

Git and Github

Rashidul Islam Rahul
rashidul.rahul@gmail.com

Whatsapp: 01623397637

Acting Backend Lead

Goama Games



distributed version-control system for
tracking changes in source code

History

Birth in 2005

Linus Torvalds and Linux community



Setup

macOS:

brew install git

Ubuntu/Debian:

sudo apt update

sudo apt install git

Windows:

download and run the installer from

<https://git-scm.com>

Simple config

Configure your identity

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

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

Optional preferences

```
git config --global pull.rebase false           # merge on pull
```

```
git config --list
```



Git Tools & Terminology

Repository (repo): a folder tracked by Git

Working directory: your checked-out files

Staging area (index): files marked for the next commit

Commit: snapshot of staged changes

Branch: movable pointer to a commit

HEAD: currently checked-out commit or branch

Remote: reference to another repository (e.g. origin)

Upstream: remote branch you pull from/push to

Git Commands

git status

git add

git commit

git push

Examples

Full, human-readable status

git status

Compact view with branch info

git status -sb

Show even ignored files

git status --ignored

Check only a subdirectory

git status src/components

Stage a single file

git add README.md

Stage all changes in the repo (including file deletions)

git add --all

Stage only modifications & deletions (skip untracked files)

git add -u

Interactively select hunks to stage in myfile.js

git add -p myfile.js

Preview staged changes without affecting the index

git add --dry-run .

Simple commit with message

git commit -m "Add user authentication flow"

Stage all tracked changes and commit

git commit -am "Refactor API endpoints"

Correct last commit (e.g., amend message or include new staged changes)

git commit --amend -m "Improve API error handling"

Commit with explicit author and custom date

**git commit -m "Backdate release tag" **

**--author="Jane Doe <jane@example.com>" **

--date="2025-06-20 10:00:00"

Push current branch to its upstream

git push

Push 'feature-x' to 'origin' and set upstream

git push -u origin feature-x

Force-push to remote (use sparingly)

git push --force origin main

Push all tags to origin

git push origin --tags

Preview what will be pushed without making changes

git push --dry-run