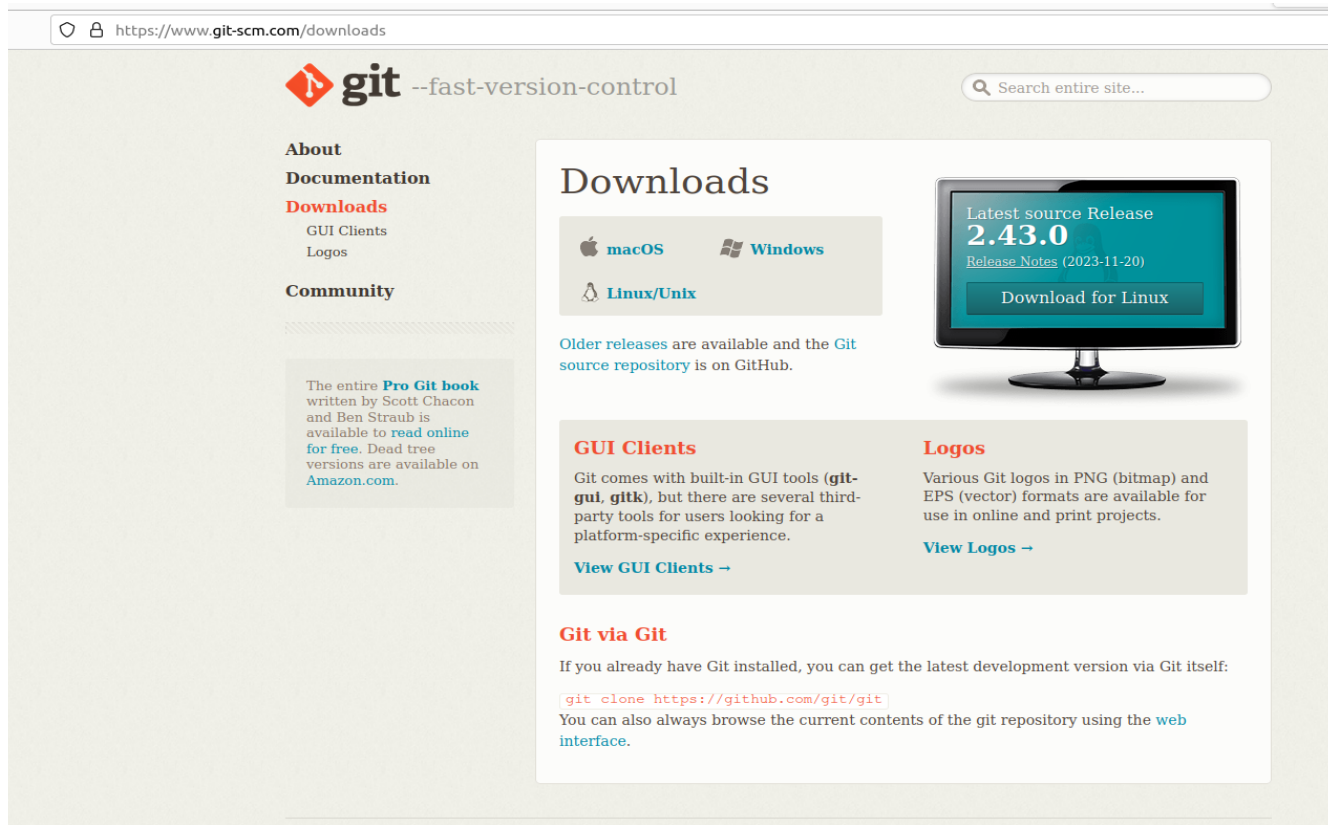


How to push code on Github

- **Step 1:** Install Git:

If you haven't installed Git on your machine, you can download and install it from the official website: [Git Downloads](https://git-scm.com/downloads)



- **Step 2:** Create a GitHub Repository:

- Log in to your GitHub account.

A screenshot of the GitHub sign-in page. At the top is the GitHub logo and the text "Sign in to GitHub". Below this is a form with two input fields: "Username or email address" and "Password". To the right of the password field is a link "Forgot password?". A green "Sign in" button is positioned below the password field. At the bottom of the form, there is a link "Sign in with a passkey" and a link "New to GitHub? Create an account".

- Click on the '+' sign in the top right

corner of the GitHub page and select "New repository."

- Fill in the necessary information, such as repository name, description, and others, then click on the "Create repository" button.

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)



Required fields are marked with an asterisk (*).

Owner * Repository name *

  /

Great repository names are short and memorable. Need inspiration? How about **probable-umbrella** ?

Description (optional)

- ☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.
- ☐  **Private**
You choose who can see and commit to this repository.

Initialize this repository with:

- ☐ **Add a README file**
This is where you can write a long description for your project. [Learn more about READMEs.](#)

Add .gitignore

.gitignore template: None ▾

Choose which files not to track from a list of templates. [Learn more about ignoring files.](#)

Choose a license

License: None ▾

A license tells others what they can and can't do with your code. [Learn more about licenses.](#)

 You are creating a public repository in your personal account.

Create repository

• **Step 3:** Open a Terminal or Command Prompt:

- On Windows, you can use the Command Prompt or PowerShell.
- On macOS or Linux, you can use the Terminal.

• **Step 4:** Navigate to Your Project Folder:

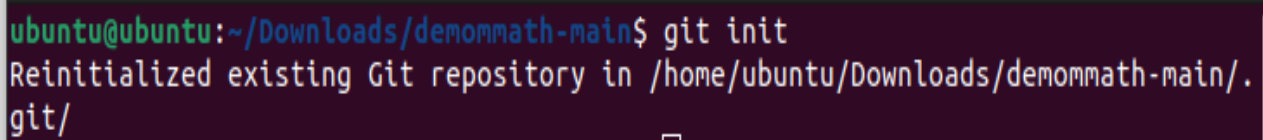
Use the `cd` command to navigate to the directory where your project folder is located.

```
```bash
cd path/to/your/project
```
```

- **Step 5:** Initialize a Git Repository (if not already done):

If your project is not already a Git repository, you'll need to initialize one.

```
```bash
git init
```
```

A terminal window with a dark purple background. The prompt is 'ubuntu@ubuntu:~/Downloads/demommath-main\$'. The command 'git init' has been entered. The output is 'Reinitialized existing Git repository in /home/ubuntu/Downloads/demommath-main/.git/'.

- **Step 6:** Add the Files/Folders to the Staging Area:

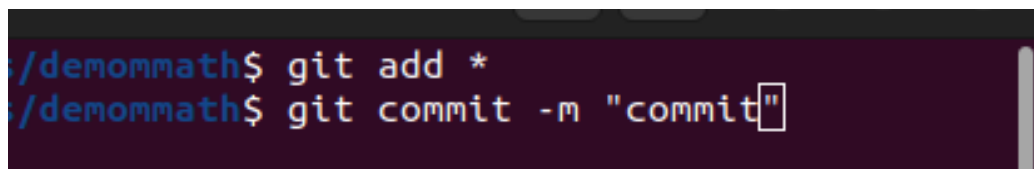
Use the following command to stage your files for the commit.

```
```bash
git add *
```
```

- **Step 7:** Commit the Changes:

Commit the staged changes with a meaningful message.

```
```bash
git commit -m "Initial commit"
```
```

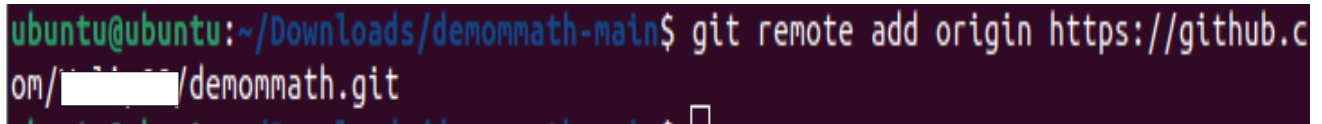
A terminal window with a dark purple background. The prompt is '/demommath\$'. The command 'git add *' has been entered. The next line shows the command 'git commit -m "commit"' with a cursor at the end of the string.

- **Step 8:** Link to the GitHub Repository:

Link your local repository to the GitHub repository you created.

```
```bash
git remote add origin https://github.com/your-username/your-repository.git
```
```

Make sure to replace `your-username` with your GitHub username and `your-repository` with the name of your repository.

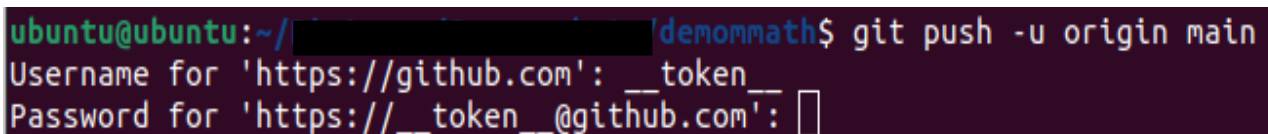
A terminal window with a dark background. The prompt is 'ubuntu@ubuntu:~/Downloads/demommath-main\$'. The command entered is 'git remote add origin https://github.com/[redacted]/demommath.git'. The cursor is at the end of the command.

```
ubuntu@ubuntu:~/Downloads/demommath-main$ git remote add origin https://github.com/[redacted]/demommath.git
```

- **Step 9:** Push the Changes:

Push your changes to GitHub.

```
```bash
git push -u origin main
```
```

A terminal window with a dark background. The prompt is 'ubuntu@ubuntu:~/[redacted]demommath\$'. The command entered is 'git push -u origin main'. Below the command, there are two prompts: 'Username for 'https://github.com': __token__' and 'Password for 'https://__token__@github.com':'. The cursor is at the end of the password prompt.

```
ubuntu@ubuntu:~/[redacted]demommath$ git push -u origin main
Username for 'https://github.com': __token__
Password for 'https://__token__@github.com':
```

If you're working with a different branch, replace `main` with the branch name.

- **Step10:** Enter GitHub Credentials:

You might be prompted to enter your GitHub username and password. Here you should need to use Personal access token. So First you need to create a personal access token.

- For Example
- username: __token__
- password: "token generate comes here"

- **Step 11:** Check GitHub Repository:

Visit your GitHub repository in a web browser to confirm that your folder and files have been successfully pushed.

That's it! Your folder is now pushed to your GitHub repository. Remember to update the repository URL and branch name according to your specific case.