change an existing file in the code section.**  
❌ **Add new files in this repo.**

As a non-collaborator, you can **view the code** and **create issues**, but you **cannot directly modify or add files** unless you fork the repository and create a pull request.

Enter a link to any issue of any repository on GitHub.

1. go to your repo in github

2. go to issues tab

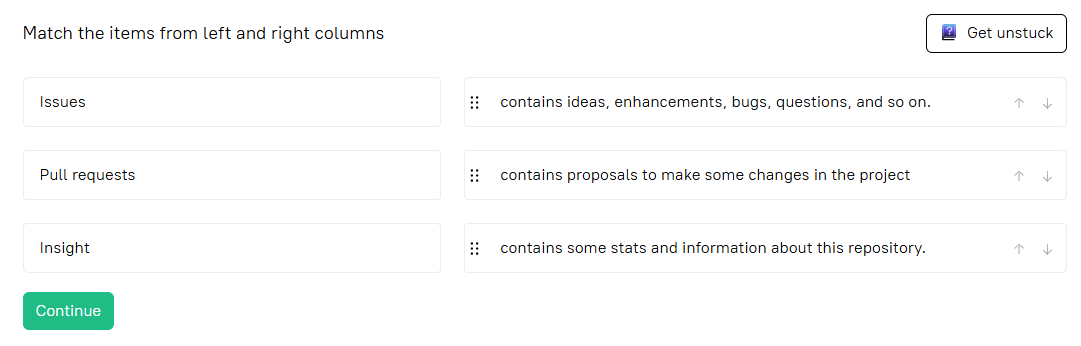
3. create / new issue, save

4. copy the link to the issue

5. done

<https://github.com/JFallouh/CodeCrafterHub/issues/1>

Match tabs from GitHub UI and their meanings.



**Explore a project repository**

Imagine you have found some project repository on GitHub. Where can you see information about the project to understand what it is about?

Correct options for finding information about the project:

✅ **in Wiki tab**  
✅ **in README.md file**

❌ **in .gitignore file**

❌ **in Issues tab**

### Explanation:

* **README.md**: Usually contains the main description, purpose, installation, usage, etc.
* **Wiki tab**: May contain detailed documentation or guides if maintained.
* **Issues tab**: Can give insights into current problems, feature requests, and project activity.
* While the **Issues tab** can give some context about what’s being worked on or problems faced, it’s not meant for the primary project description. Thanks for the clarification!
* **.gitignore**: Only lists files/directories to be ignored by Git; not useful for understanding the project.

A GitHub repository is

a container for project and its data

**Make a change**

Create a fork of our [repository](https://github.com/hyperskill/intro-to-github), make some changes there and create a pull request. Give a link to your PR as the answer to this task.  
Once you are done - please close the PR and delete the branch.