

Exp No. 1	INSTALLATION OF GIT AND CREATING REPOSITORIES.	

Aim

The aim is to demonstrate pushing changes to a GitHub repository, covering setup, managing visibility, and executing basic Git operations.

Description

The aim of the experiment is to demonstrate the process of pushing changes to a GitHub repository. This involves:

1. Setting up Git: Ensure Git is installed on your computer.
2. Managing Repository Visibility: Understand the effects of making a repository private or public on GitHub.
3. Changing Repository Visibility: Learn how to change the visibility of a repository on GitHub.
4. Pushing Changes to the Repository: Make changes to a local repository, commit them, and push them to the remote repository on GitHub.

Introduction of GIT

Git is one of the ways of implementing the idea of version control. It is Distributed Version Control System.

Installing GIT

Before you start using Git, you have to make it available on your computer.

it's already installed, it's probably a good idea to update to the latest version. You can either install it as a package or via another installer, or download the source code and compile it yourself.

Installing on Windows

There are also a few ways to install Git on Windows. The most official build is available for download on the Git website. Just go to <https://git-scm.com/download/win> and the download will start automatically

To get an automated installation you can use the [Git Chocolatey package](#).

The easiest way to get Git is to download the executable from the Git website.

Click "64-bit Git for Windows Setup" to start the download, and then wait a moment — the download is only about 50 megabytes, so it shouldn't take very long.

Download for Windows

[Click here to download](#) the latest (2.37.3) **64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **12 days ago**, on 2022-08-30.

Other Git for Windows downloads

Standalone Installer

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

Portable ("thumbdrive edition")

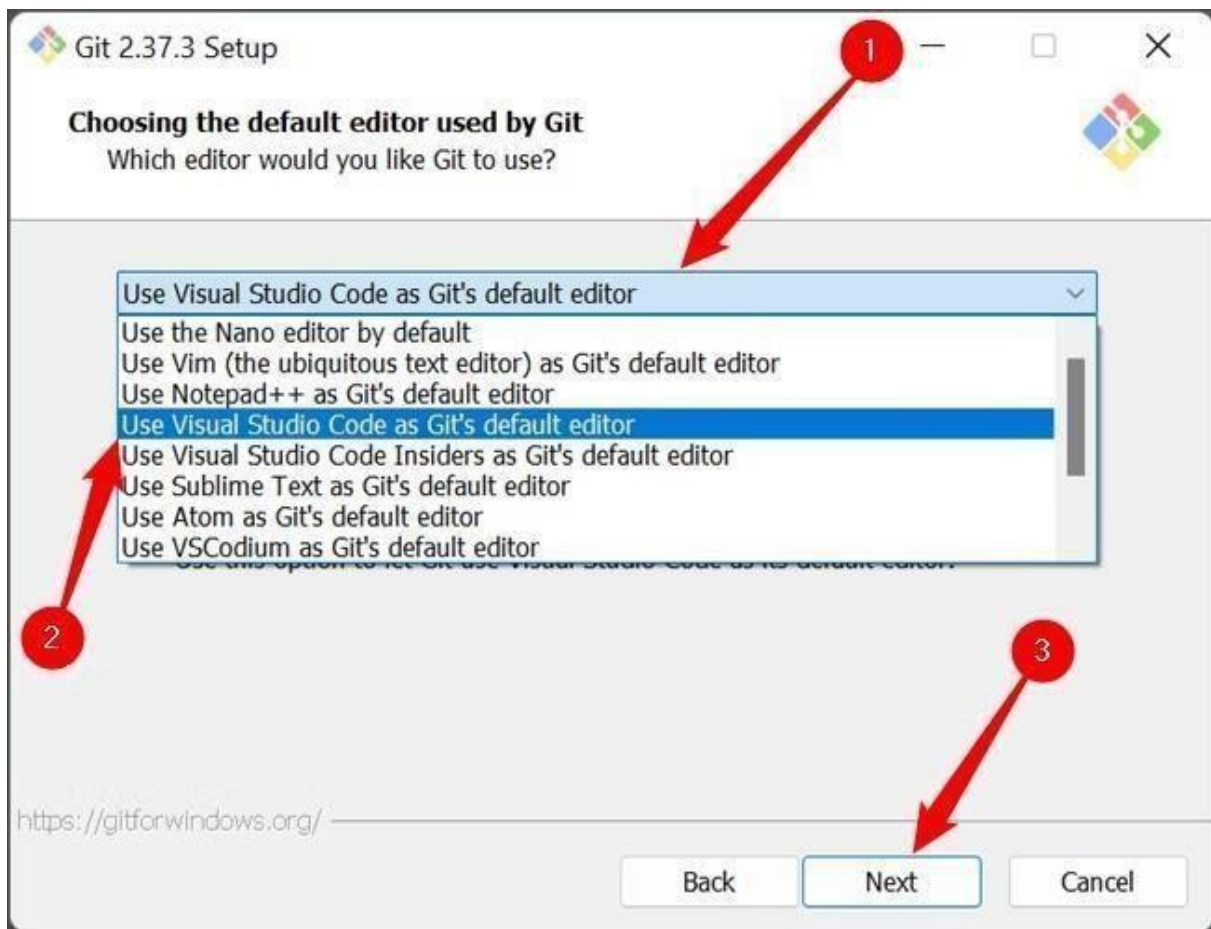
[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

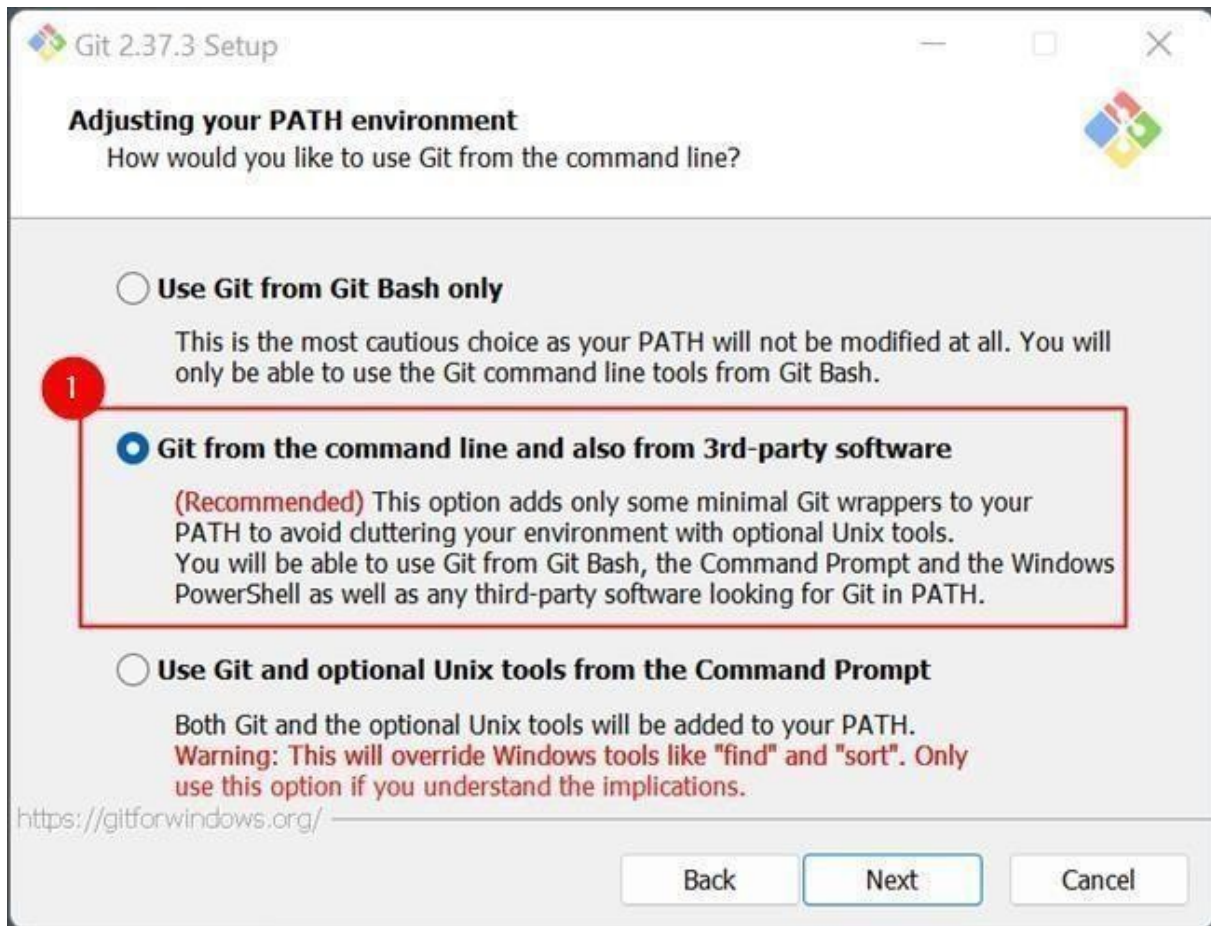
Double-click the executable you just downloaded, then click "Next" to move through the installation prompts.

The first is the text editor Git will use. The default selection is Vim. Vim is ubiquitous and a hallmark of command-line interfaces everywhere but learning to use its idiosyncratic

commands can be daunting. You should probably pick something else instead, like Visual Studio Code, Sublime, NotePad++, or any other plain text editor you like.



The second is the way Git integrates itself into your PC's PATH. Make sure that the "Git From The Command Line And Also From 3rd-Party Software" is selected.



Click through the remaining options, and wait for everything to finish downloading. The time requires to download everything will vary depending on what you chose to install. The default selection results in a download that is about 270 megabytes.

Managing Private and Public Repository

Making a repository Private

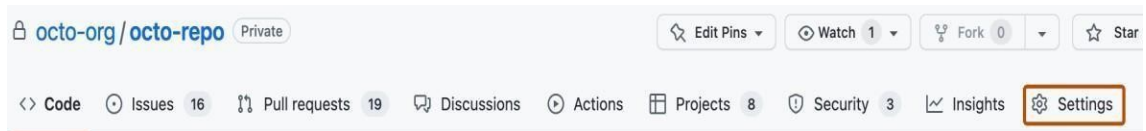
- GitHub will detach public forks of the public repository and put them into a new network. Public forks are not made private.
- If you're using GitHub Free for personal accounts or organizations, some features won't be available in the repository after you change the visibility to private. Any published GitHub Pages site will be automatically unpublished. If you added a custom domain to the GitHub Pages site, you should remove or update your DNS records before making the repository private, to avoid the risk of a domain takeover.
- GitHub will no longer include the repository in the GitHub Archive Program.
- GitHub Advanced Security features, such as code scanning, will stop working.

Making a repository Public

- GitHub will detach private forks and turn them into a standalone private repository.
- If you're converting your private repository to a public repository as part of a move toward creating an open source project.
- Once your repository is public, you can also view your repository's community profile to see whether your project meets best practices for supporting contributors.
- The repository will automatically gain access to GitHub Advanced Security features.

Changing a repository's Visibility

1. On GitHub.com, navigate to the main page of the repository.
2. Under your repository name, click **Settings**. If you cannot see the "Settings" tab, select the dropdown menu, then click **Settings**.



3. In the "Danger Zone" section, to the right of to "Change repository visibility", click **Change visibility**.
4. Select a visibility.
5. To verify that you're changing the correct repository's visibility, type the name of the repository you want to change the visibility of.
6. Click **I understand, change repository visibility**.

Git Repository:

The screenshot shows the GitHub profile page for user 'luepiga'. The profile picture is a green and white pixelated logo. The page includes a navigation bar with 'Overview', 'Repositories' (1), 'Projects', 'Packages', and 'Stars'. Below the profile picture, there's a section for 'Popular repositories' showing a repository named 'luepi-' with a 'Public' label. To the right, there's a link to 'Customize your pins'. Below this, a section titled '11 contributions in the last year' shows a calendar grid for the year 2024. The grid shows contributions on specific days: Dec 1, Jan 1, Jan 2, Jan 3, Jan 4, Jan 5, Jan 6, Jan 7, Jan 8, Jan 9, Jan 10, Jan 11, Jan 12, Jan 13, Jan 14, Jan 15, Jan 16, Jan 17, Jan 18, Jan 19, Jan 20, Jan 21, Jan 22, Jan 23, Jan 24, Jan 25, Jan 26, Jan 27, Jan 28, Jan 29, Jan 30, Jan 31, Feb 1, Feb 2, Feb 3, Feb 4, Feb 5, Feb 6, Feb 7, Feb 8, Feb 9, Feb 10, Feb 11, Feb 12, Feb 13, Feb 14, Feb 15, Feb 16, Feb 17, Feb 18, Feb 19, Feb 20, Feb 21, Feb 22, Feb 23, Feb 24, Feb 25, Feb 26, Feb 27, Feb 28, Feb 29, Mar 1, Mar 2, Mar 3, Mar 4, Mar 5, Mar 6, Mar 7, Mar 8, Mar 9, Mar 10, Mar 11, Mar 12, Mar 13, Mar 14, Mar 15, Mar 16, Mar 17, Mar 18, Mar 19, Mar 20, Mar 21, Mar 22, Mar 23, Mar 24, Mar 25, Mar 26, Mar 27, Mar 28, Mar 29, Mar 30, Mar 31. Below the calendar, there's a link to 'Learn how we count contributions' and a 'Less' button. To the right of the calendar, there's a 'Contribution settings' dropdown and a '2024' button. Below the calendar, there's a 'Contribution activity' section for 'April 2024' with a message: 'Luepiga has no activity yet for this period.'

Pushing changes to the repository

1. On Github.com, ensure your repository is public.
2. Clone the repository to your local machine.
 - i. Open a terminal on your local machine.
 - ii. Use the following command to clone the repository to your local machine. Replace 'your-username' and 'your-repo-name' with your GitHub username and repository name
`git clone https://github.com/your-username/your-repo-name.git`
 - iii. Move into the newly cloned repository
`cd your-repo-name`
3. Make a change to the README file (Create a README.md file if it does not exist).
4. Commit the change.
 - i. Add the modified README file to the staging area using the following command:
`git add README.md`
 - ii. Commit the changes with a meaningful commit message:
`git commit -m "Update README file"`
5. Push it to the GitHub repository using the following command. If your main branch is called 'master', then replace 'main' with 'master'.
`git push origin main`

SAMPLE OUTPUT:

```
Your branch is up to date with 'origin/main'
Mailings
Review
View
Paragreph
Search
Help
Normal
Untracked files:
(use "git add <file>..." to include in what #will be committed)
text.md
nothing added to commit but untracked files present (use "git add to track)
PS D:\random\RANDOM> git add text.md
PS D:\random\RANDOM> git commit -m "first commit"
[main 2b78748] first commit
1 file changed, insertions(+), 0 deletions(-)
create mode 100644 text.md
PS D:\random\RANDOM> git push
Enumerating `objects: 4, done.
Gounting objects: 100% (4/4), done:
Delta compression using up to 4 threads
Gompressing objects: 100% (2/2), done.
Mriting objects: 100% (3/3), 310 bytes | 310.00 KiB/s, done.
Total 3 (delta 0), reused 1 (delta 0), pack-reused 0
To https://github.com/URK23CS1083/RANDOM.git
26b126e..2b78748 main -> main
PS D:\random\RANDOM> git branch -m TEMP
ps D: \random\RANDOM> git branch
TEMP
```