

How to use git

Zric

January 8, 2026

Abstract

This document provides a brief introduction to using Git, for new learners. It covers basic commands and setup instructions to help users get started with Git for version control in their projects. I also include some of my personal tips for privacy protection when using GitHub if you don't want to expose your personal information.

Contents

1	Introduction	1
2	Basic Git Knowledge	1
2.1	Installation	1
2.2	Setting up Git	2
2.3	Pull and Push	2

1 Introduction

Git is a distributed version control system that allows multiple people to work on a project simultaneously without interfering with each other's changes. It is widely used in software development for tracking changes in source code during software development.

2 Basic Git Knowledge

2.1 Installation

For windows users, download the installer from <https://git-scm.com/download/win> and follow the installation instructions. For macOS users, you can install Git using Homebrew with the command:

```
1 brew install git
```

Listing 1: Installing Git on macOS

If successful, you can check the installed Git version with in the command:

```
1 git --version
```

Listing 2: Checking Git Version

2.2 Setting up Git

Commands to set up Git on your local machine:

```
1 git config --global user.name "Your_Name"  
2 git config --global user.email "Your_email"
```

Listing 3: Setting up Git

For my personal setup, because I don't want others to know who I am, I first use my private email to sign up a GitHub account, then set my email private in GitHub settings, GitHub will then give you a no-reply email address to commit (you can find it in the email settings page), usually like this:

```
1 ID+username@users.noreply.github.com
```

Listing 4: GitHub no-reply email example

This is actually not a real email address, but GitHub will recognize it when you commit through Git and link the commit to your "fake" account. If you unfortunately exposed your real email address in previous commits in one repository and want to hide your real email address, you can use the following command to rewrite the commit history:

```
1 git filter-branch -f --env-filter '  
2     GIT_AUTHOR_NAME=$(git config user.name)  
3     GIT_AUTHOR_EMAIL=$(git config user.email)  
4     GIT_COMMITTER_NAME=$(git config user.name)  
5     GIT_COMMITTER_EMAIL=$(git config user.email)  
6 ' -- --all
```

Listing 5: Rewriting commit history to hide email

2.3 Pull and Push

To clone a repository from GitHub to your local machine, use the command:

```
1 git clone name_of_repository
```

Listing 6: Cloning a repository

To pull the latest changes from the remote repository to your local repository, use:

```
1 git pull origin branch_name
```

Listing 7: Pulling changes from remote repository