



Basic Commands

Configuration

<code>git config --global user.name "Your Name"</code>	Sets the name you want attached to your commit transactions.
<code>git config --global user.email "your_email@example.com"</code>	Sets the email you want attached to your commit transactions.
<code>git config --list</code>	Lists all Git configuration settings.
<code>git config --global core.editor "code --wait"</code>	Sets VS Code as the default editor for Git (replace <code>code</code> with your preferred editor's command-line tool).

Starting a Repository

<code>git init</code>	Initializes a new Git repository in the current directory.
<code>git clone <repository_url></code>	Clones an existing Git repository from a remote URL.

Basic Workflow

<code>git status</code>	Shows the status of the working directory and staging area.
<code>git add <file></code>	Adds a file to the staging area.
<code>git commit -m "Commit message"</code>	Commits the staged changes with a descriptive message.
<code>git log</code>	Shows the commit history of the repository.

Branching and Merging

Branch Management

<code>git branch</code>	Lists all local branches in the repository.
<code>git branch <branch_name></code>	Creates a new branch with the specified name.
<code>git checkout <branch_name></code>	Switches to the specified branch.
<code>git branch -d <branch_name></code>	Deletes the specified branch (if it has been merged).
<code>git branch -D <branch_name></code>	Forces deletion of the specified branch, even if it hasn't been merged.

Merging Branches

<code>git merge <branch_name></code>	Merges the specified branch into the current branch.
<code>git mergetool</code>	Opens the configured merge tool to resolve merge conflicts.
<code>git commit</code>	After resolving conflicts, commit the merged changes.

Rebasing

<code>git rebase <branch_name></code>	Rebases the current branch onto the specified branch.
<code>git rebase -continue</code>	Continues the rebasing process after resolving conflicts.
<code>git rebase -abort</code>	Aborts the rebasing process and returns to the original branch state.

Remote Repositories

Connecting to Remotes

<code>git remote add <name> <url></code>	Adds a remote repository with the specified name and URL.
<code>git remote -v</code>	Lists all configured remote repositories with their URLs.
<code>git remote remove <name></code>	Removes the specified remote repository.

Pushing and Pulling

<code>git push <remote> <branch></code>	Pushes the local branch to the specified remote repository.
<code>git pull <remote> <branch></code>	Pulls changes from the specified remote branch and merges them into the current branch.
<code>git fetch <remote></code>	Downloads objects and refs from another repository.

Tracking Branches

<code>git push -u <remote> <branch></code>	Sets up a tracking connection between the local branch and the remote branch. Use <code>--all</code> to push all branches.
<code>git branch --set-upstream-to= <remote>/<branch></code>	Manually sets the upstream branch for a local branch.

Undoing Changes

Modifying Commits

<code>git commit --amend -m "New commit message"</code>	Amends the last commit with new staged changes and/or a new commit message.
<code>git reset HEAD~1</code>	Unstages the last commit, keeping the changes in the working directory.

Reverting Changes

<code>git checkout - <file></code>	Discards changes to a file in the working directory, reverting it to the last committed version.
<code>git revert <commit></code>	Creates a new commit that undoes the changes made in the specified commit.

Resetting

<code>git</code>	Resets the staging area and working
<code>reset --</code>	directory to the state of the specified
<code>soft</code>	commit, but leaves the changes in the
<code><commit></code>	working directory.
<code>git</code>	Resets the staging area to the state of the
<code>reset --</code>	specified commit, but leaves the changes in
<code>mixed</code>	the working directory (default behavior).
<code><commit></code>	
<code>git</code>	Resets the staging area and working
<code>reset --</code>	directory to the state of the specified
<code>hard</code>	commit, discarding all changes.
<code><commit></code>	