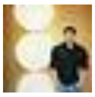
How to Use Git and GitHub



Biswaraj Sahoo

7 min read

·

Just now

Git and GitHub are common tools used in programming. They help you manage different versions of your code and collaborate with other developers.

Table of Contents

What are Git and GitHub?

Git was developed in 2005 by Linus Torvalds as *open source software* for tracking changes in a *distributed version control system*.

Git is open source because its source code is made freely available for anyone to modify and use, aside from its creator. Open-source projects are built and maintained collectively by different developers in different locations.

Git track changes via a distributed version control system. This means that Git can track the state of different versions of your projects while you're developing them. It is distributed because you can access your code files from another computer — and so can other developers.

When you're building an open source project, you'll need a way to document or track your code. This helps make your work organized, and lets you keep track of the changes you've made. This is what Git lets you do.

But you also need a place to host your code — which makes controlling each version of your project easier and faster. This is where GitHub comes in.

GitHub is a “hub” (a place or platform) where Git users build software together. GitHub is also an hosting provider and version control platform you can use to collaborate on open source projects and share files. When you’re using GitHub, you’re working with Git beneath the hood.

Why Should You Learn Git and GitHub?

According to Techmonitor.ai, over 73 million developers use GitHub as of November 2021. And the GitHub community is set to hit 100 million users by 2025.

As you can see, millions of people all over the world use these tools, and the numbers just keep going up.

Because of this, more companies are requiring new hires to know how to use Git and GitHub. So if you’re looking for a developer job, these are essential skills to have.

If you’re not using Git and GitHub, it’s clear — you should be!

Differences between Git and GitHub

Git is a version control system that manages and keeps track of your code. GitHub, on the other hand, is a service that let you host, share, and manage your code files on the internet.

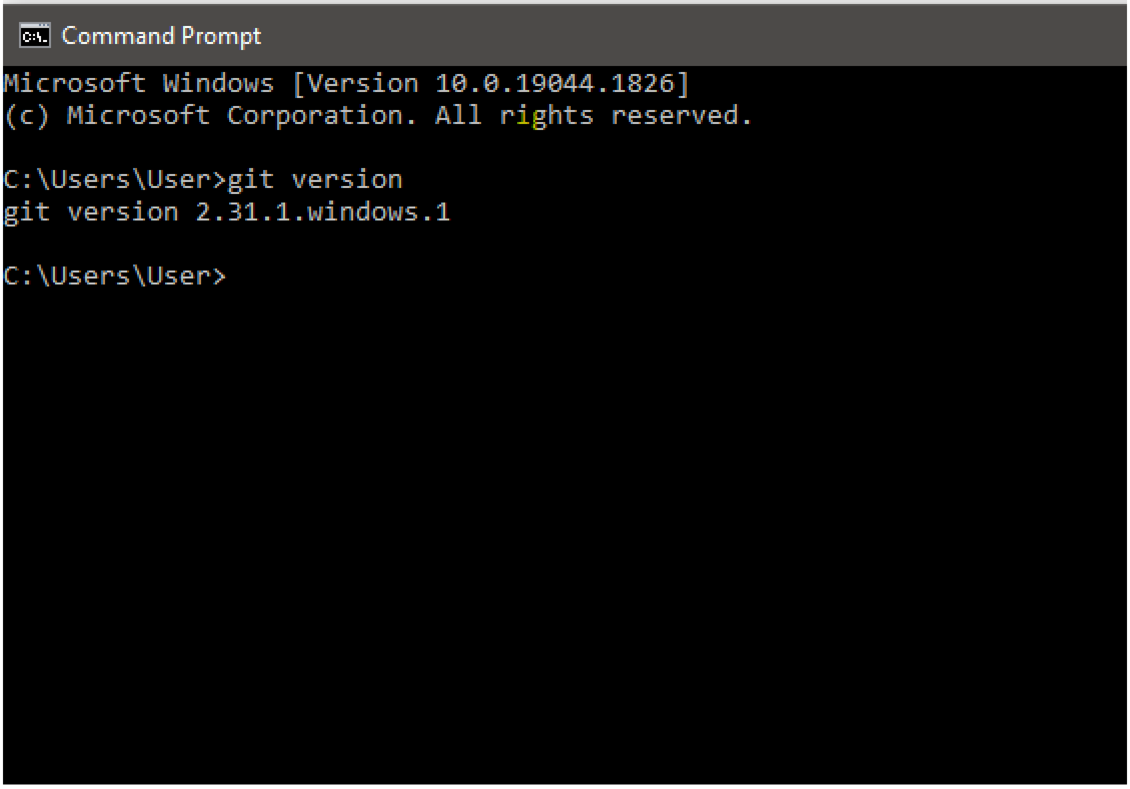
GitHub uses Git underneath, and lets you manage your Git repositories or folders easily on its platform.

So Git is the actual version control system and GitHub is the platform where you host your code.

How to Start Using Git and GitHub

Step 1 — Install Git

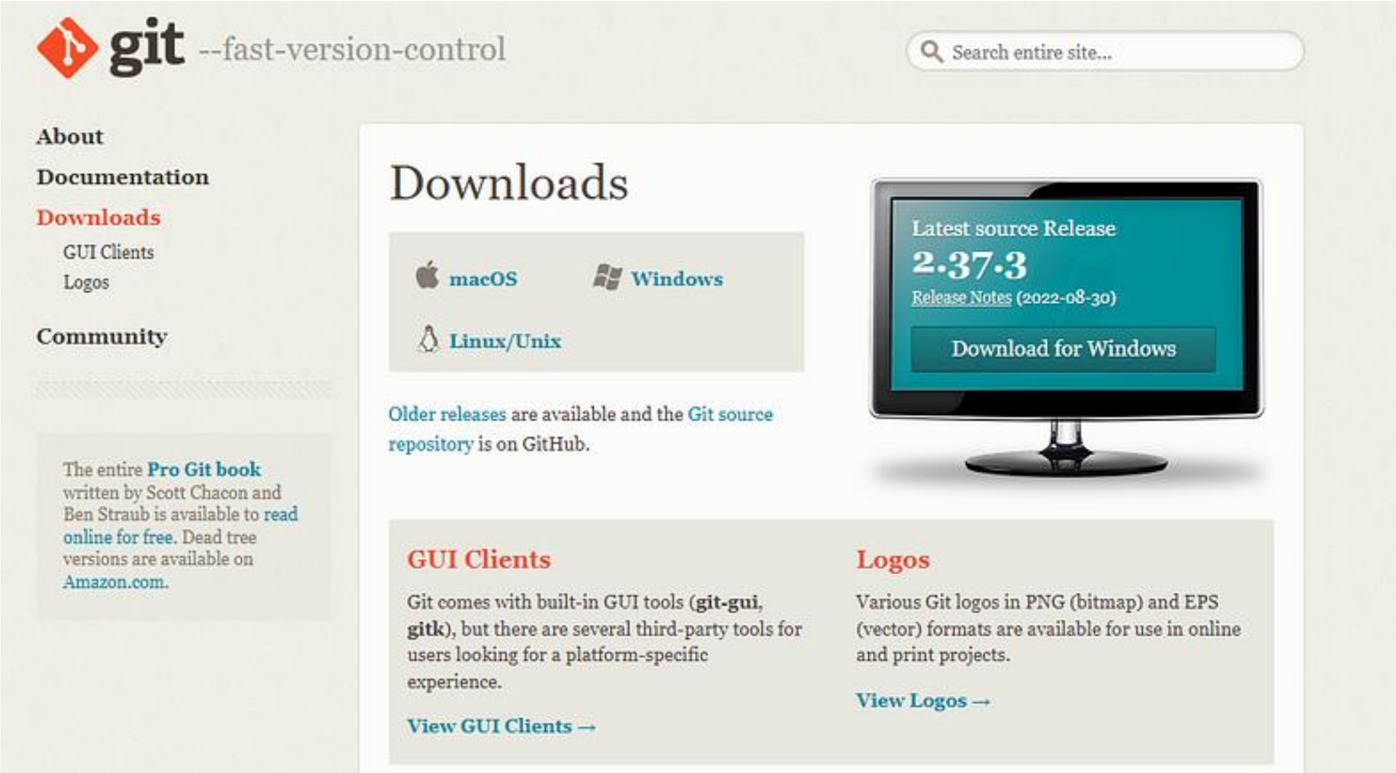
Git comes preinstalled in some Macs and Linux-based systems, but you can always check if you have Git installed in your machine by typing `git version` in your terminal. You can use the Command Prompt to do this.



```
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

C:\Users\User>git version
git version 2.31.1.windows.1

C:\Users\User>
```

As you can see above, I have Git version 2.31.1 installed on my Windows computer. If you don't have Git installed in your computer, you won't get a version.

You can download Git here and then select your operating system to download.



Downloads

Older releases are available and the Git source repository is on GitHub.

Latest source Release
2.37.3
Release Notes (2022-08-30)

The entire Pro Git book written by Scott Chacon and Ben Straub is available to read online for free. Dead tree versions are available on Amazon.com.

GUI Clients

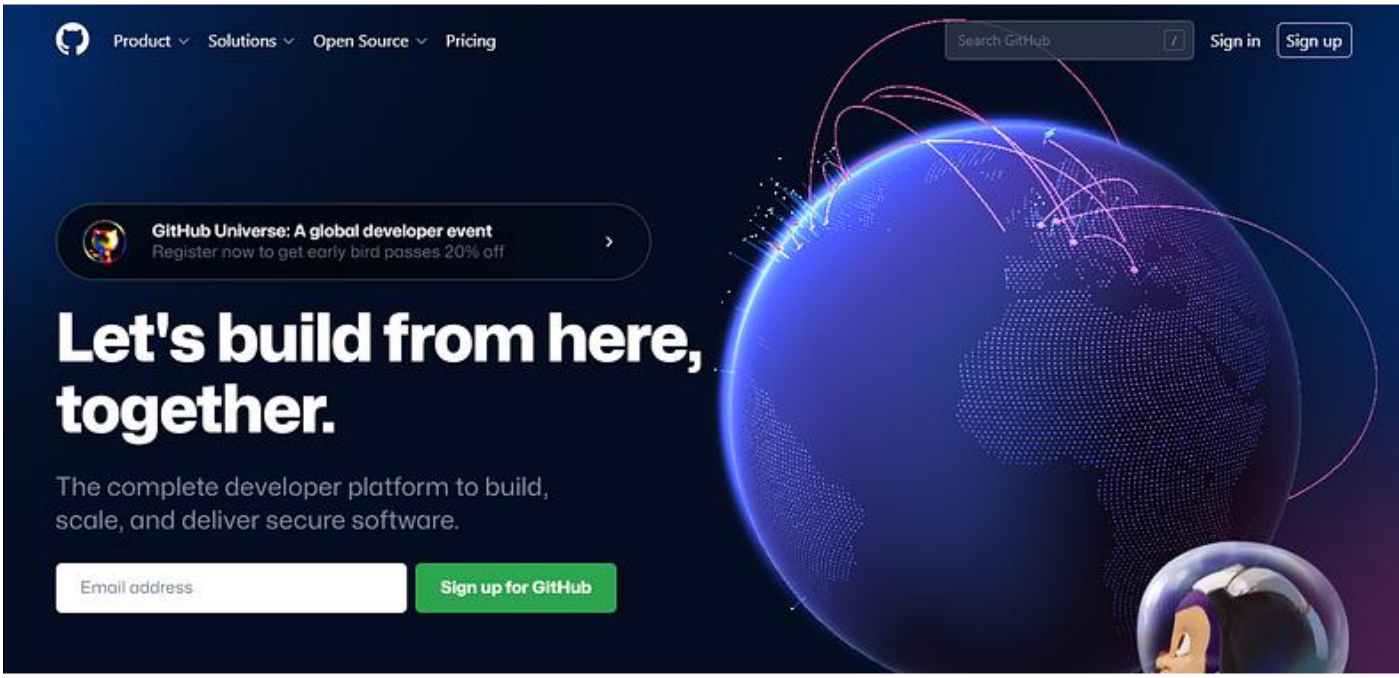
Git comes with built-in GUI tools (**git-gui**, **gitk**), but there are several third-party tools for users looking for a platform-specific experience.

Logos

Various Git logos in PNG (bitmap) and EPS (vector) formats are available for use in online and print projects.

Follow the necessary installer guide until installation is complete. Open the command prompt and type `git version` to verify that Git was successfully installed.

Step 2 — Create a GitHub Account.



GitHub Universe: A global developer event
Register now to get early bird passes 20% off

Let's build from here, together.

The complete developer platform to build, scale, and deliver secure software.

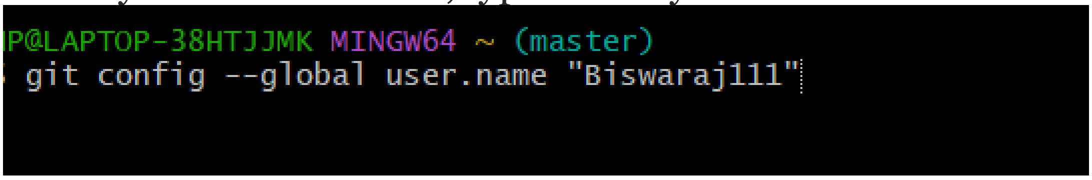
To create an account on GitHub, you will be asked for some personal information like name, confirm your email, set a username and password, and your account should be set up in minutes.

Create an account on GitHub.com here.

Step 3 — Connect your GitHub account to your Git account.

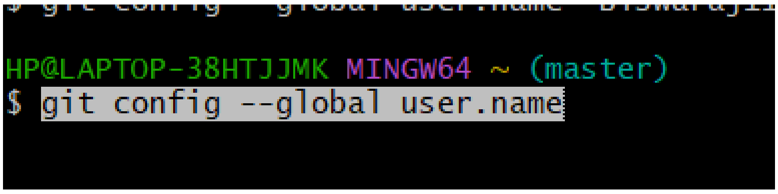
You’ll do this from your terminal.

To set your Git username, type this in your terminal:



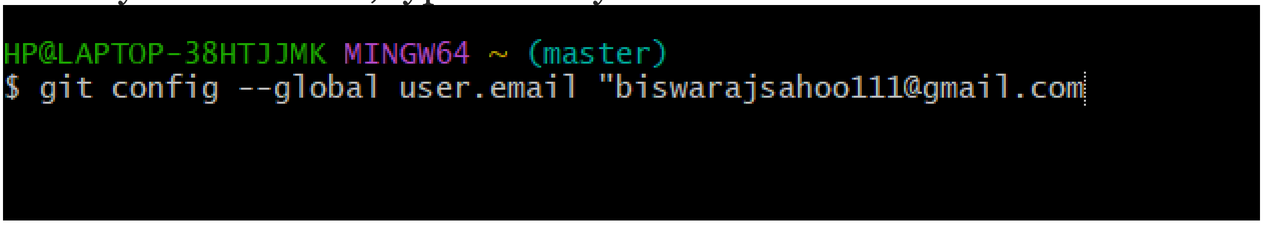
```
P@LAPTOP-38HTJJMK MINGW64 ~ (master)
 git config --global user.name "Biswaraj111"
```

To confirm that you have set your Git username correctly, type this:



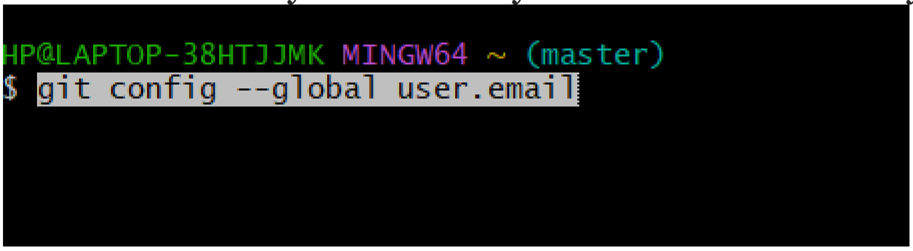
```
HP@LAPTOP-38HTJJMK MINGW64 ~ (master)
$ git config --global user.name
```

To set your Git email, type this in your terminal:



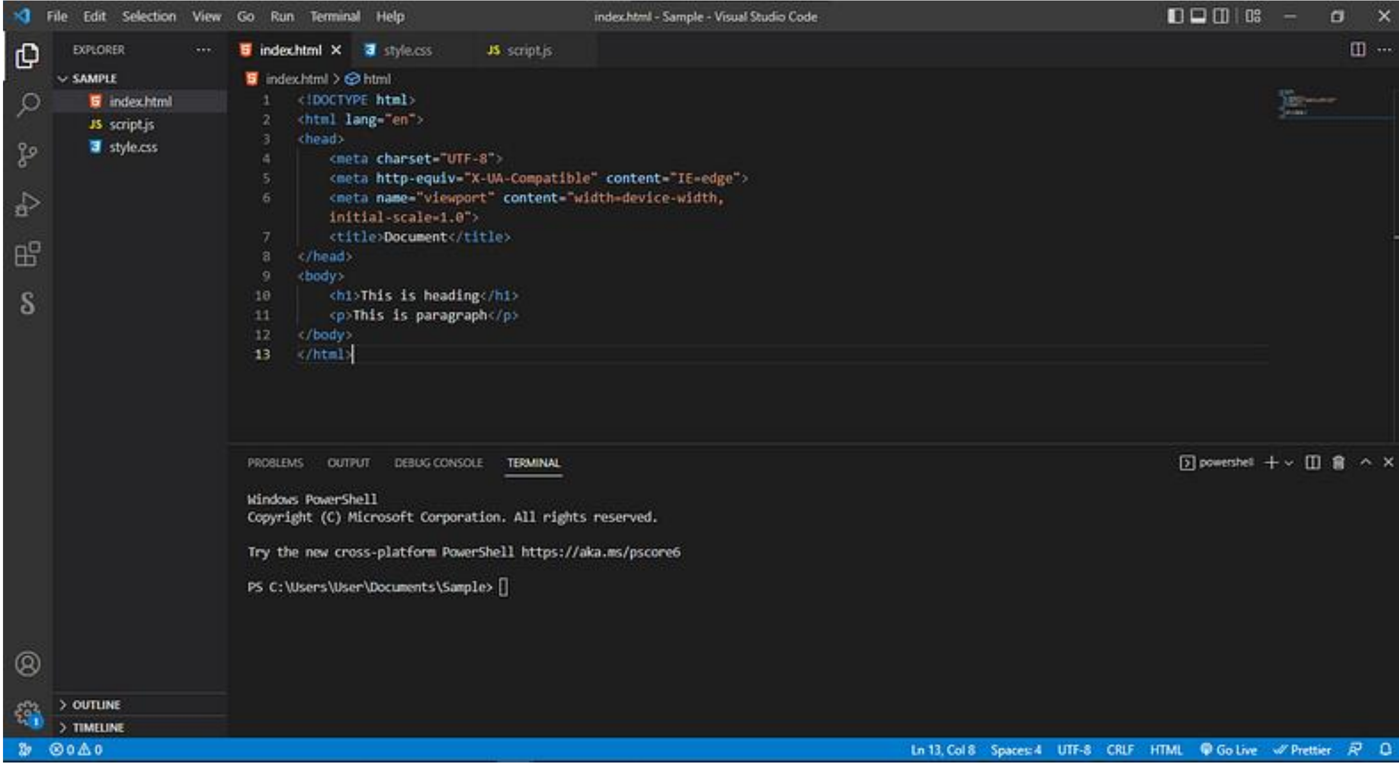
```
HP@LAPTOP-38HTJJMK MINGW64 ~ (master)
$ git config --global user.email "biswarajsahoo111@gmail.com
```

To confirm that you have set your Git email correctly, type this:



```
HP@LAPTOP-38HTJJMK MINGW64 ~ (master)
$ git config --global user.email
```

Step 4 — Create and edit your code files locally



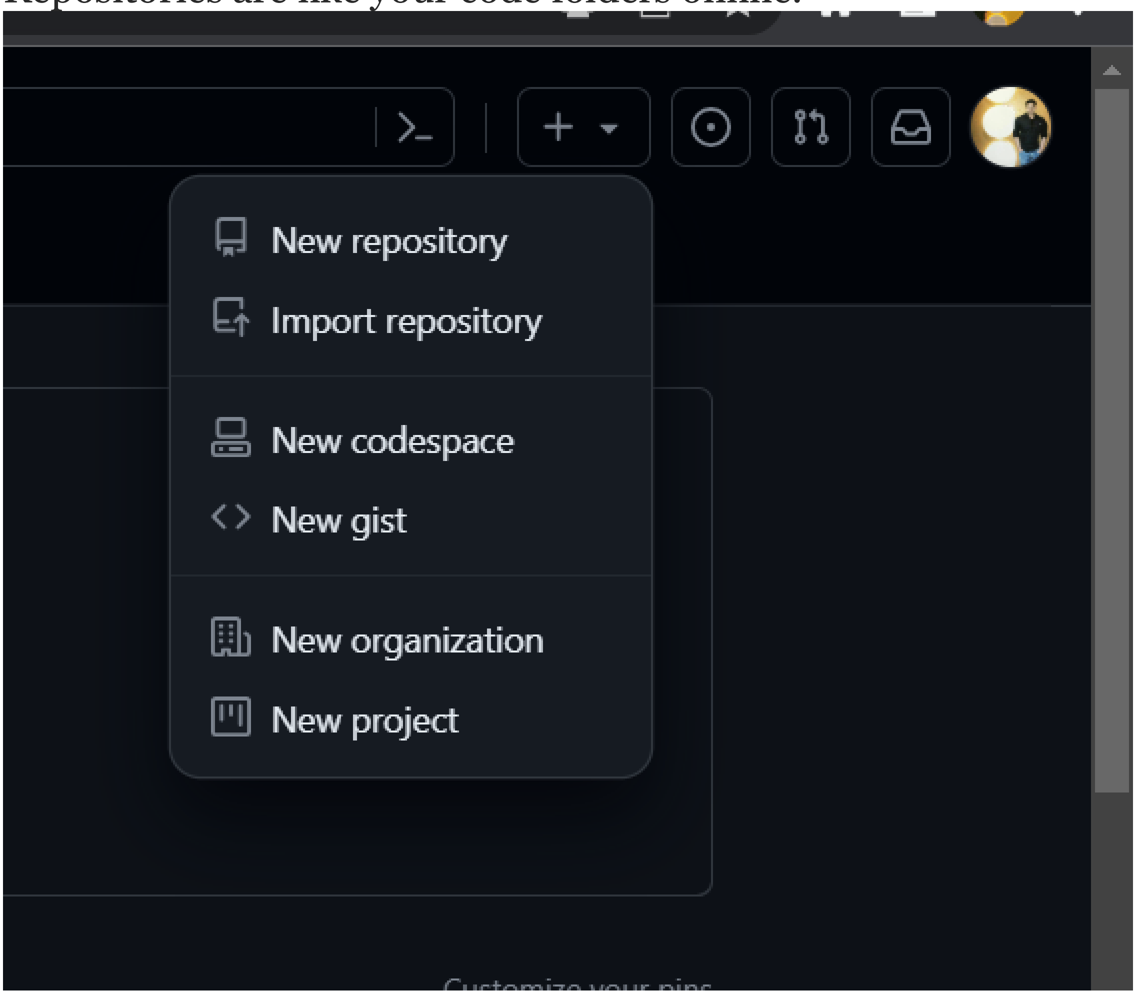
File Edit Selection View Go Run Terminal Help
index.html - Sample - Visual Studio Code
EXPLORER
SAMPLE
index.html
script.js
style.css
index.html
style.css
script.js
index.html > html
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
OUTLINE
TIMELINE
Ln 13, Col 8 Spaces: 4 UTF-8 CRLF HTML Go Live Prettier

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
    <title>Document</title>
</head>
<body>
    <h1>This is heading</h1>
    <p>This is paragraph</p>
</body>
</html>
```

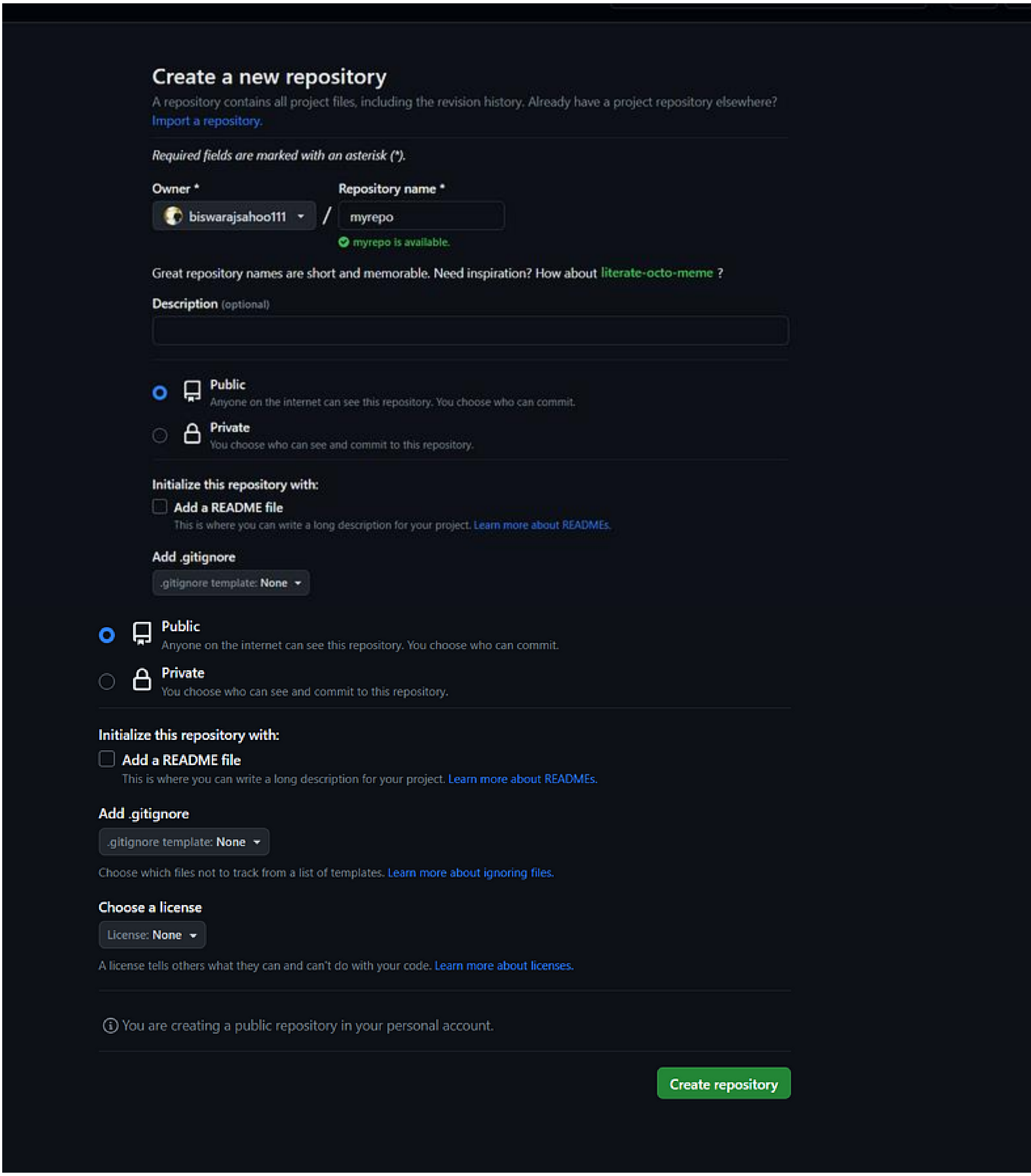
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\User\Documents\Sample>
```

Step 5 — Create a repository on GitHub

Click the + sign at the top right corner to create a new repository. Repositories are like your code folders online.



You will be prompted to this page:



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 * biswarajsahoo111 / Repository name * myrepo

✓ myrepo is available.

Great repository names are short and memorable. Need inspiration? How about literate-octo-meme ?

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

● 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

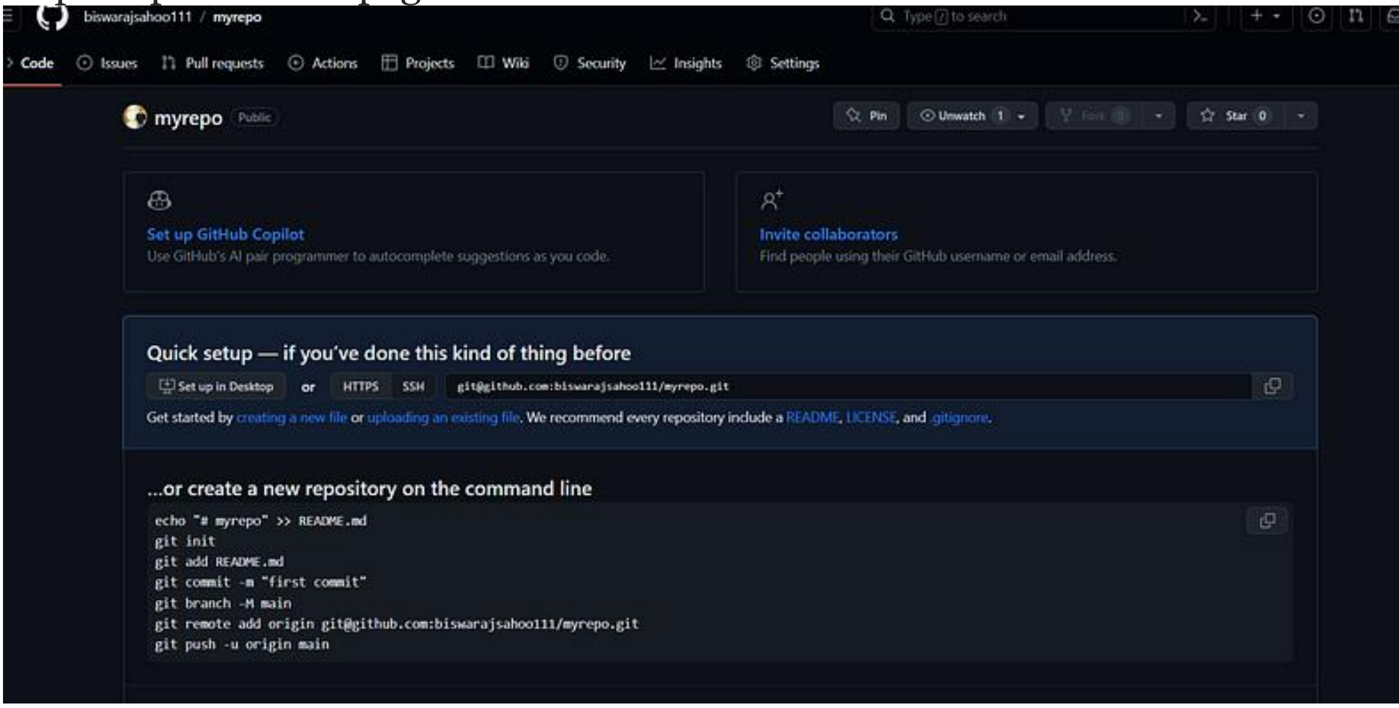
License: None

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.

Name your repository and give it a description (this is optional).

Click the “Create repository” button to create the repository. You will be prompted to this page



Set up GitHub Copilot

Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Invite collaborators

Find people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

```
git@github.com:biswarajsahoo111/myrepo.git
```

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

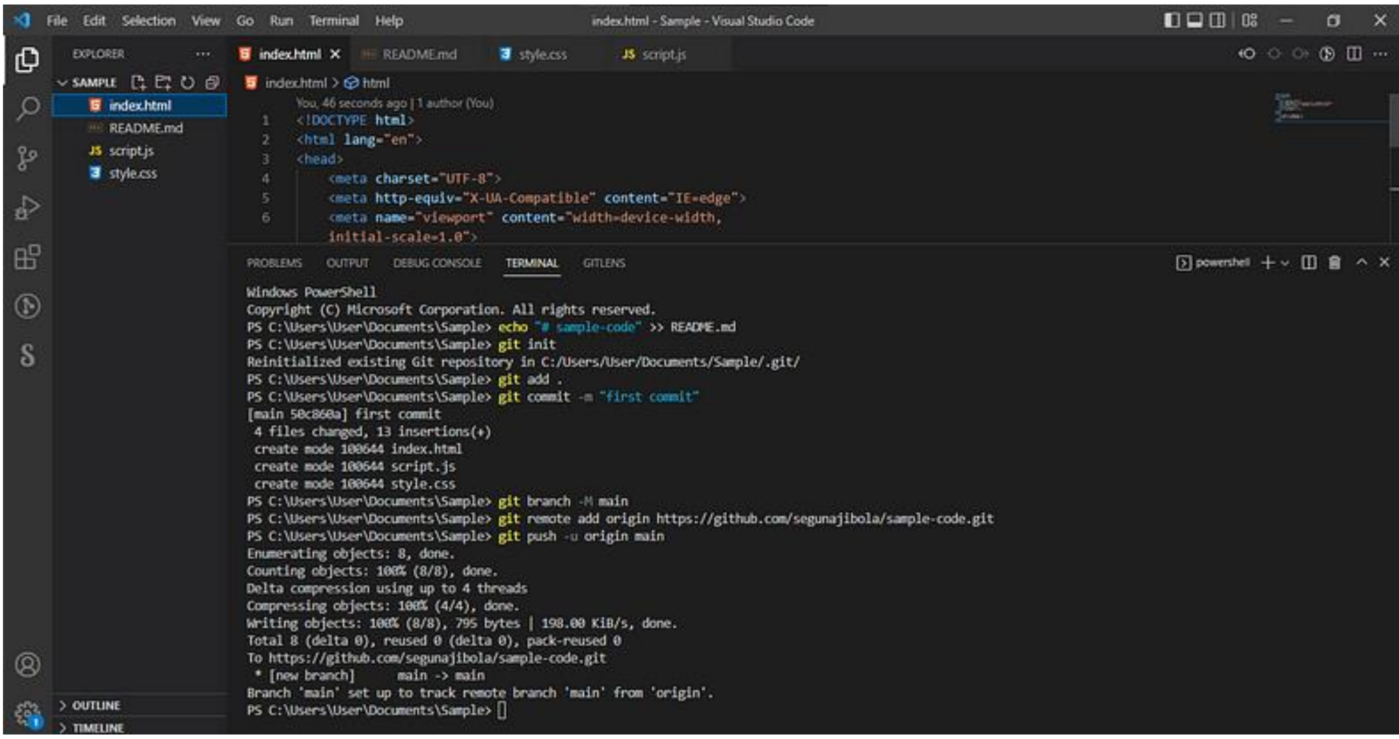
...or create a new repository on the command line

```
echo "# myrepo" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin git@github.com:biswarajsahoo111/myrepo.git
git push -u origin main
```

Step 6 — Push your local code to GitHub

You can use the code editor built-in terminal to use Git to push your code to GitHub. Click `ctrl` + `shift` + `'` to open the terminal in VSCode.

Input the commands below one after the other in your terminal. Press the `Enter` key to proceed after every input.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
PS C:\Users\User\Documents\Sample> echo "# sample-code" >> README.md
PS C:\Users\User\Documents\Sample> git init
Reinitialized existing Git repository in C:/Users/User/Documents/Sample/.git/
PS C:\Users\User\Documents\Sample> git add .
PS C:\Users\User\Documents\Sample> git commit -m "first commit"
[main 50c860a] first commit
 4 files changed, 13 insertions(+)
 create mode 100644 index.html
 create mode 100644 script.js
 create mode 100644 style.css
PS C:\Users\User\Documents\Sample> git branch -M main
PS C:\Users\User\Documents\Sample> git remote add origin https://github.com/segunajibola/sample-code.git
PS C:\Users\User\Documents\Sample> git push -u origin main
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 795 bytes | 198.00 KiB/s, done.
Total 8 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/segunajibola/sample-code.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
PS C:\Users\User\Documents\Sample> 
```

```
echo "# sample-code" >> README.md
```

```
git init
```

```
git add .
```

```
git commit -m "first commit"
```

```
git branch -M main
```

```
git remote add origin https://github.com/segunajibola/sample-code.git
```

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

Note that we have `git add README.md` in the repository on GitHub. But here we have `git add .`, which is to let Git add all our code files instead of the `README.md` file which will be created by `echo "# sample-code" >> README.md`. So if you have created other files in your local folder, you need to use `git add .` to add all files.

Take note that `git remote add origin https://github.com/segunajibola/sample-code.git` will contain the link to your own repository and it will have have the name of your GitHub account.

Common Git Commands to Know

They are many Git commands you can use in the terminal, and that can get overwhelming. So I'd suggest focusing on some of the most popular ones first.

Here they are:

`git init` lets you initialize Git in your folder.

`git add Readme.md` lets you add the Readme file, while `git add .` lets you add all files in the present folder.

`git commit` stores the added files. Use `-m` for message followed by the actual message.

`git branch` creates a new branch which is a new version of the repository as it appears when added, and `-M` to move the name to `main`.

`git remote add origin` finally connects the local folder to the repository on GitHub. It is followed by the repository's link.

`git push -u origin main` pushes the code to GitHub. The `-u` flag creates a tracking reference for the branch, and `origin main` puts the code in the `main` branch.

Those are some of the main commands you'll use all the time. This is a beginner and non-technical guide to help you get started using Git and GitHub, so we won't go into too much more detail here.

The more you continue using GitHub, the more comfortable you'll get using these commands. The key is to start small and maintain your momentum.

It will eventually get easier as you build small projects and host them on GitHub using Git.

If you find it hard to use the terminal to navigate between folders, spend some time to practice with it. Again, it gets easier with time and use.

How to Customize Your GitHub Profile

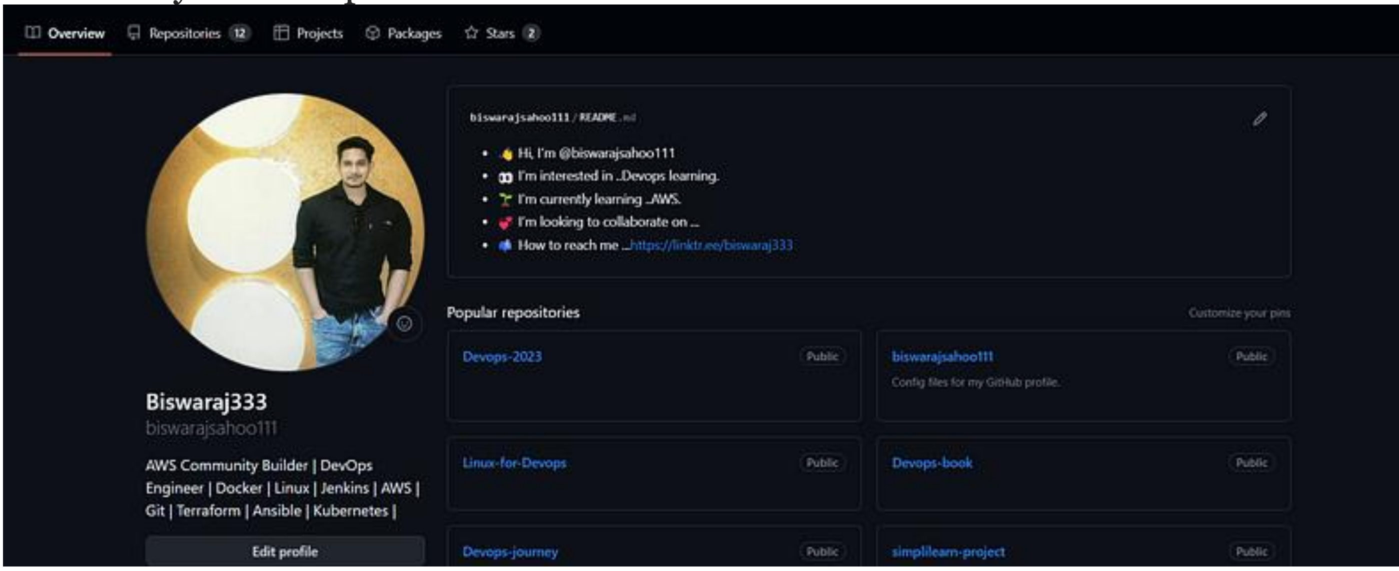
Customizing your GitHub profile README helps you stand out from random GitHub users.

The README.md file helps you describe your GitHub profile, and you can use it to show what you're currently learning along with your skills and contributions.

The GitHub README.md uses markdown to format its content. It has an easy-to-learn syntax.

Here is a simple guide to create and customize your GitHub account.

Here is my GitHub profile's README.md file.



biswarajsahoo111 / README.md

- 👋 Hi, I'm @biswarajsahoo111
- 👀 I'm interested in ...Devops learning.
- 🌱 I'm currently learning ...AWS.
- 💞️ I'm looking to collaborate on ...
- 📫 How to reach me ...https://linktr.ee/biswaraj333

Biswaraj333
biswarajsahoo111

AWS Community Builder | DevOps Engineer | Docker | Linux | Jenkins | AWS | Git | Terraform | Ansible | Kubernetes |

Popular repositories

Devops-2023 Public

biswarajsahoo111 Public
Config files for my GitHub profile.

Linux-for-Devops Public

Devops-book Public

Devops-journey Public

simplilearn-project Public

— AWS Community Builder | DevOps Engineer | Docker | Linux | Jenkins | AWS | Git | Terraform | Docker | kubernetes

Empowering communities via open source and education.

Connect with me over linktrr: https://linktr.ee/biswaraj333

Git

Github

Repositories

DevOps