

Updated March 16, 2025

# The Beginner's Guide to Git

A Simplified Approach to Version  
Control



Git Logo by Jason Long is licensed under the Creative Commons Attribution 3.0 Unported License.

Abdirashid Ahmed  
ENGL335

# Introduction

## What is Git?

Git, released on April 7th, 2005, is a version control system that allows you to take checkpoints in your progress and contribute them to building a history of your project. In 2018, a survey by StackOverflow revealed that nearly 90% of professional software developers use Git, while "less than 4% confessed to not using a version control system at all" (FutureLearn, 2022). Git is a useful, almost necessary, tool to learn as it contributes immensely to the software industry.

## Who is this guide for?

This guide is intended for beginner coders, programmers, solo developers, and anyone else who simply wants to learn how to use Git.

## How to use this guide

This is a straightforward guide with several images and visuals to help you set up and learn how to use Git.

---

## 1. Installing Git

- Go to [Git's official website](#).
- Select your operating system (Windows, macOS, Linux).
- Download and install Git where it states, 'Click here to download'.
- Open command prompt or a terminal of your choice to verify by typing:

```
git -version
```

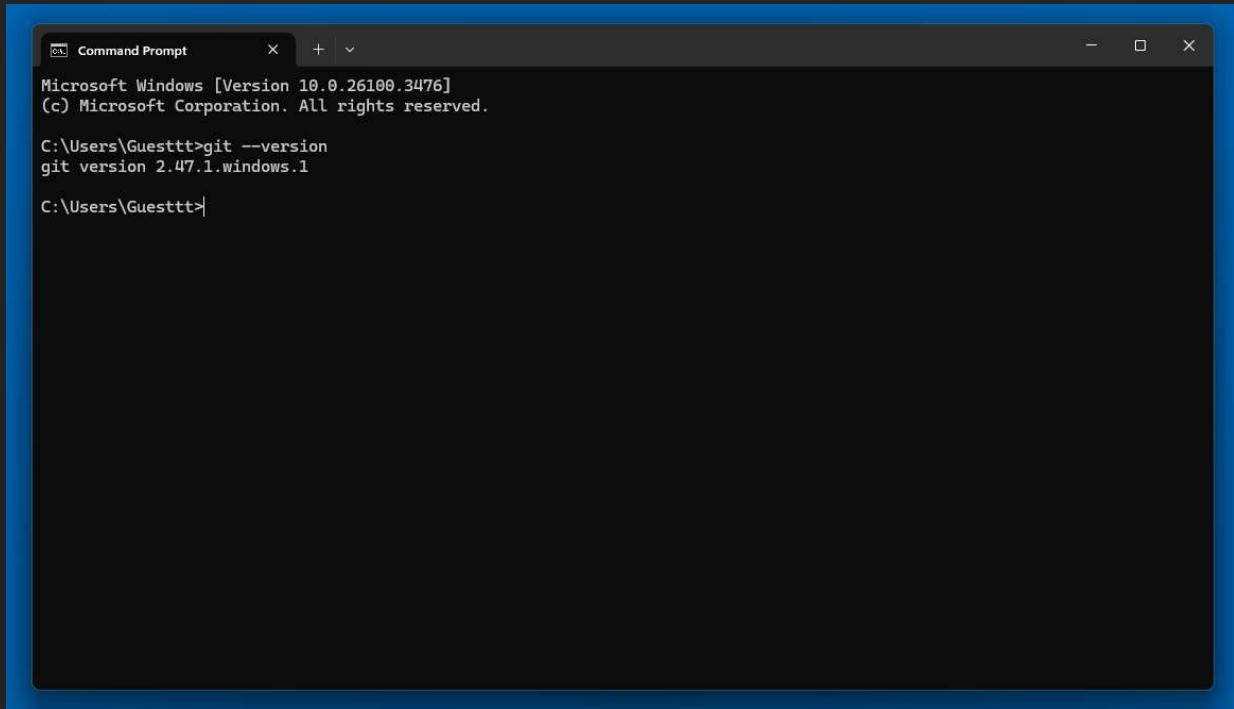


Figure 1. Screenshot showing successful ‘git –version’ command. Source: Abdirashid Ahmed (2025).

---

## 2. Setting up Git

- Set up your username by typing:

```
git config -global user.name "Your Name"
```

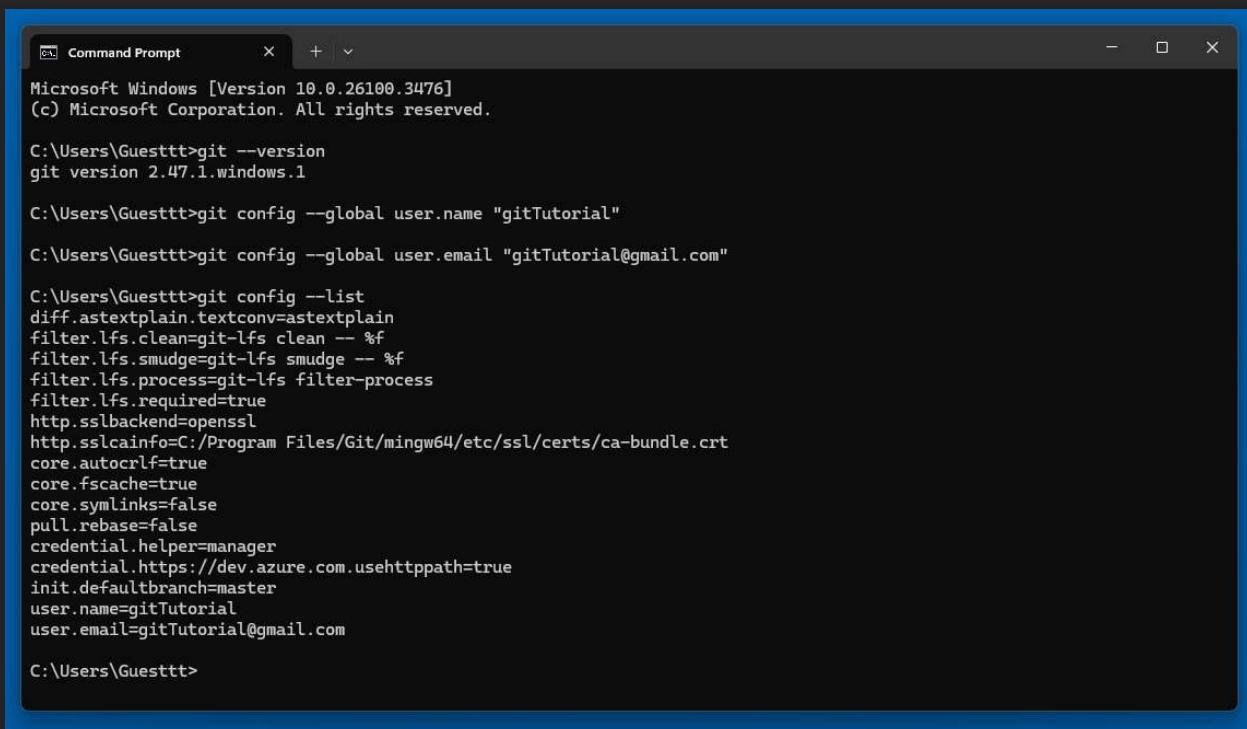
- Set up your email by typing:

```
git config -global user.email "youremail@example.com"
```

- Confirm your settings with:

```
git config list
```

This should display lines of text but simply look at the last two rows. It should show your set username and email.



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following command-line session:

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Guesttt>git --version
git version 2.47.1.windows.1

C:\Users\Guesttt>git config --global user.name "gitTutorial"
C:\Users\Guesttt>git config --global user.email "gitTutorial@gmail.com"

C:\Users\Guesttt>git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/etc/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.name=gitTutorial
user.email=gitTutorial@gmail.com

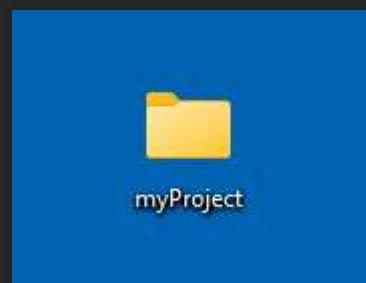
C:\Users\Guesttt>
```

Figure 2. Screenshot showing successful account setup commands. Source: Abdirashid Ahmed (2025).

---

### 3. Making a New Git Repository

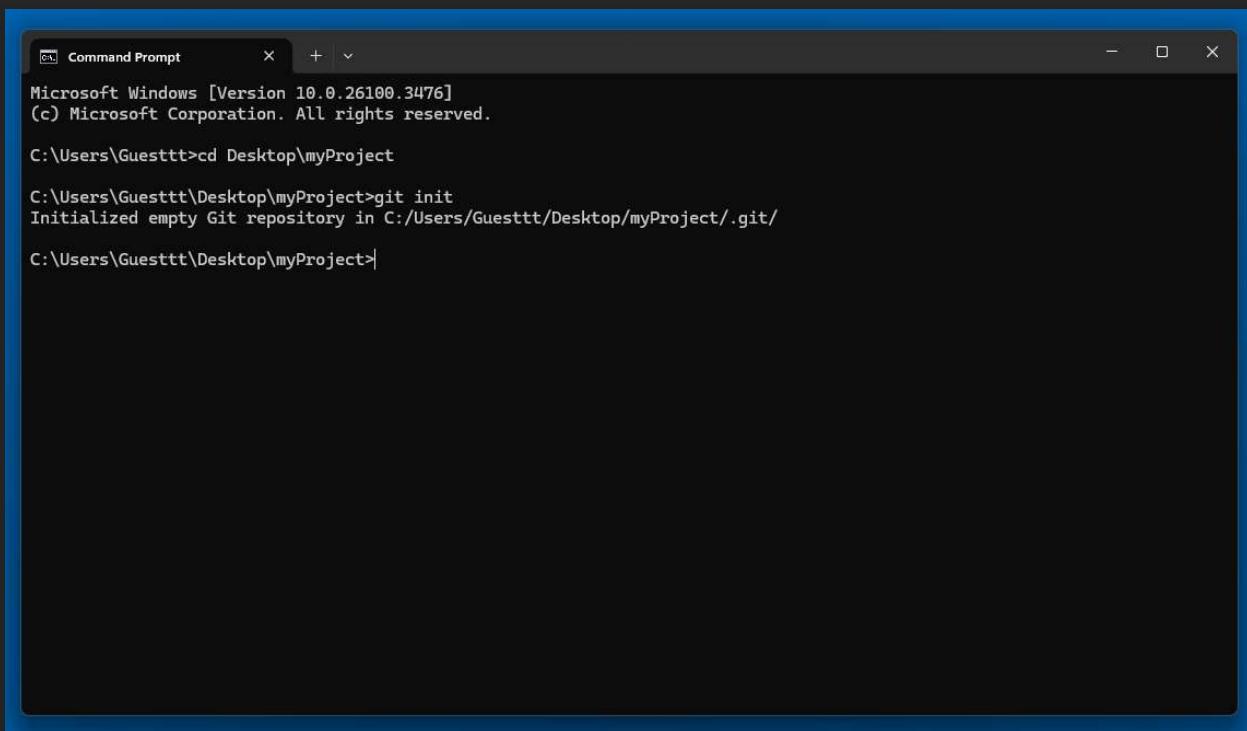
- Navigate to your project folder in your terminal. In my example case, I made a folder on my desktop called 'myProject', so I would type 'cd Desktop\myProject'. This could be different for you depending on where your project exists.



```
cd path\to\yourProject
```

- Set up a new repository with:

```
git init
```

A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the following text:

```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Guesttt>cd Desktop\myProject

C:\Users\Guesttt\Desktop\myProject>git init
Initialized empty Git repository in C:/Users/Guesttt/Desktop/myProject/.git/

C:\Users\Guesttt\Desktop\myProject>
```

Figure 3. Screenshot showing successful project initialization commands.

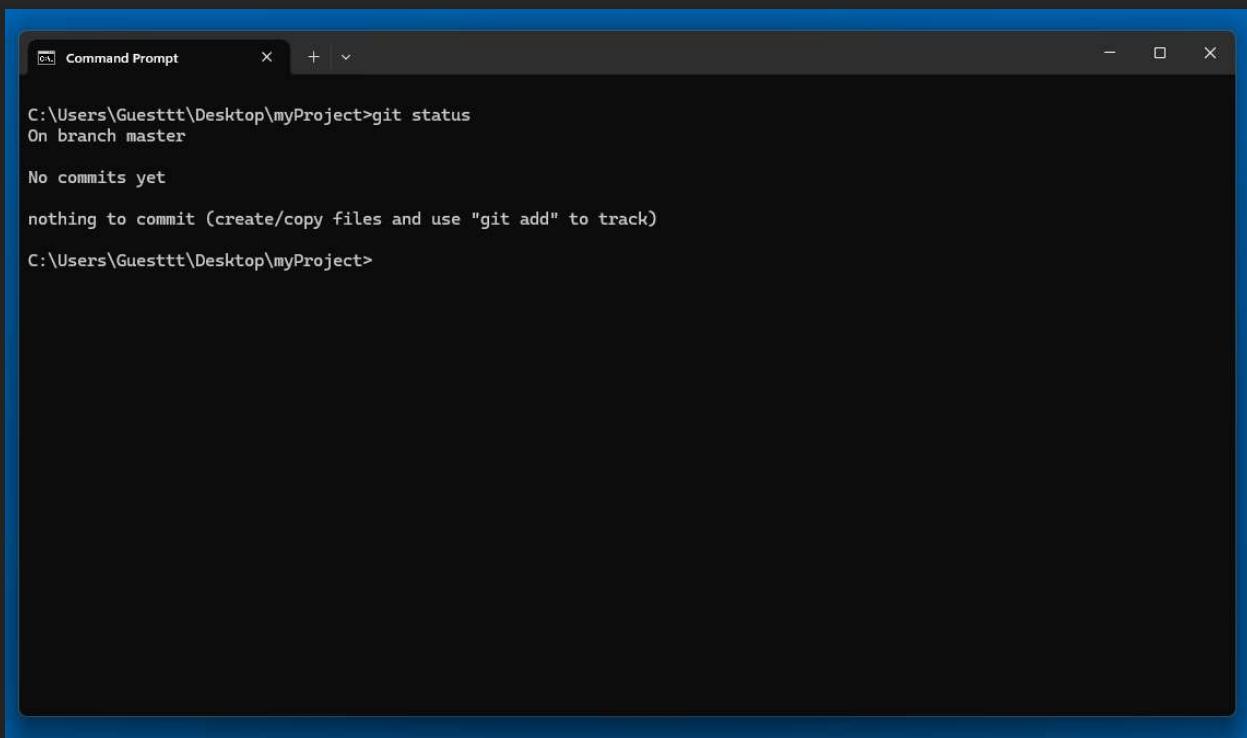
Source: Abdirashid Ahmed (2025).

---

## 4. Checking the Status of Your Project

- Check the current status of your repository to make sure everything is good so far:

```
git status
```



```
C:\Users\Guesttt\Desktop\myProject>git status
On branch master
No commits yet
nothing to commit (create/copy files and use "git add" to track)
C:\Users\Guesttt\Desktop\myProject>
```

Figure 4. Screenshot showing ‘git status’ command. Source: Abdirashid Ahmed (2025).

---

## 5. Adding Files to be Tracked

- Open your code editor. I will be using VS Code for this guide.
- Select your terminal, I will be using Command Prompt.

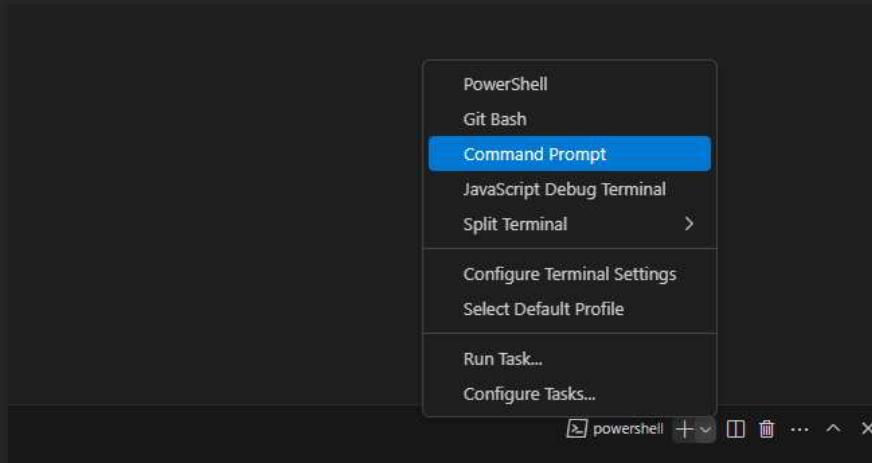
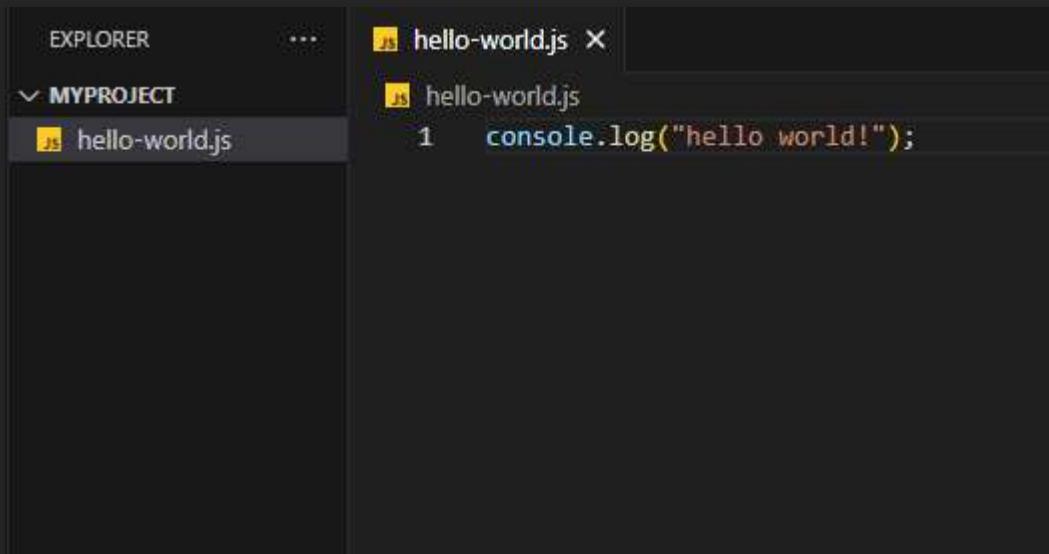


Figure 5. Screenshot showing list of terminal options. Source: Abdirashid Ahmed (2025).

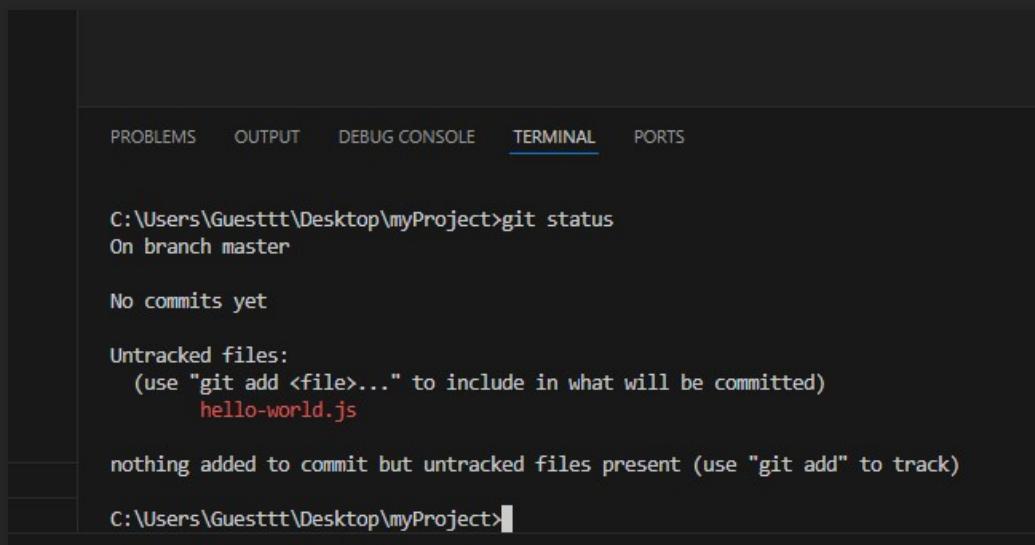
- Starting building and progressing on your project. Whenever you have enough progress that you would not feel comfortable losing, this is when you add and save, or ‘commit’, your changes.
- In this example, I will simply make a file that will print out ‘hello world!’ into the console:



The screenshot shows the Visual Studio Code interface. On the left, the Explorer sidebar displays a project named 'MYPROJECT' with a file 'hello-world.js'. The main editor area shows the code: '1 console.log("hello world!");'. The status bar at the bottom indicates the file path: 'C:\Users\Guesttt\Desktop\myProject\hello-world.js'.

Figure 6. Screenshot showing a JavaScript file with one line of code that prints ‘hello world!’.  
Source: Abdirashid Ahmed (2025).

**Pro Tip:** A good practice is to constantly do ‘git status’ to make sure you’re aware of what stages you are on Git.



The screenshot shows the VS Code terminal window. The tab bar at the top has 'TERMINAL' underlined, indicating it is active. The terminal output shows the command 'git status' being run in the directory 'C:\Users\Guesttt\Desktop\myProject'. The output indicates the user is on the 'master' branch and there are no commits yet. It also lists an untracked file, 'hello-world.js', and states that nothing has been added to commit. The prompt 'C:\Users\Guesttt\Desktop\myProject>' is visible at the bottom.

```
C:\Users\Guesttt\Desktop\myProject>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    hello-world.js

nothing added to commit but untracked files present (use "git add" to track)

C:\Users\Guesttt\Desktop\myProject>
```

Figure 7. Screenshot showing ‘git status’ command. Source: Abdirashid Ahmed (2025).

- Now, add your specific file, or files, by typing them by name in the console with:

```
git add filename.txt
```

```
git add filename.txt filename2.txt filename3.txt
```

- Or to simply add all:

```
git add .
```

```
C:\Users\Guesttt\Desktop\myProject>git add "hello-world.js"
C:\Users\Guesttt\Desktop\myProject>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   hello-world.js

C:\Users\Guesttt\Desktop\myProject>
```

Figure 8. Screenshot showing ‘git add’ command. Source: Abdirashid Ahmed (2025).

**Pro Tip:** Use git status after adding files to verify they’re staged for saving.

---

## 6. Committing Changes

- Commit (which is to ‘save’) your changes with a message:

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

```
C:\Users\Guesttt\Desktop\myProject>git commit -m "Initial commit"
[master (root-commit) 1136bb8] Initial commit
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 hello-world.js

C:\Users\Guesttt\Desktop\myProject>
```

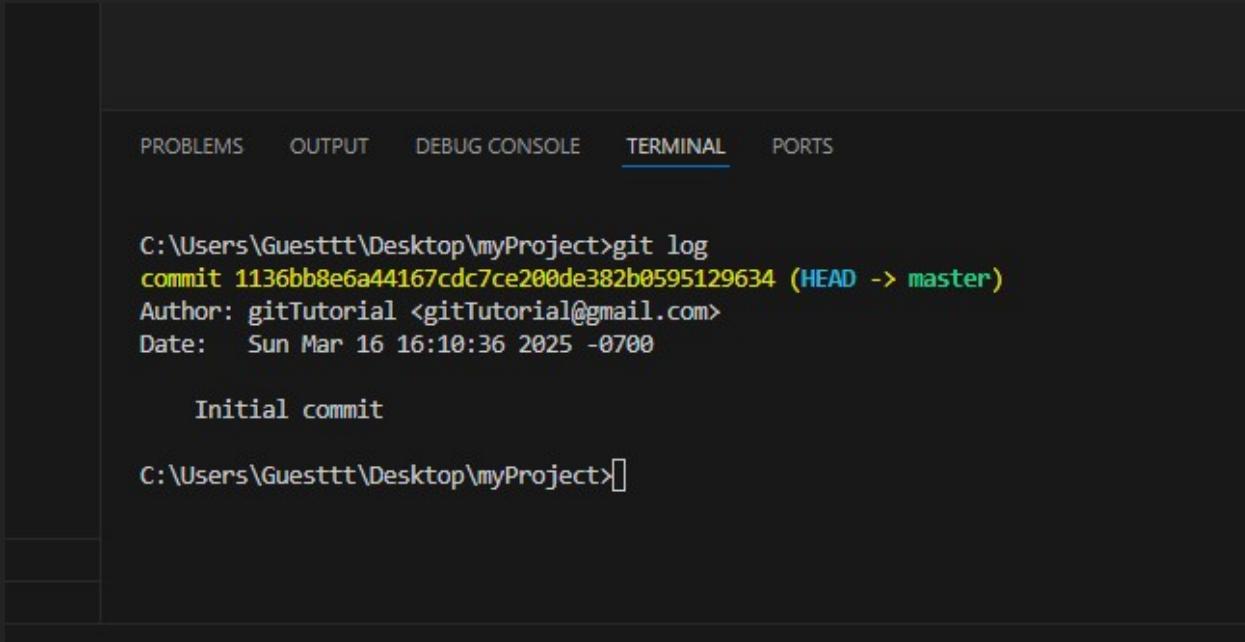
Figure 9. Screenshot showing ‘git commit’ command. Source: Abdirashid Ahmed (2025). 7

**Pro Tip:** Write clear and concise commit messages for understandable project history.

## 7. Viewing Commit History

- To view your commit history, type:

```
git log
```



The screenshot shows a terminal window with several tabs at the top: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined in blue), and PORTS. The terminal window displays the following command and its output:

```
C:\Users\Guesttt\Desktop\myProject>git log
commit 1136bb8e6a44167cdc7ce200de382b0595129634 (HEAD -> master)
Author: gitTutorial <gitTutorial@gmail.com>
Date:   Sun Mar 16 16:10:36 2025 -0700

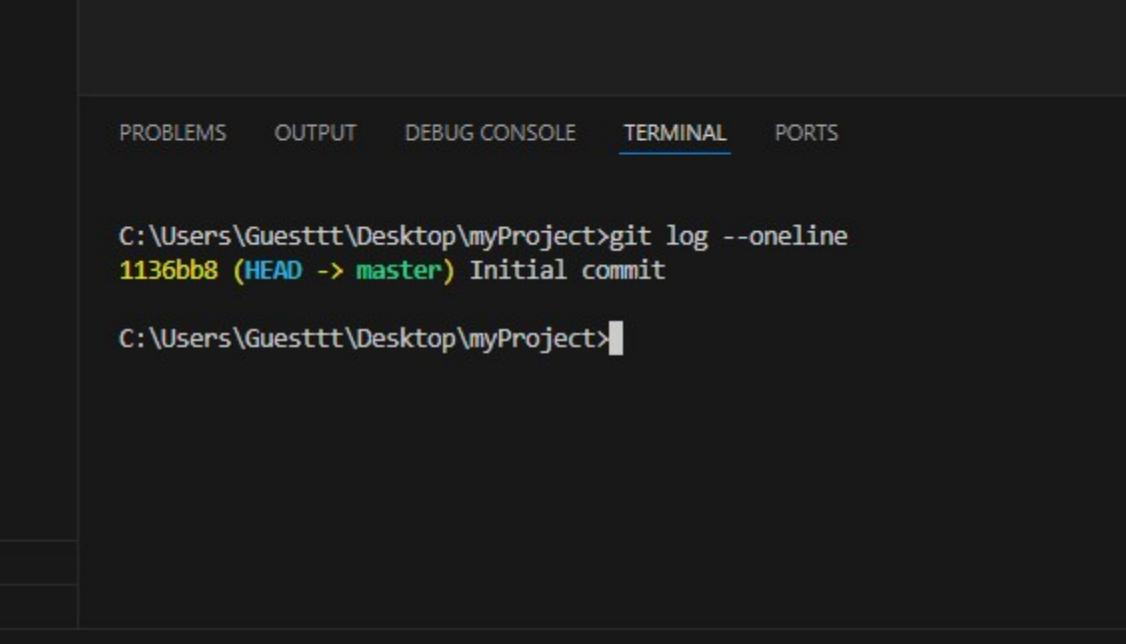
Initial commit

C:\Users\Guesttt\Desktop\myProject>[]
```

Figure 10. Screenshot showing ‘git log’ command. Source: Abdirashid Ahmed (2025).

- And for a shorter history view, you can also type:

```
git log --oneline
```



The screenshot shows a terminal window within a dark-themed IDE interface. The terminal tab is active, displaying the command 'git log --oneline' and its output. The output shows a single commit: '1136bb8 (HEAD -> master) Initial commit'. The terminal prompt 'C:\Users\Guesttt\Desktop\myProject>' is visible at the bottom.

Figure 11. Screenshot showing ‘git log --oneline’ command. Source: Abdirashid Ahmed (2025)

---

## 8. (Optional) Pushing to GitHub

If you’d like to continue building your skillset with Git, you can connect your saved stages to GitHub!

- First, make a new repository on GitHub:

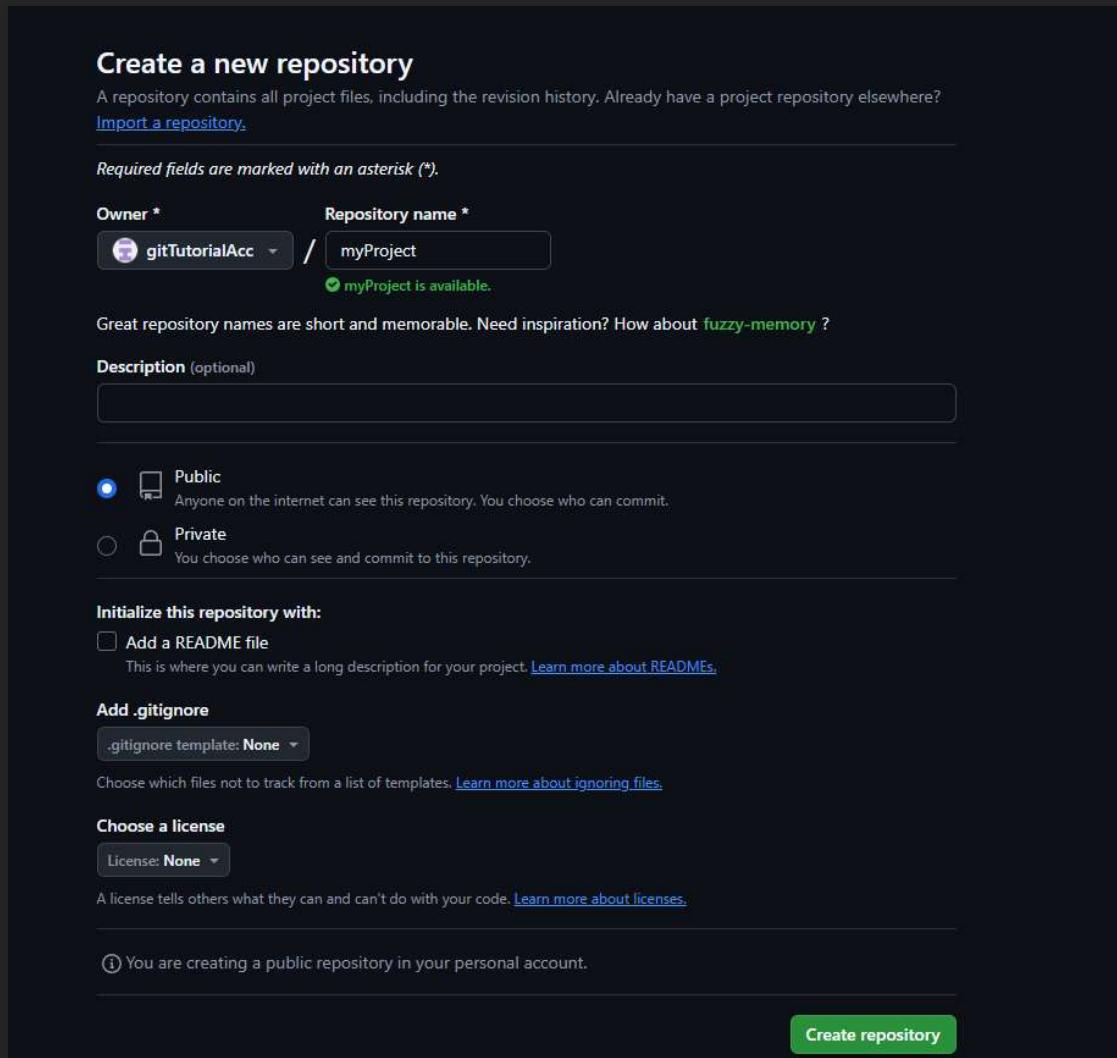


Figure 12. Screenshot showing new GitHub repository being made. Source: Abdirashid Ahmed (2025).

- From there, type or copy and paste the given text into your project's terminal:

```
git remote add origin "yourGitHubURL"
```

```
git branch -M main
```

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

**...or push an existing repository from the command line**

```
git remote add origin https://github.com/gitTutorialAcc/myProject.git  
git branch -M main  
git push -u origin main
```

Figure 13. Screenshot showing GitHub's push instructions. Source: Abdirashid Ahmed (2025).

- You may be asked to sign into GitHub once you paste and run the copied code:

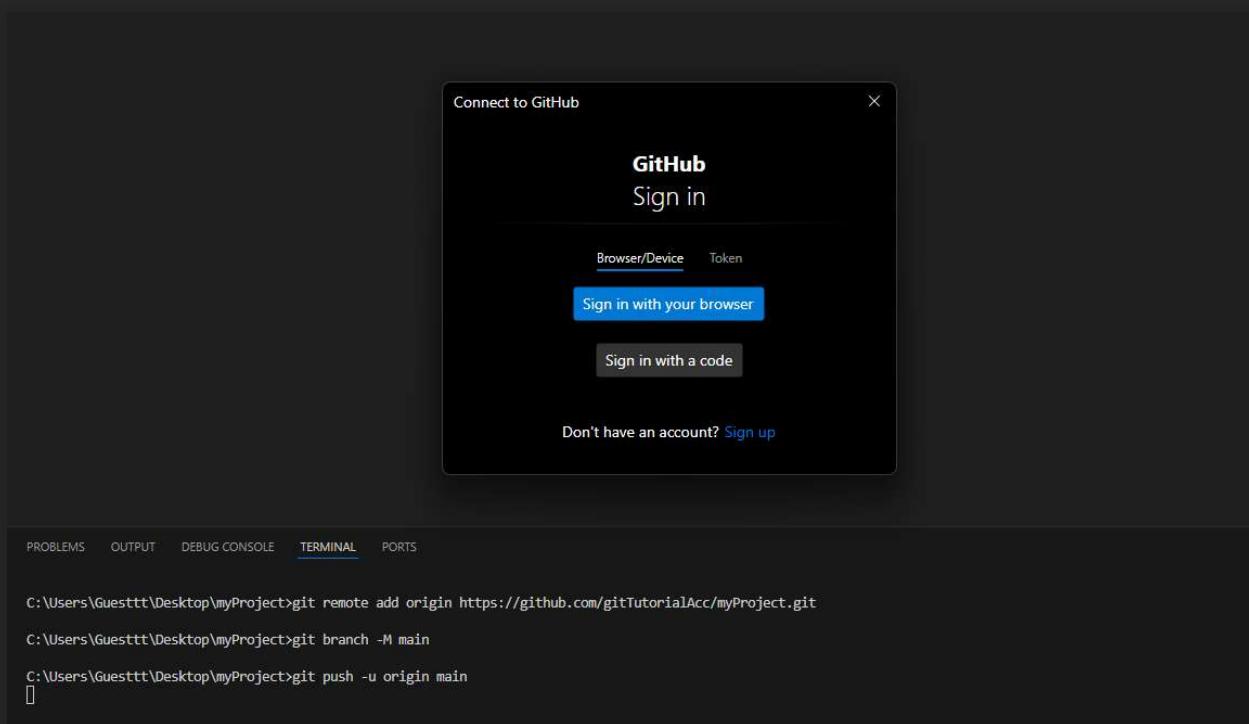
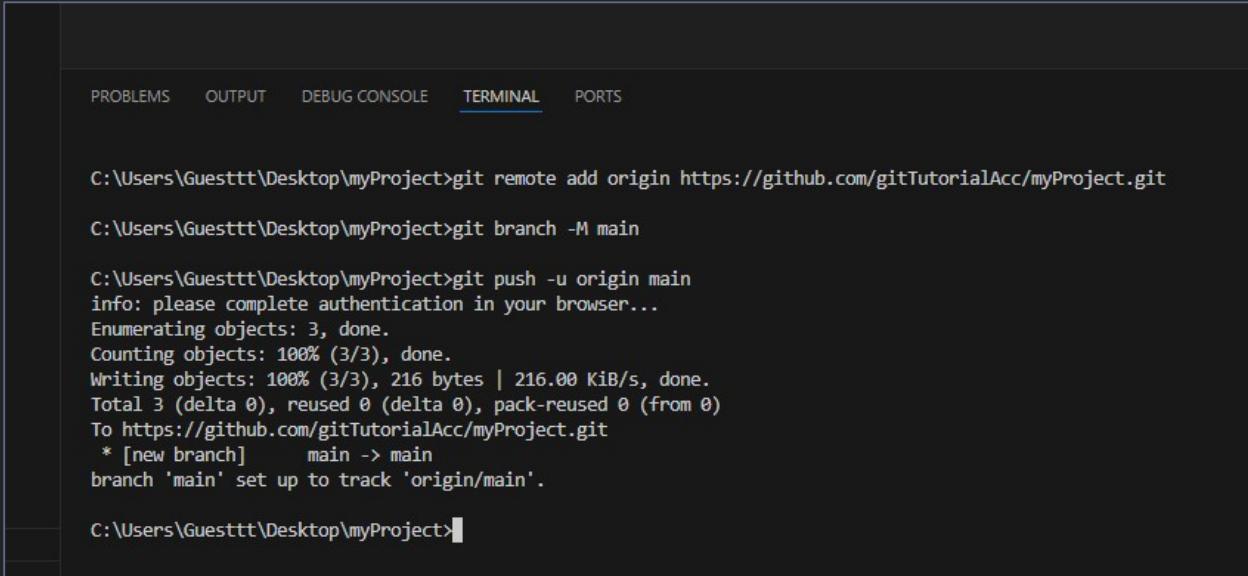


Figure 14. Screenshot showing GitHub Sign in request. Source: Abdirashid Ahmed (2025).

- After successfully signing in, the code should finish running successfully as well as finish connected to your GitHub repository:



The screenshot shows a terminal window with the following command history:

```
C:\Users\Guesttt\Desktop\myProject>git remote add origin https://github.com/gitTutorialAcc/myProject.git
C:\Users\Guesttt\Desktop\myProject>git branch -M main
C:\Users\Guesttt\Desktop\myProject>git push -u origin main
info: please complete authentication in your browser...
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 216 bytes | 216.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/gitTutorialAcc/myProject.git
 * [new branch]      main -> main
branch 'main' set up to track 'origin/main'.

C:\Users\Guesttt\Desktop\myProject>
```

Figure 15. Screenshot showing successful GitHub repository connection.

Source: Abdirashid Ahmed (2025).

To make sure you pushed accurately, go to your GitHub repository and see if your recent changes have been successfully pushed:

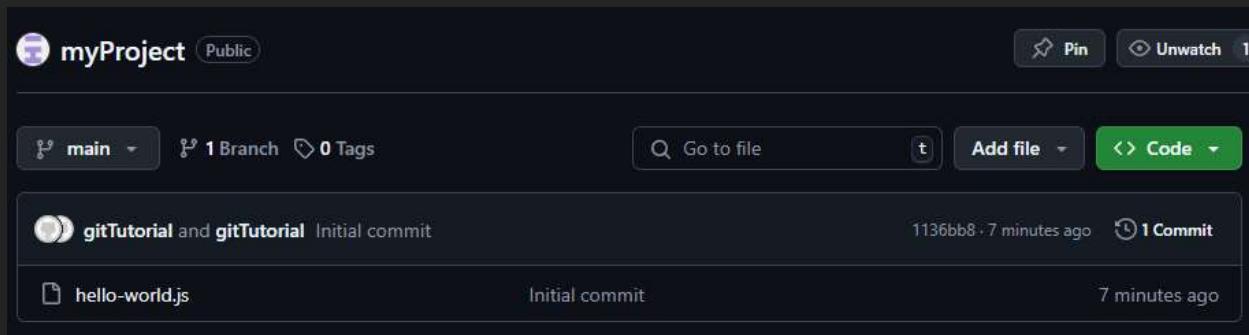


Figure 16. Screenshot showing successful push to GitHub. Source: Abdirashid Ahmed (2025).

**⚠ Warning:** Make sure you committed all changes before pushing.

## Quick Tips

- Use git status regularly to stay updated.
- Commit frequently to track better progress.
- Write clear commit messages (this can help your future-self understand what a certain save, or commit, was for).

Command	Description
<code>git init</code>	Makes a new Git repository.
<code>git config -global user.name “Your Name”</code>	Sets your name for commits.
<code>git config -global user.email “youremail@example.com”</code>	Sets your email for commits.
<code>git status</code>	Checks the status of your working project.
<code>git add .</code>	Stages all files for commit.
<code>git add yourfilename.txt</code>	Stages a specific file (for multiple, separate with spaces).
<code>git commit -m “Your commit message”</code>	Commits and describes update with a message.
<code>git remote add origin yourRepositoryURL</code>	Links your local repository to your remote GitHub repository.
<code>git branch -m main</code>	Renames the default branch to ‘main’.
<code>git push -u origin main</code>	Pushes your local commits to the remote "main" branch on GitHub.

# Resources

- [Official Git Documentation](#)
- [Git Cheat Sheet](#)

**Closing Statement:** Continue practicing these fundamental steps to become confident with Git. Remember, practice is what makes an expert!

---

*This guide was made with attention to accessibility in regard to the visual design and to produce an ease of understanding overall.*

## References

- Ahmed, A. (2025). *Successful project ‘git –version’ command* [Screenshot].
- Ahmed, A. (2025). *Successful project initialization commands* [Screenshot].
- Ahmed, A. (2025). *‘git status’ command* [Screenshot].
- Ahmed, A. (2025). *List of terminal options* [Screenshot].
- Ahmed, A. (2025). *‘hello world!’ JavaScript file* [Screenshot].
- Ahmed, A. (2025). *‘git status’ command again* [Screenshot].
- Ahmed, A. (2025). *‘git add’ command* [Screenshot].
- Ahmed, A. (2025). *‘git commit command* [Screenshot].
- Ahmed, A. (2025). *‘git log command* [Screenshot].
- Ahmed, A. (2025). *‘git log –oneline’ command* [Screenshot].
- Ahmed, A. (2025). *New GitHub repository* [Screenshot].
- Ahmed, A. (2025). *GitHub’s push instructions* [Screenshot].
- Ahmed, A. (2025). *GitHub Sign in request* [Screenshot].
- Ahmed, A. (2025). *Successful GitHub repository connection* [Screenshot].
- Ahmed, A. (2025). *Successful push to GitHub* [Screenshot].
- FutureLearn. (2022, October 25). What is Git and why is it worth learning about?  
<https://www.futurelearn.com/info/courses/collaborative-coding-with-git/0/steps/213616>
- GitHub Education. (n.d.). *Git cheat sheet*. Retrieved from <https://education.github.com/git-cheat-sheet-education.pdf>
- Git Documentation. (n.d.). *Git manual*. Retrieved from <https://git-scm.com/doc>