

How to Set Up Git and Sign Up for GitHub

This tutorial will guide you through downloading and installing Git on Windows and Mac (Intel or M1/M2), as well as signing up for GitHub.

What is Git?

Git is a tool that helps you keep track of changes you make to your files. It allows you to save different versions of your work, so you can go back and see what changes you've made or revert to an earlier version if needed.

What is GitHub?

GitHub is a website where you can store your projects online and share them with others. You can use **GitHub** to collaborate with others on projects, track issues, and host your project's documentation and website.

What is a Git Credential Manager?

A **Git Credential Manager** helps you securely save your username and password when you are working with online repositories (like GitHub). This way, you don't have to enter your credentials every time you perform an action like saving your changes or downloading a project.

1. Sign Up for GitHub

A. Visit the GitHub Website:

- Open your web browser and go to [GitHub](https://github.com).

B. Create an Account:

- Click on the "Sign up" button in the top-right corner.
- Enter your email address and click "Continue".
- Create a password, choose a username, and follow the prompts to complete the sign-up process.
- Verify your email address by clicking the link sent to your email.

2. Installing Git on Windows (skip this step if you completed it in class)

A. Download Git for Windows:

- Go to the [Git for Windows](https://git-scm.com/download/win) website.
- Click on the "Download" button to download the installer.

B. Install Git:

- Run the downloaded installer and follow the prompts.
- I suggest that you use the default options with one exception:
 - In the **Adjusting the name of the initial branch in new repositories pane** choose the **Override the default branch name for new repositories** with `main` in the text field.

3. Installing Git on Mac (Intel or M1/M2)

A. If you are using a Mac you already have Git.

4. Installing the GitHub Credential Manager

- A. Windows: The **Git for Windows** program already has this functionality built-in.
- B. For Mac, you'll need to know if you have an M1/M2 chip or an Intel chip. Click on the Apple icon in the top-left of your screen, then **"About This Mac."** Look at the **"Chip"** or **"Processor"** entry for "M1", "M2", "Intel" or related terms.
- C. Mac with M1/M2 chips: click on [this link](#) to download the Credential Manager. Install the disk image.
- D. Mac with Intel chip: click on [this link](#) to download the Credential Manager. Install the disk image.

5. Configure Git with Your GitHub Account

A. Set Up Your Git Username and Email:

- Open Git BASH (Windows) or Terminal (Mac).
- Configure your Git username and email:

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

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

B. Authenticate with GitHub (**this will happen the first time we use Git**):

- When you perform a Git operation that requires authentication (e.g., cloning a repository), you will be prompted to enter your GitHub username and password.
- Git Credential Manager will save your credentials securely, so you won't need to enter them every time.

Additional Tips

- **Basic Git Commands:**

- **Clone a Repository:**

```
git clone [insert link to github repository]
```

*Explanation: * This command creates a copy of a project from GitHub on your local machine. Essentially, you are downloading a project so you can work on it.

- **Check the Status:**

```
git status
```

*Explanation: * This command shows the current state of your project. It tells you which changes have been staged, which files are being tracked, and which are not. It's like a quick overview of what's going on with your project.

- **Add changes:**

```
git add .
```

*Explanation: * This command stages all changes you've made for the next commit. Staging is like preparing your changes to be saved.

- **Commit changes:**

```
git commit -m "Your commit message"
```

*Explanation: * This command saves your staged changes with a message describing what you did. Committing is like making a permanent record of your changes. The message helps you and others understand what was changed and why.

- **Push Changes:**

```
git push
```

*Explanation: * This command uploads your committed changes to GitHub. It's like sending your saved work to the cloud so others can see and collaborate with you.

- **Learning Resources:**

- [Git Documentation](#)
- [GitHub Skills](#)