Git Cheatsheet

A concise reference for Git commands and concepts, ideal for quick lookups and reminders.

Basic Commands

Configuration

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

Starting a Repository

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

Basic Workflow

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

Branching and Merging

Branch Management

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

Merging Branches

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

Rebasing

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

Remote Repositories

Connecting to Remotes

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

Pushing and Pulling

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

Tracking Branches

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

Undoing Changes

Modifying Commits

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

Reverting Changes

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

Resetting

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