Dharmsinh Desai University

MCA Sem 2

Data structure using C

Report

Name: Mansi Nukani

Roll no: MA012

Student id: 18MAPOG005

Submitted to

Prof.Himansu Purohit

What is GitHub?

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

What is Git?

- Git is a distributed revision control and source code management system with an emphasis on speed. Git was initially designed and developed by Linus Torvalds for Linux kernel development.

❖ **Version Control System (VCS)** is a software that helps software developers to work together and maintain a complete history of their work.

- Listed below are the functions of a VCS −
  - Allows developers to work simultaneously.
  - Does not allow overwriting each other's changes.
  - Maintains a history of every version.
- Following are the types of VCS −
  - Centralized version control system (CVCS).
  - Distributed/Decentralized version control system (DVCS).

❖ **Distributed Version Control System**

- DVCS clients not only check out the latest snapshot of the directory but they also fully mirror the repository. If the server goes down, then the repository from any client can be copied back to the server to restore it.
- Every checkout is a full backup of the repository. Git does not rely on the central server and that is why you can perform many operations when you are offline. You can commit changes, create branches, view logs, and perform other operations when you are offline.
- You require network connection only to publish your changes and take the latest changes.

What is the use of GitHub?

- GitHub is an online; browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more.

- There are also social networking elements of the service including feeds, the ability to follow other users, and network related graphs. This article will walk you through the process of signing up for GitHub.

Advantages of Git

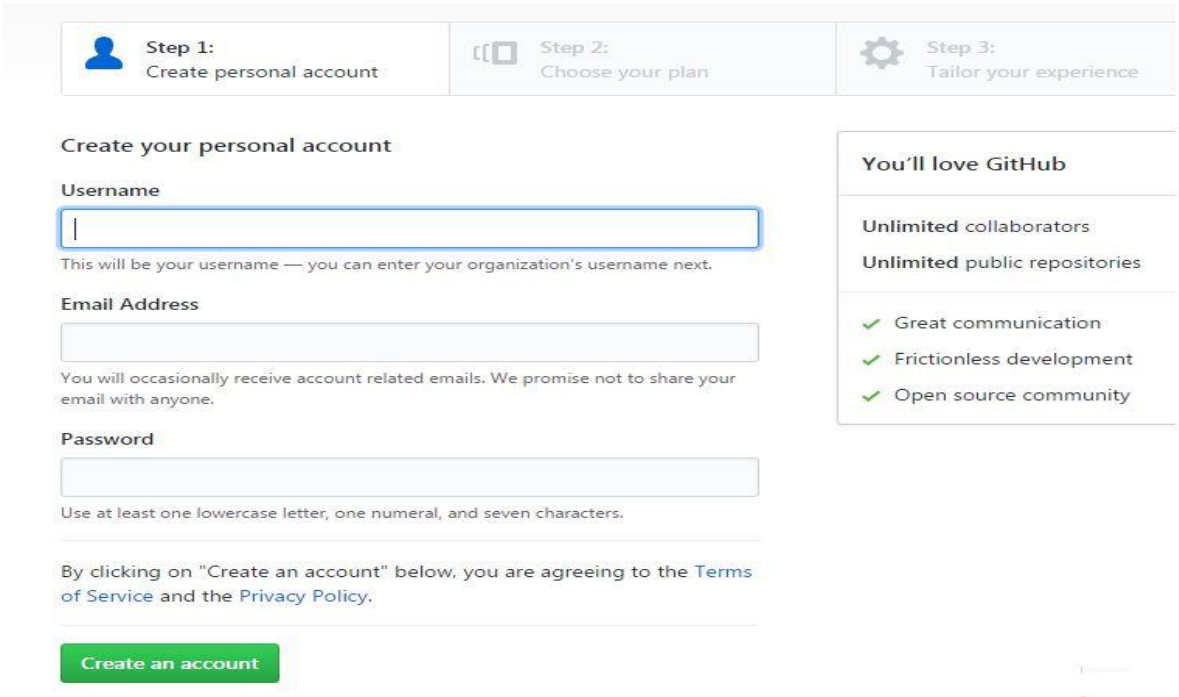
- Free and open source
- Fast and small
- Implicit backup
- Security
- No need of powerful hardware
- Easier branching

Disadvantages

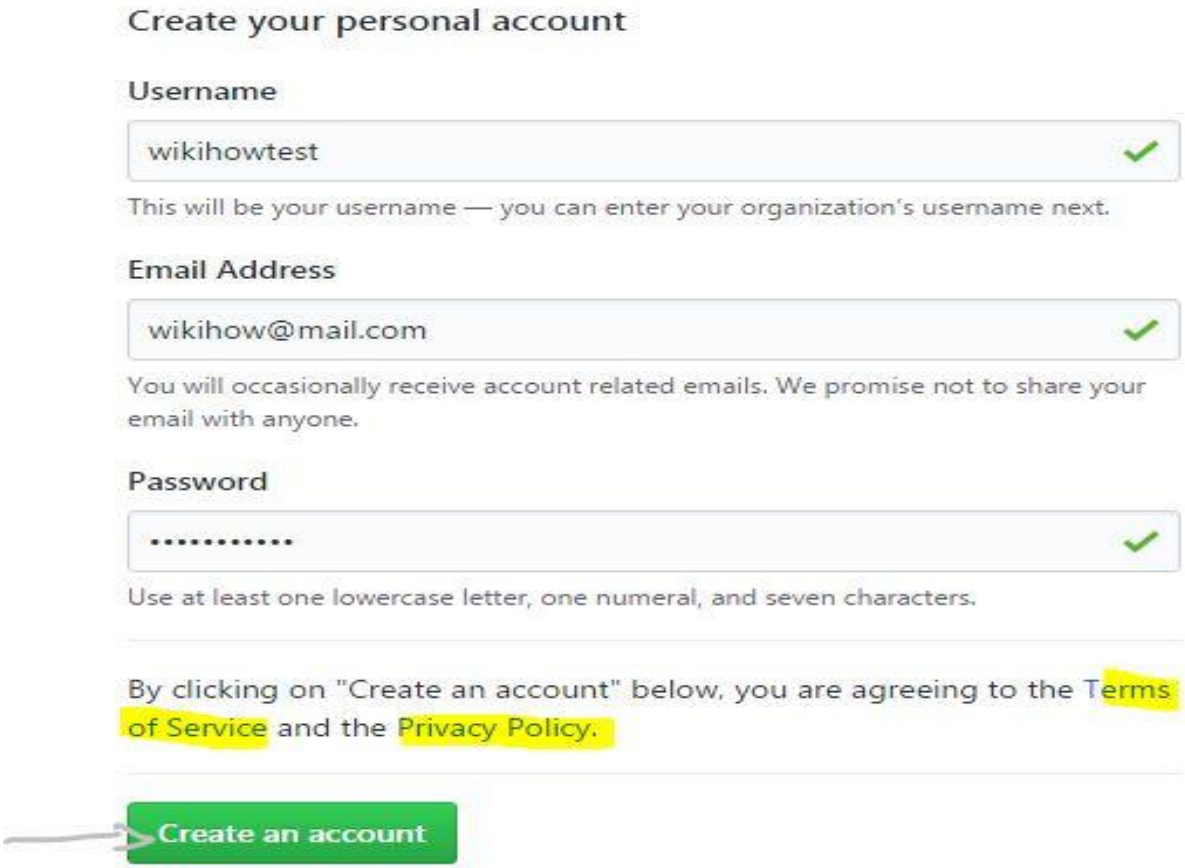
- Steep learning curve: Many commands with many options, some commands are non-intuitive and need a level of understanding the internals of git, commands and arguments are inconsistent to some degree
- Binary files are a big no: If your project has non-text files that are updated frequently (images for websites or MS Office documents), then git becomes bloated and slow. (I believe it still does better than most systems).

Steps to create account

1. go to the github sign up page

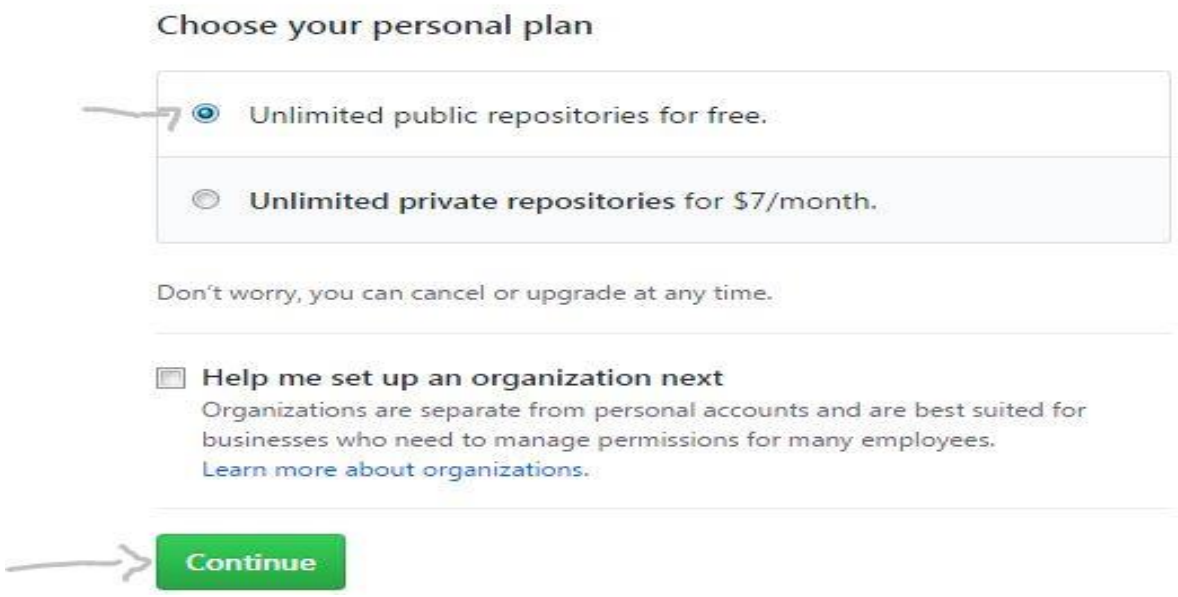


2. Enter a username, valid email address, and password.Use at least one lowercase letter, one numeral, and seven characters.

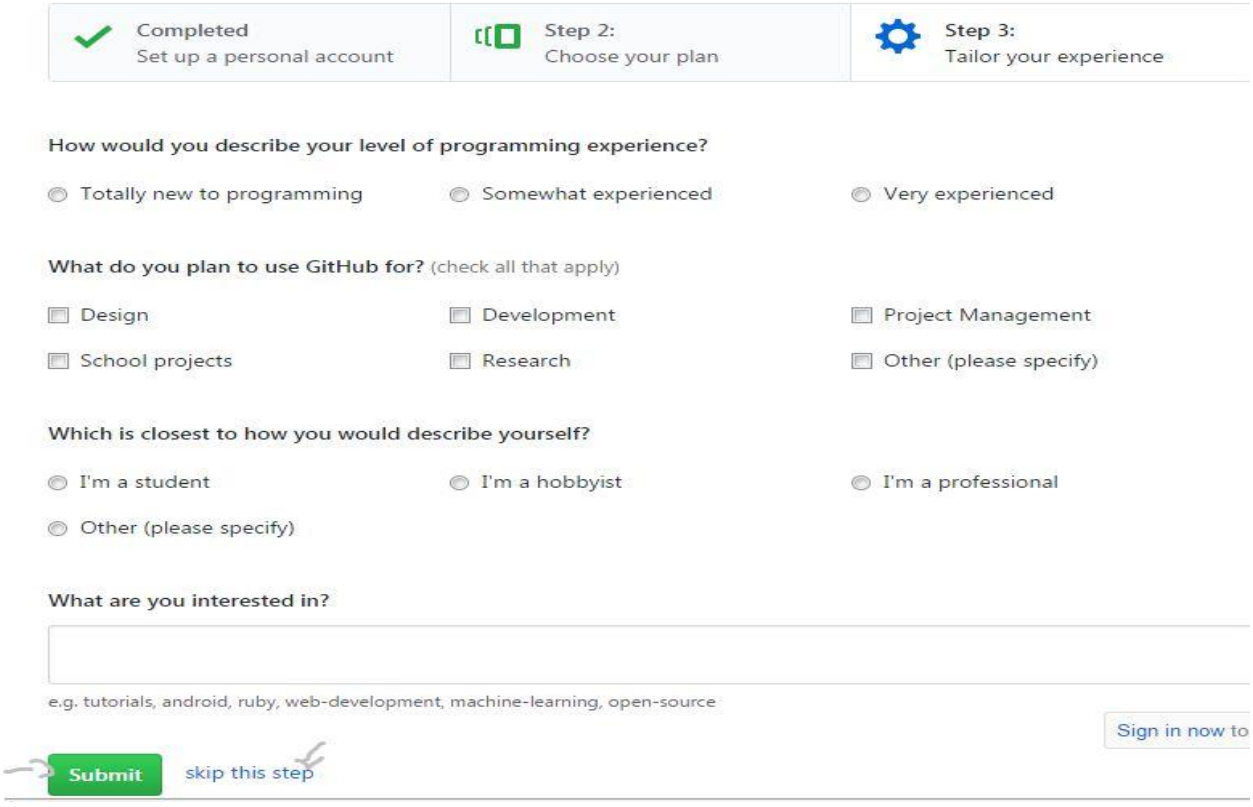


3 **Review carefully the GitHub Terms of Service and Privacy Policy before continuing.** Upon clicking the "Create an account" button you will simultaneously be agreeing to these documents.

4 **Choose a plan.** You have two choice: Free and paid, the paid version has private repositories with $7/month. You should try the free version then have the suitable choice.



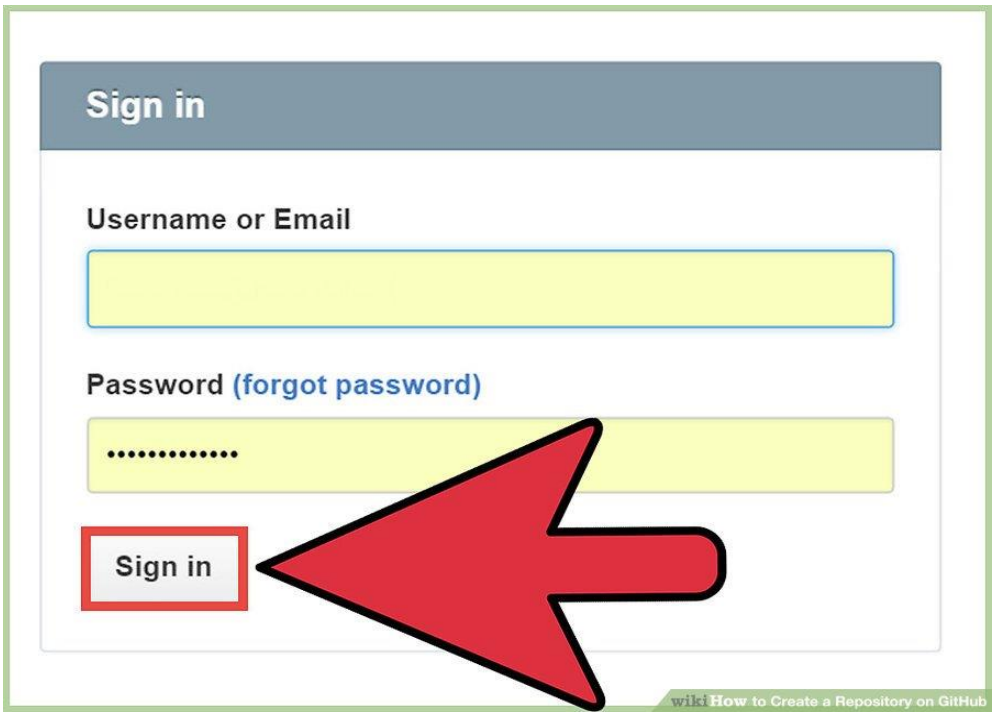
5 **Tailor experience.** If you have time, fill in the survey or skip it



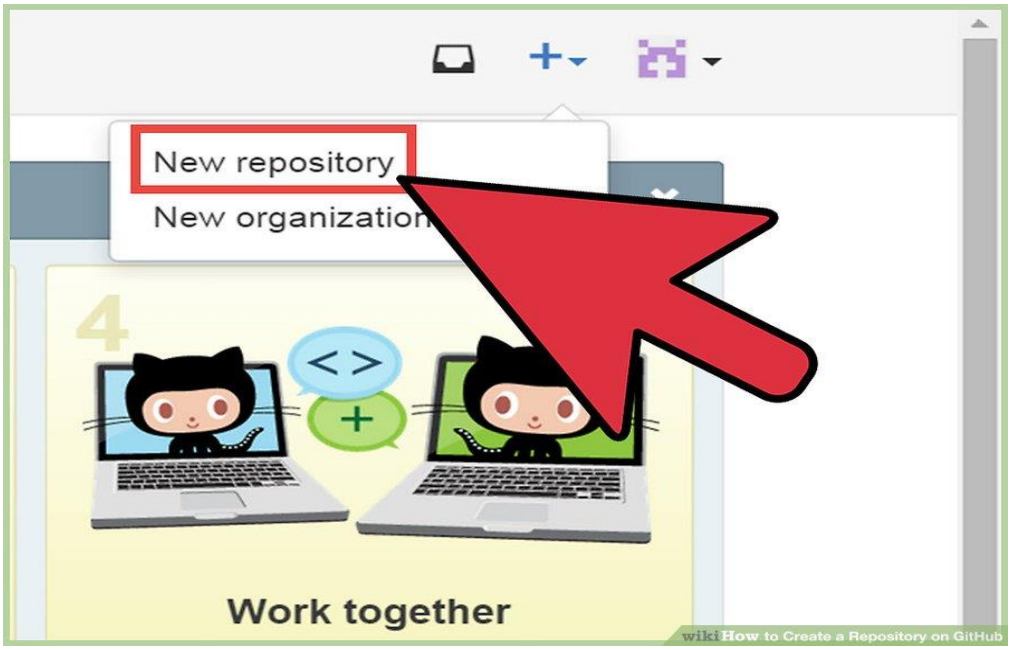
6 **You finished!** Your Github account created!

How to create repository

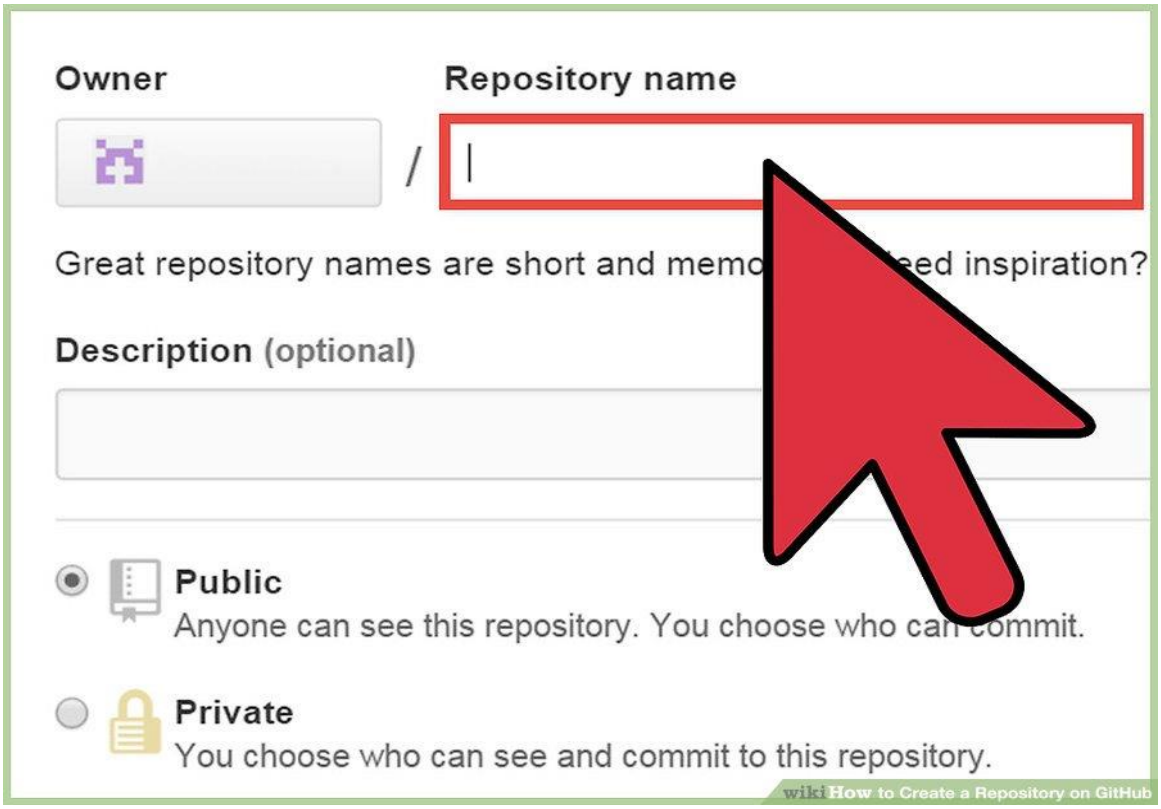
1. Sign up for a GitHub account, if you don't already have one.
2. Go to the GitHub login page and log in using the email and password associated with your account.



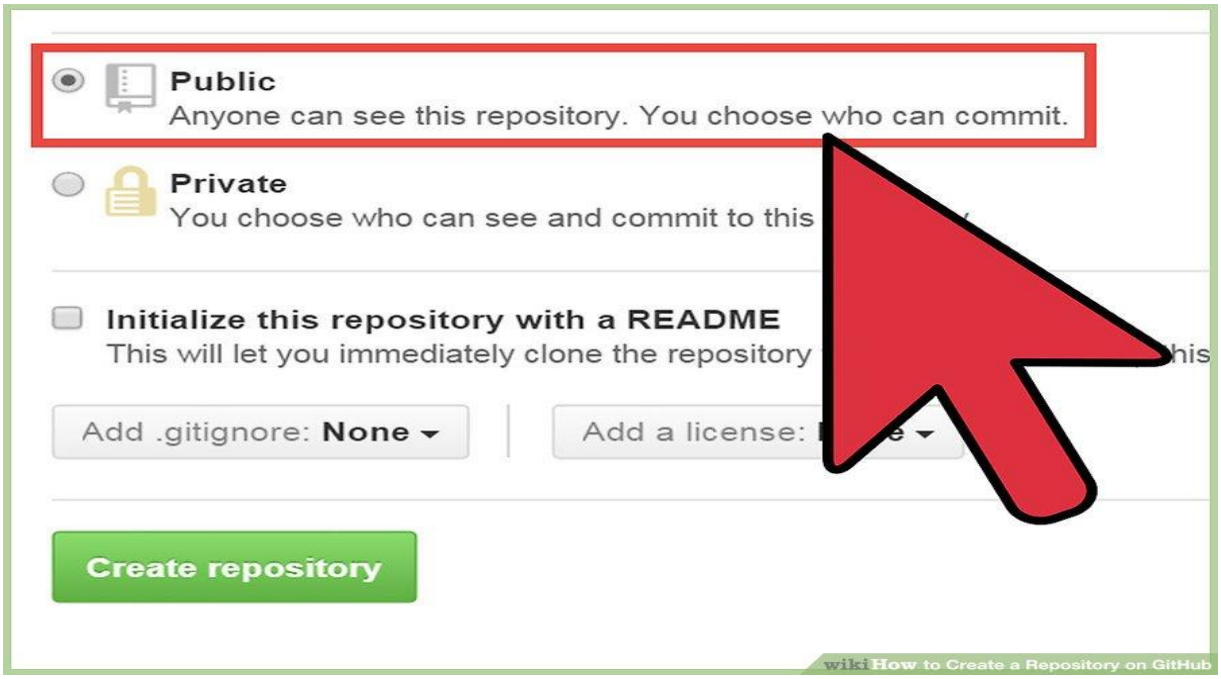
3. Click the "New Repository" button in the "Your Repositories" pane on right side of the page.



4. **Enter a project name, and an optional description and homepage URL.**



5. **Select the "Public" option to create a free public repo that anyone can access, or select the "Private" option to upgrade to a paid account and create a private repository.**



6. Click the "Create Repository" button to create your repo.

Adding file to repository

- You can upload and commit an existing file to a GitHub repository. Drag and drop a file to any directory in the file tree, or upload files from the repository's main page.

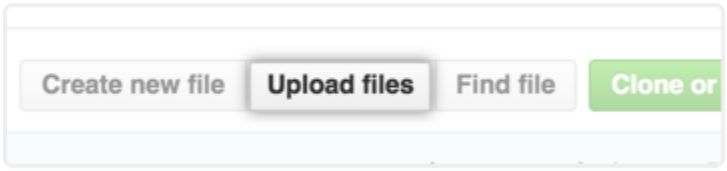
- Files that you add to a repository via a browser are limited to 25 MB per file. You can add larger files, up to 100 MB each, via the command line. For more information, see "Adding a file to a repository using the command line."

Tips:

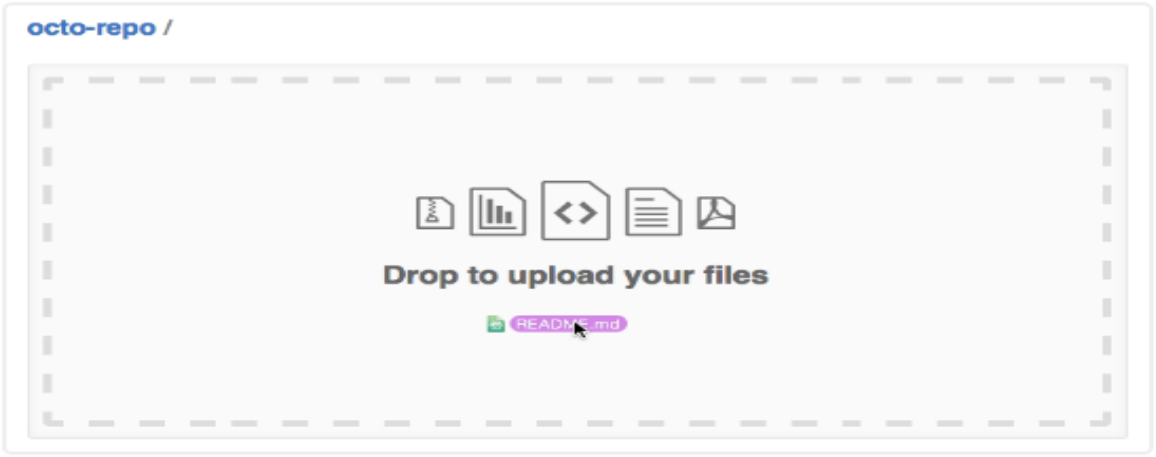
- You can upload multiple files to GitHub at the same time.
- If a repository has any protected branches, you can't edit or upload files in the protected branch using the web interface.

1. On GitHub, navigate to the main page of the repository.

2. Under your repository name, click **Upload files**.

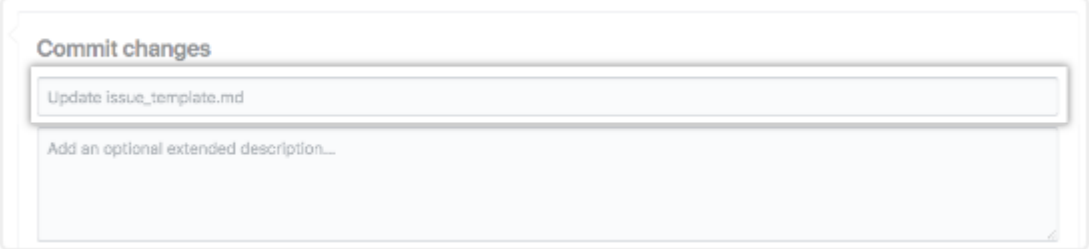


3. Drag and drop the file or folder you'd like to upload to your repository onto the file tree.

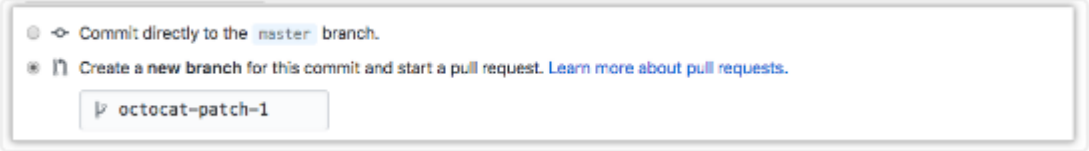


4. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in

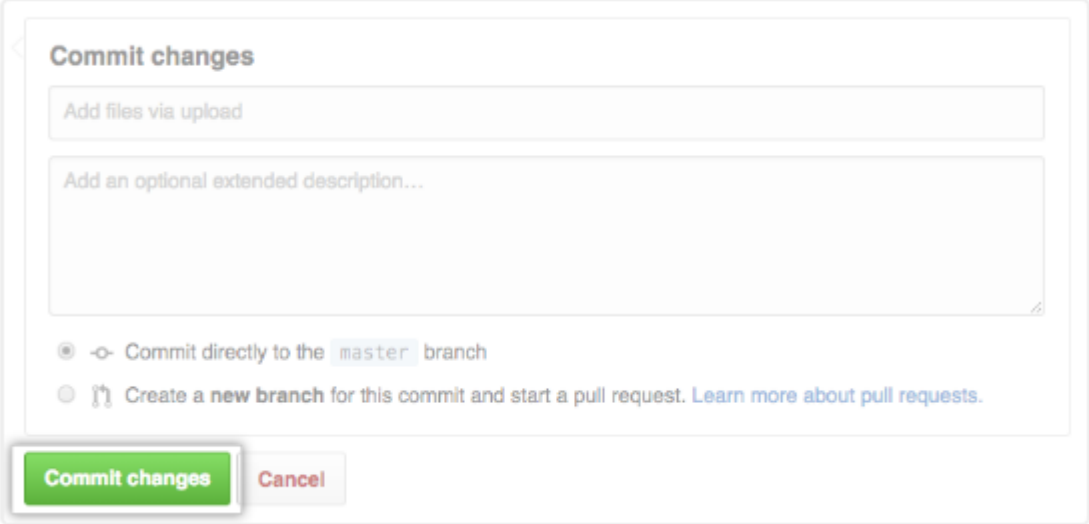
the commit message. For more information, see "Creating a commit with multiple co-authors."



5. Below the commit message fields, decide whether to add your commit to the current branch or to a new branch. If your current branch is master, you should choose to create a new branch for your commit and then create a pull request.



6 Click **Commit changes**.



- You can upload an existing file to a GitHub repository using the **command line.**
- You have to create repository

1. On your computer, move the file you'd like to upload to GitHub into the local directory that was created when you cloned the repository.
2. Open Git Bash.
3. Change the current working directory to your local repository.
4. Stage the file for commit to your local repository.

```
$ git add .
# Adds the file to your local repository and stages it for commit.
#To unstage a file, use 'git reset HEAD YOUR-FILE'.
```

5. Commit the file that you've staged in your local repository.

```
$ git commit -m "Add existing file"
# Commits the tracked changes and prepares them to be pushed to a remote
#repository. To remove this commit and modify the file, use 'git reset –
#soft HEAD~1' and commit and add the file again.
```

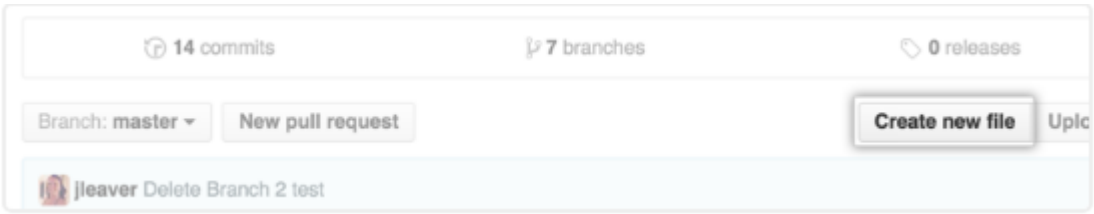
6. Push the changes in your local repository to GitHub.

```
$ git push origin your-branch
# Pushes the changes in your local repository up to the remote repository
#you specified as the origin
```

Creating new files

- You can create new files directly on GitHub in any repository you have write access to.

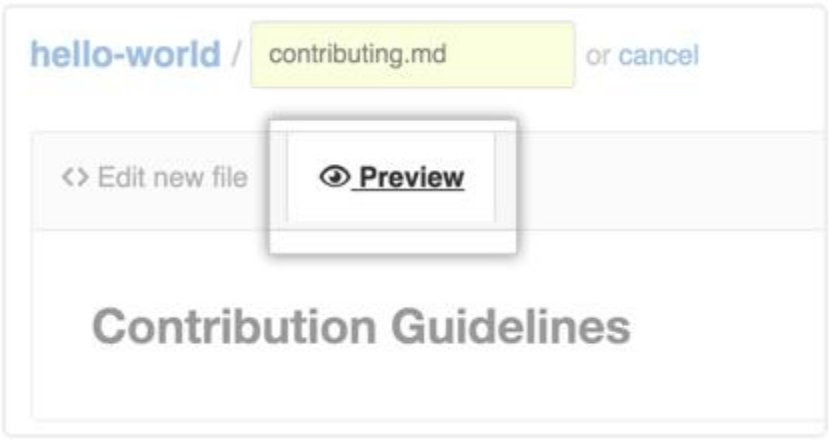
1. On GitHub, navigate to the main page of the repository.
2. In your repository, browse to the folder where you want to create a file.
3. Above the file list, click **Create new file**.



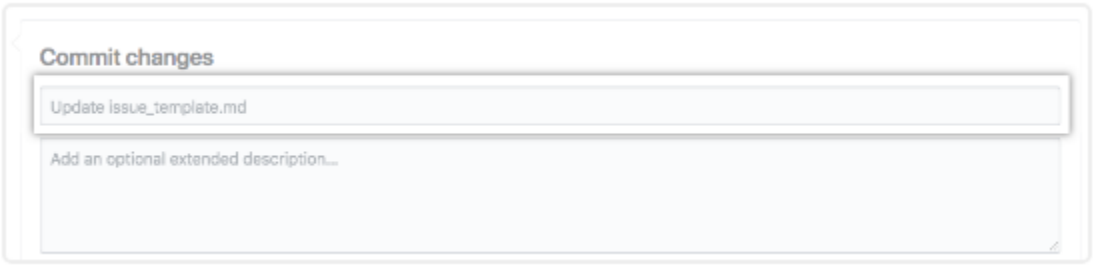
4. In the file name field, type the name and extension for the file. To create subdirectories, type the / directory separator.

5. On the **Edit new file** tab, add content to the file.

6. To review the new content, click **Preview**.



7. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see "Creating a commit with multiple co-authors."



Deleting files

- You can delete any file within your repositories on GitHub.

1. Browse to the file in your repository that you want to delete.
2. At the top of the file, click delete
3. At the bottom of the page, type a short, meaningful commit message that describes the change you made to the file. You can attribute the commit to more than one author in the commit message. For more information, see "Creating a commit with multiple co-authors."