# An Intro to Git and GitHub for Beginners

## What is Git?

Git is the most popular version control system. Version control stores data about which changes were made, who made them, when were thy made and why they were needed. Version control also makes it possible to restore previous versions if something goes wrong. There’s a further advantage that changes are easy to document.

Git stores all information locally to its own storage that is called repository. Git commands are usually made using command line.

## What is GitHub?

GitHub is a cloud-based platform for storing and sharing code and collaborating on a shared project. GitHub is a remote repository with a graphical, web-based user interface which is easy to share with others.

## Install and Configure Git

Install Git to your computer from <https://git-scm.com/download>. Verify the installation by typing **git –version** in your terminal.

The user’s name and email address are saved with every change, so set your Git username and email after installation.

git config --global user.name ”Your Name”

git config --global user.email ”[your.email@example.com](mailto:your.email@example.com)”

## Create a GitHub account

Go to [www.github.com](http://www.github.com/) an register an account. You will be asked to verify your email address. The account can be either public or private.

## Create your first local repository

If you are using Windows, open Bash command prompt in the folder that you want to make a repository from. In Linux, you can just use the default terminal.

Enter the following commands

git init

git add .

git commit -m ”First commit”

## Move your changes from your own repository to remote repository

Git remote add origin https://github.com/user/example.git

git push origin master

### Branches

Branches are a good way to keep different phases of a project in different places, yet in the same repository. Ready, tested parts of the application should be kept in master branch. Parts being developed can be stored for example in a branch called development. Additionally, you can make more branches for less important components or ideas. This way you avoid dependencies between parts before you decide to merge the branch to master branch. Remember to commit your changes before switching branches.

## Some common Git commands

git add . Compares current directory to local repository and moves all changes to staged state, which means the are now ready to commit into local repository.

git branch testing – Creates a new branch called ”testing”.

git checkout <branch name> switch branches (In modern Git versions, git switch is preferred for switching branches.)

git checkout is a command used to switch branches or restore files in your Git repository.

git clone <repository-url> - Clones the remote repository to local repository.

git commit Stores files that are in staged state into local repository.

git config --global user.email <email> Sets user’s email address, which will be seen in user’s commits.

git config --global user.name <username> Sets user name, which will be seen in user’s commits.

git fetch <name of the remote repo> git fetch updates your local repository with the latest changes from the remote repository, including new branches or commits, but it doesn’t merge or modify your working directory.

git init Creates a local empty repository.

git merge <branch name> combines the changes from one branch into your current branch. It integrates the commit history of the two branches, creating a new merge commit if necessary.

git pull <name of the remote repo> git pull fetches the latest changes from the remote repository and automatically merges them into your current branch. It's equivalent to running git fetch followed by git merge.

git push - git push uploads your local branch's commits to a remote repository, updating the corresponding branch on the remote. It's how you share changes with others.

git reset – Cancels staged state (git add)

git revert <commit> - Reverses changes and creates a new commit which takes us back to an earlier commit. Git revert commit HEAD takes us back to the previous commit.

git rm <file name> Removes a file and sets the removal in staged state. You’ll have to commit it for the removal to be effective. If you want to remove a directory, use selector -r: git rm -r <folder name>

git status – Shows files that are in staged state.

### Sources

Freecodecamp 2022. Introduction to Git and GitHub for Beginners. URL: <https://www.freecodecamp.org/news/introduction-to-git-and-github/> Accessed: 19.1.2025

W3Schools.com. Git and GitHub Introduction. URL: <https://www.w3schools.com/git/git_intro.asp?remote=github> Accessed 19.1.2025

Ruonavaara, M. 2020. Git 101. Git-opas Haaga-Helian opiskelijoille. URL: <https://github.com/mruonavaara/git101> Accessed: 19.1.2025.

GitHub. About GitHub and Git. Luettavissa: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git> Accessed: 19.1.2025.

Atlassian. URL: <https://atlassian.com/git> Accessed: 19.1.2025.

Github docs. Start Your Journey. URL: <https://docs.github.com/en/get-started/start-your-journey/about-github-and-git> Accessed: 19.1.2025.