How to GIT?

Contents

Step 1: Install Git

Go to the Git website: https://git-scm.com/downloads

Download: The site should detect your OS. Click the download link for your system.

Install:

Windows: Run the downloaded .exe installer.

CRITICAL: During installation, you'll be presented with many options. For beginners, the **default options are usually fine**.

Pay attention to the step about **"Adjusting your PATH environment"**. The recommended option (usually selected by default) is **"Git from the command line and also from 3rd-party software"**. This ensures you can use git commands in VS Code's terminal and the standard Command Prompt/PowerShell.

For the default editor, you can leave it as Vim (though it has a learning curve) or choose VS Code if it's listed.

For the default branch name, choose **"Let Git decide"** or **"Override the default branch name for new repositories"** and enter main. (main is the modern standard, replacing the older master).

Keep clicking "Next" through the remaining steps, accepting defaults unless you have a specific reason not to.

macOS:

If you have Xcode Command Line Tools installed, Git might already be there.
Open Terminal (Applications > Utilities > Terminal) and type git --version. If it shows a version, you're good.

If not, the easiest way is often to install the **Xcode Command Line Tools**. Open Terminal and type xcode-select --install. Follow the prompts.

Alternatively, use the installer downloaded from the Git website or install via Homebrew (if you have it): brew install git.

Verify Installation:

Open your terminal (Command Prompt, PowerShell, Git Bash on Windows; Terminal on macOS/Linux) and type:

```
git --version
```

You should see the installed Git version number. If you get an error, the installation might not have completed correctly, or Git wasn't added to your system's PATH.

Step 2: Create a GitHub Account

1. **Go to GitHub:** https://github.com/
2. **Sign Up:** Click "Sign up" and follow the instructions to create a free account. You'll need a username, email address, and password. Verify your email address.

Step 3: Configure Git Locally

Before you start using Git, you need to tell it your name and email address. This information will be attached to every change (commit) you make. Use the *same email address* you used for your GitHub account.

1. **Open your terminal** (Command Prompt, PowerShell, Git Bash, or Terminal).
2. **Set your username:**

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

3. **Set your email address:**

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

(Replace "your.email@example.com" with the email you used for GitHub).

Step 4: Create Your First Project Locally

1. **Create a Project Folder:** Choose a location on your computer where you want to store your projects (e.g., Documents/Projects or C:\Users\YourUser\Projects). Create a new folder for this specific project. Let's call it my-first-website

You can do this through your **file explorer** or using the **terminal**:
Terminal Commands # Navigate to where you want to store projects (example)
# Create the new project folder

```
cd Documents/Projects
```

# Move into the new folder

```
mkdir my-first-website
```

#open your folder

```
cd my-first-website
```

2. **Open the Folder in VS Code:**
   - Open VS Code.
   - Go to File > Open Folder....
   - Navigate to and select the my-first-website folder you just created. Click "Open" or "Select Folder".

3. **Create an HTML File:**
   - In the VS Code Explorer panel (usually on the left), right-click on the empty space under the MY-FIRST-WEBSITE folder name.
   - Select New File.
   - Name the file index.html and press Enter.

4. **Add Basic HTML5 Content:**
   - Click inside the index.html file editor.
   - Type ! (an exclamation mark) and press Tab or Enter. VS Code's built-in Emmet tool will expand this into a basic HTML5 structure.
   - Modify it slightly to add some content:

5. **Save the file:** Press Ctrl+S (Windows/Linux) or Cmd+S (macOS).

Step 5: Initialize Git in Your Project

Now, you'll tell Git to start tracking changes within your my-first-website folder.

1. **Open the VS Code Terminal:**
   - Go to the top menu in VS Code: Terminal > New Terminal.
   - A terminal panel will open at the bottom of VS Code, already navigated to your project folder (my-first-website).

2. **Initialize Git:** Type the following command in the VS Code terminal and press Enter:

   ```
   git init
   ```

You should see output like Initialized empty Git repository in **/path/to/your/my-first-website/.git/**. This creates a hidden .git subfolder where Git stores all its tracking information.

Step 6 Create a Repository on GitHub

Now you need a place on GitHub to store your code remotely.

1. **Go to GitHub:** Log in to your GitHub account.
2. **Create a New Repository:**
   - Click the + icon in the top-right corner and select New repository.
   - **Repository name:** Give it a name. It's often good practice to use the same name as your local folder, so enter my-first-website.
   - **Description:** (Optional) Add a brief description.
   - **Public/Private:** Choose Public (anyone can see it, you choose who can contribute) or Private (you choose who can see and contribute). For learning, Public is fine.
   - **IMPORTANT: Do NOT** initialize the repository with a README, .gitignore, or license *at this stage*. Since you already have your index.html file locally, you want to push that existing content first. Initializing here would create files on GitHub that don't exist locally yet, leading to conflicts later.
3. **Click Create repository**.

Step 7: Connect Your Local Repository to GitHub

GitHub will now show you a page with instructions. We want the section under "...or push an existing repository from the command line".

1. **Copy the Remote URL:** Find the line that looks like this (use the HTTPS version for simplicity):

   ```
   git remote add origin https://github.com/YourUsername/my-first-website.git
   ```

Replace YourUsername with your actual GitHub username.

2. **Add the Remote:** Paste this command into your VS Code terminal and press Enter. This command tells your local Git repository where the remote version (origin) lives on GitHub.

   ```
   git remote add origin https://github.com/YourUsername/my-first-website.git
   ```

3. **(Optional but Recommended) Rename the default branch to main:** If your Git default branch isn't already main, run this:

   ```
   git branch -M main
   ```

This ensures your local main branch matches the standard name used on GitHub.

Step 8: Stage, Commit, and Push Your Code

This is the core Git workflow: telling Git which changes you want to save, saving them with a message, and sending those saved changes to GitHub.

1. **Stage Changes:** Tell Git you want to include the index.html file in your next save point (commit).

```
git add index.html
```

Alternatively, to add *all* new or modified files in the current folder and subfolders, you can use:

```
git add .
```

(Note the dot .)

2. **Commit Changes:** Save the staged changes with a descriptive message.

```
git commit -m "Initial commit: Add basic index.html structure"
```

- The -m flag allows you to write the commit message directly. Good messages explain *what* changed and *why*.

3. **Push to GitHub:** Send your committed changes from your local main branch to the origin remote (GitHub).

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

- -u sets the upstream link. You only need to do this the *first* time you push this branch. Future pushes for this branch can just use **git push**.
- You might be prompted to enter your GitHub username and password (or a Personal Access Token if you have 2FA enabled - GitHub will guide you if needed).

Step 9: Verify on GitHub

1. **Go back to your GitHub repository page** in your web browser (the one for `my-first-website`).
2. **Refresh the page.**
3. You should now see your `index.html` file listed there!

Congratulations! You have successfully:

1. Installed VS Code and Git.
2. Created a GitHub account.

3. Configured Git locally.
4. Created a local project folder with an `index.html` file.
5. Initialized a Git repository in that folder.
6. Created a corresponding repository on GitHub.
7. Connected the local and remote repositories.
8. Made your first commit and pushed it to GitHub.

Your Ongoing Workflow:

Now, whenever you make changes to your project:

1. **Edit Code:** Modify `index.html` or add new files (e.g., `style.css`) in VS Code.
2. **Save Files:** (`Ctrl+S` / `Cmd+S`).
3. **Stage Changes:** Open the VS Code terminal (`Terminal` > `New Terminal`) and type:

```
git add .  # Or specify files: git add index.html style.css
```

4. **Commit Changes:**

```
git commit -m "Describe your changes here (e.g., Add paragraph, style heading)"
```

5. **Push Changes:**

```
git push
```

(You don't need `-u origin main` after the first time for this branch).

Keep practicing this Add -> Commit -> Push cycle! It's the foundation of using Git. Good luck with your HTML5 coding!