

## Git commands

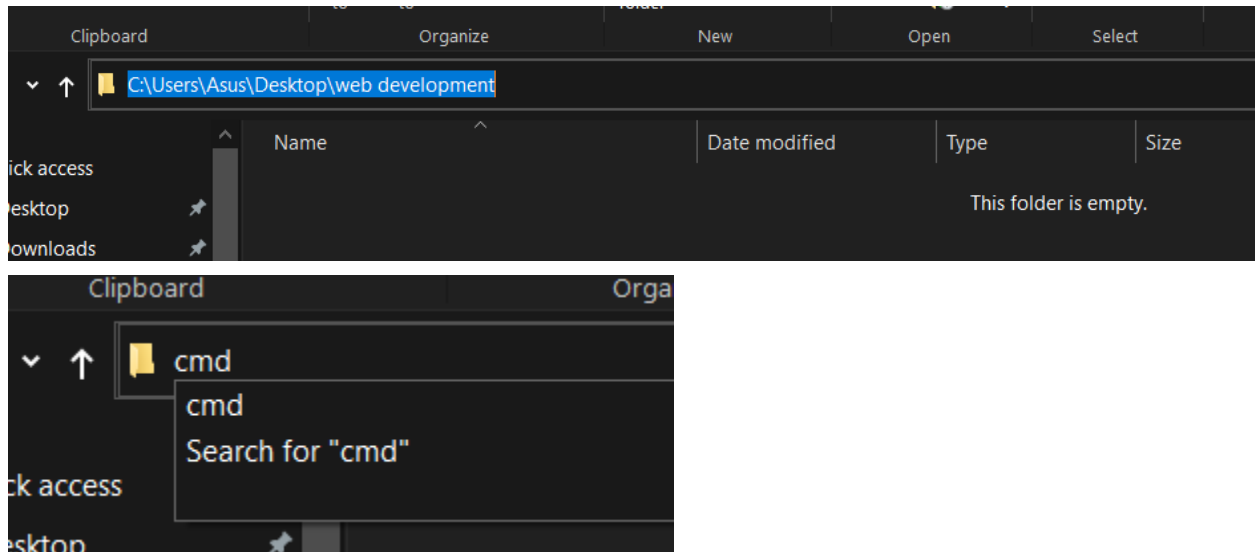
**Note:** Always navigate to the folder where you want to initialize git.

### FOR MAC OS USERS:

Follow the video to download git bash on your laptop

<https://youtu.be/ZM3I16Z-lxI>

Step 1: Open the folder navigate to the folder location and open **cmd** or right-click and select **git bash here**



Step 2: Check whether you have git or not by typing the following command

**git --version**

```
C:\Users\Asus\Desktop\web development>git --version
git version 2.28.0.windows.1
```

Your git version will appear if not use git bash here. If you have an error saying “git is not recognized as the internal and external command” then follow the following YOUTUBE video:

<https://youtu.be/v3RCp26naol>

This video will help you fix your problem.

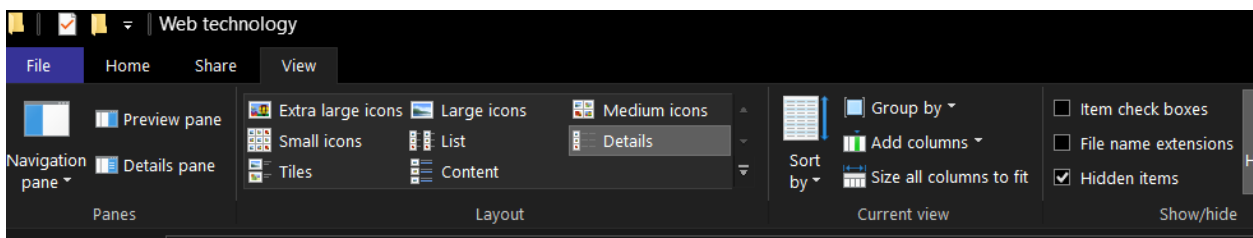
Step 3:

**git init**

```
C:\Users\Asus\Desktop\Web technology>git init
Initialized empty Git repository in C:/Users/Asus/Desktop/Web technology/.git/
```

Name	Date modified	Type	Size
.git	12/9/2022 8:48 AM	File folder	
Workshop1	12/4/2022 9:07 AM	File folder	
Workshop2	12/5/2022 8:24 AM	File folder	

It forms a **.git** folder in your folder. Since it is hidden you have to click in hidden items to see it.



Step 4:  
**git status**

```
C:\Users\Asus\Desktop\Web technology>git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Workshop1/

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

We run git status to know what files must be added to GitHub.

Step 5:  
**git add .**

```
C:\Users\Asus\Desktop\Web technology>git add .
```

Make sure you have kept a space between add and .

**git status**

```
C:\Users\Asus\Desktop\Web technology>git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
    new file:   Workshop1/.vscode/settings.json
    new file:   Workshop1/Bio.html
    new file:   Workshop1/about.html
    new file:   Workshop1/background.jpg
    new file:   Workshop1/bg.jpg
    new file:   Workshop1/contact.html
    new file:   Workshop1/home.css
    new file:   Workshop1/home.html
    new file:   Workshop1/index.css
    new file:   Workshop1/index.html
    new file:   Workshop1/script.js
    new file:   Workshop1/square-facebook.svg
    new file:   Workshop1/style.css
```

By rerunning the git status the **red mark** will turn **green** signaling that your files in the folder are ready to be sent to GitHub.

Step 6:

**git commit -m "Write a statement about what changes you made since you pushed the files since last time"**

```
C:\Users\Asus\Desktop\Web technology>git commit -m "My first portfolio website using HTML"
[master (root-commit) ac3c0d5] My first portfolio website using HTML
13 files changed, 378 insertions(+)
create mode 100644 Workshop1/.vscode/settings.json
create mode 100644 Workshop1/Bio.html
create mode 100644 Workshop1/about.html
create mode 100644 Workshop1/background.jpg
create mode 100644 Workshop1/bg.jpg
create mode 100644 Workshop1/contact.html
create mode 100644 Workshop1/home.css
create mode 100644 Workshop1/home.html
create mode 100644 Workshop1/index.css
create mode 100644 Workshop1/index.html
create mode 100644 Workshop1/script.js
create mode 100644 Workshop1/square-facebook.svg
create mode 100644 Workshop1/style.css
```

Adding the git commit -m “message” will commit the files to GitHub.

## Step 7: Make sure you have made a **git repository**

[Pull requests](#) [Issues](#) [Codespaces](#) [Marketplace](#) [Explore](#)

## Create a new repository

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

---

### Repository template


Start your repository with a template repository's contents.

No template ▾

---

Owner \*

Repository name \*


 jeshmi-hub ▾

 /


Great repository names are short and memorable. Need inspiration? How about [refactored-octo-waffle?](#)

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:**

Skip this step if you're importing an existing repository.


☐ Add a README file

Start your repository with a template repository's contents.

No template ▾

---


**Owner \*** **Repository name \***


 jeshmi-hub ▾ / Web ✓

Great repository names are Web is available. rable. Need inspiration? How about **refactored-octo-waffle**?

**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:**

Make sure you keep your repository **public**

- 
- ☒  **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:

Skip this step if you're importing an existing repository.

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

### Add .gitignore

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


.gitignore template: None ▼

### Choose a license

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

License: None ▼

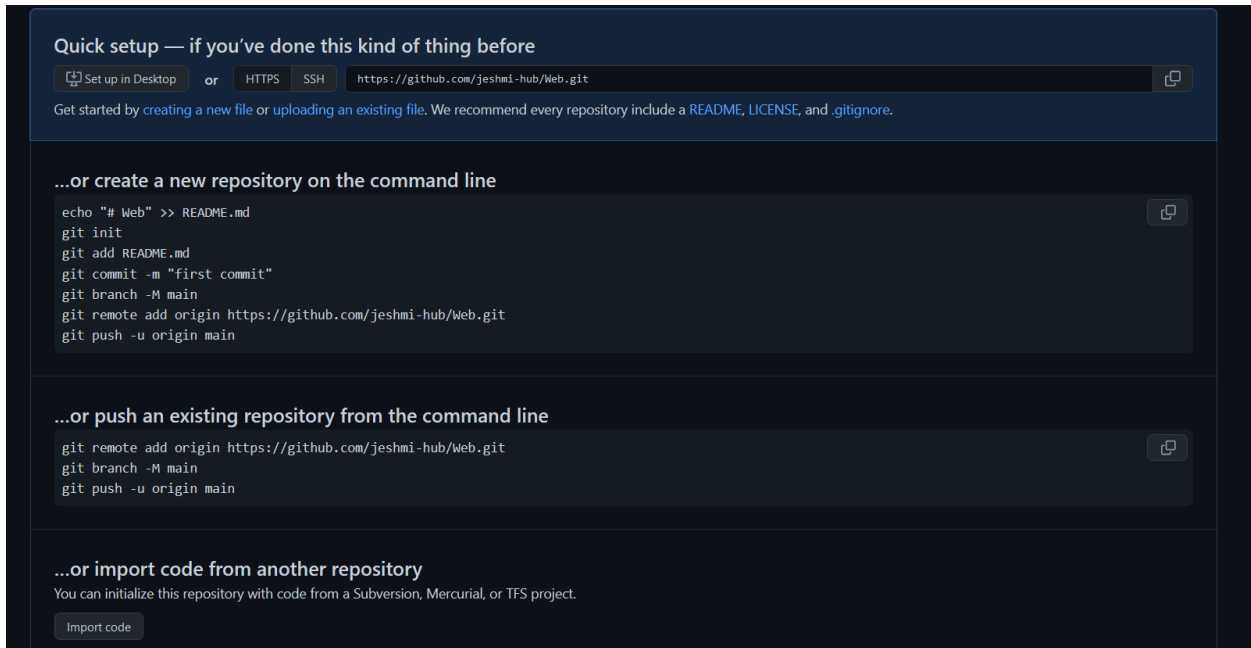
---

 You are creating a public repository in your personal account.

---

Create repository

Keep everything default and click on **Create repository**



Your repository is created.

Step 8: Copy the **git remote add origin URL** command and paste it into your terminal

```
C:\Users\Asus\Desktop\Web technology>git remote add origin https://github.com/jeshmi-hub/Web.git
```

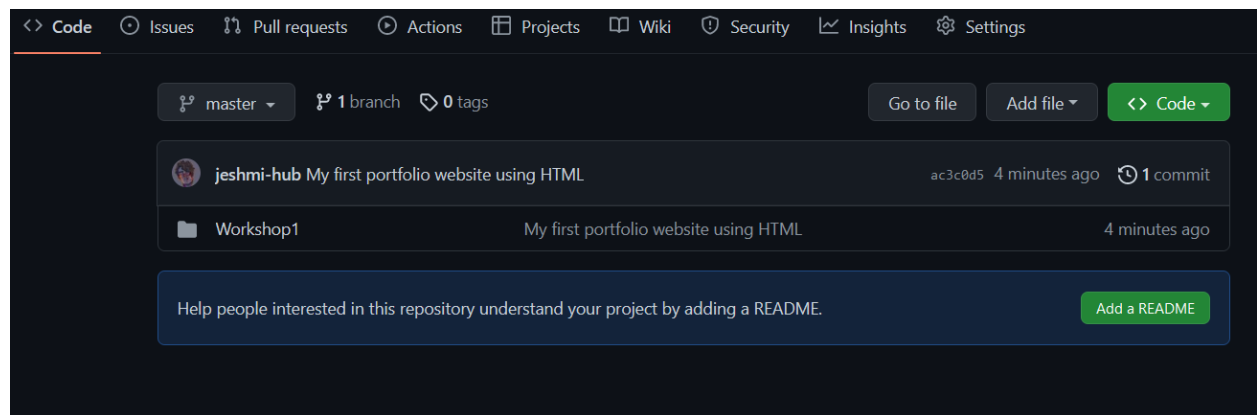
This command points to the repository that you have created in GitHub and where to send the files to.

Step 9: Type **git push -u origin master**

```
C:\Users\Asus\Desktop\Web technology>git push -u origin master
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (13/13), done.
Writing objects: 100% (17/17), 315.72 KiB | 10.52 MiB/s, done.
Total 17 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/jeshmi-hub/Web.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

If your message Enumerating appears be sure that your files are sent to the repository you made in GitHub.

Refresh your GitHub repository.



Your files are added to the GitHub repository.