



Basic Git Commands

Configuration

<code>git config --global user.name "Your Name"</code>	Sets your name for commit messages.
<code>git config --global user.email "email@example.com"</code>	Sets your email address for commit messages.
<code>git config --global core.editor "code --wait"</code>	Sets VS Code as the default editor for commit messages (replace <code>code --wait</code> with your editor command).
<code>git config --list</code>	Lists all git configuration settings.

Starting a Repository

<code>git init</code>	Initializes a new Git repository in the current directory.
<code>git clone <repository_url></code>	Clones a 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 add .</code>	Adds all changes in the current directory to the staging area.
<code>git commit -m "Commit message"</code>	Commits the staged changes with a descriptive message.

Branching and Merging

Branch Management

<code>git branch</code>	Lists all local branches. The current branch is marked with an asterisk (*).
<code>git branch <branch_name></code>	Creates a new branch.
<code>git checkout <branch_name></code>	Switches to the specified branch.
<code>git checkout -b <branch_name></code>	Creates and switches to a new branch.
<code>git branch -d <branch_name></code>	Deletes a branch (only if it has been merged).
<code>git branch -D <branch_name></code>	Forces deletion of a 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 a merge tool to resolve conflicts.
<code>git log --graph --oneline --decorate</code>	Visualize the branch and merge history.

Rebasing

<code>git rebase <branch_name></code>	Rebases the current branch onto the specified branch. Rewrites commit history.
<code>git rebase --abort</code>	Abort the rebase process.
<code>git rebase --continue</code>	Continue the rebase process after resolving conflicts.

Remote Repositories

Connecting to Remotes

<code>git remote add origin <repository_url></code>	Adds a remote repository named 'origin'.
<code>git remote -v</code>	Lists all remote connections.
<code>git remote show origin</code>	Shows information about the remote 'origin'.

Pushing and Pulling

<code>git push origin <branch_name></code>	Pushes the specified branch to the remote 'origin'.
<code>git push -u origin <branch_name></code>	Sets up tracking information for the branch so that <code>git pull</code> and <code>git push</code> can be used without specifying the remote and branch.
<code>git pull origin <branch_name></code>	Pulls changes from the remote 'origin' to the specified branch.
<code>git fetch</code>	Fetches all changes from the remote but does not merge them.

GitHub Specifics

<code>git request-pull <start> <url> <end></code>	Generates a request pull message.
<code>gh repo clone <repo></code>	Clone a repository. Requires GitHub CLI (<code>gh</code>).

Undoing Changes

Undoing Local Changes

<code>git checkout -- <file></code>	Discards changes in the working directory for a specific file.
<code>git restore <file></code>	Alternative to <code>git checkout -- <file></code> for discarding local changes (more modern).
<code>git reset HEAD <file></code>	Removes a file from the staging area.
<code>git clean -n</code>	Dry run to see which files would be removed by <code>git clean</code> .
<code>git clean -f</code>	Removes untracked files from the working directory.

Undoing Commits

<code>git commit --amend -m "New commit message"</code>	Amends the last commit with new changes or message.
<code>git reset --soft HEAD-1</code>	Resets the last commit, keeping the changes in the staging area.
<code>git reset --mixed HEAD-1</code>	Resets the last commit, keeping the changes in the working directory.
<code>git reset --hard HEAD-1</code>	Resets the last commit and discards the changes. Use with CAUTION.
<code>git revert <commit_hash></code>	Creates a new commit that reverts the changes from the specified commit. Safe way to undo changes in shared repositories.