

Git - General Commands Cheatsheet

Setup & Configuration

<code>git config --global user.name "Name"</code>	- Sets your commit author name globally
<code>git config --global user.email "email"</code>	- Sets your commit author email globally
<code>git config --list</code>	- Lists all Git configuration settings

Repository Basics

<code>git init</code>	- Initializes a new Git repository in current directory
<code>git clone <url></code>	- Clones a remote repository to local machine
<code>git clone <url> <directory></code>	- Clones repository into specified directory
<code>git clone --depth 1 <url></code>	- Shallow clone with only latest commit history
<code>git remote -v</code>	- Lists all configured remote repositories
<code>git remote add origin <url></code>	- Adds a new remote named 'origin'
<code>git remote set-url origin <url></code>	- Changes URL of existing remote

Daily Workflow

<code>git status</code>	- Shows working tree status and staged changes
<code>git add <file></code>	- Stages specific file for commit
<code>git add .</code>	- Stages all changes in current directory
<code>git commit -m "message"</code>	- Commits staged changes with message
<code>git commit -am "message"</code>	- Stages tracked files and commits in one step
<code>git commit --amend</code>	- Modifies the most recent commit
<code>git push</code>	- Pushes commits to default remote branch
<code>git push -u origin <branch></code>	- Pushes and sets upstream tracking branch
<code>git pull</code>	- Fetches and merges changes from remote
<code>git pull --rebase</code>	- Fetches and rebases local commits on top
<code>git fetch</code>	- Downloads objects and refs from remote

Branching & Merging

<code>git branch</code>	- Lists all local branches
<code>git branch -a</code>	- Lists all local and remote branches
<code>git branch <name></code>	- Creates a new branch
<code>git branch -d <name></code>	- Deletes a (merged) branch
<code>git branch -D <name></code>	- Force deletes a branch
<code>git checkout <branch></code>	- Switches to a specific branch
<code>git checkout -b <branch></code>	- Creates and switches to new branch
<code>git switch <branch></code>	- Switches to a specific branch (modern approach)
<code>git switch -c <branch></code>	- Creates and switches to new branch (modern approach)
<code>git merge</code>	- Merges specified branch into current
<code>git merge --no-ff <branch></code>	- Displays configuration hierarchy

Git - General Commands Cheatsheet - 2

Rebasing

<code>git rebase <branch></code>	- Rebases current branch onto specified branch
<code>git rebase --continue</code>	- Continues rebase after resolving conflicts
<code>git rebase --abort</code>	- Aborts rebase and returns to original state
<code>git rebase --onto <new> <old> <branch></code>	- Rebases branch onto new base

Stashing

<code>git stash</code>	- Stashes uncommitted changes
<code>git stash push -m "message"</code>	- Stashes with message
<code>git stash list</code>	- Lists all stashed changes
<code>git stash pop</code>	- Applies and removes most recent stash
<code>git stash clear</code>	- Removes all stash entries

Viewing History

<code>git log</code>	- Shows commit history
<code>git log --oneline</code>	- Shows condensed commit history
<code>git log --graph --oneline --all</code>	- Shows branch graph visualisation
<code>git log -p <file></code>	- Shows commit history with diffs for file
<code>git log --author="name"</code>	- Filters commits by author
<code>git diff</code>	- Shows unstaged changes
<code>git diff --staged</code>	- Shows staged changes
<code>git diff <branch1>..<branch2></code>	- Shows differences between branches
<code>git show <commit></code>	- Shows details of specific commit
<code>git blame <file></code>	- Shows who changed each line of a file

Undoing Changes

<code>git restore <file></code>	- Discards changes in working directory
<code>git restore --staged <file></code>	- Unstages file, keeps changes
<code>git reset HEAD~1</code>	- Undoes last commit, keeps changes
<code>git reset --soft HEAD~1</code>	- Undoes last commit, keeps changes staged
<code>git reset --hard HEAD~1</code>	- Undoes last commit, discards all changes
<code>git revert <commit></code>	- Creates new commit that undoes changes
<code>git reflog</code>	- Shows history of HEAD changes
<code>git checkout <commit> -- <file></code>	- Restores file from specific commit

Notes

Stashing: Git stashing is a way to temporarily save (shelve) your uncommitted changes so you can switch branches or work on something else, without committing.

Rebasing: Rebasing is an operation that moves or replays your commits onto a new base commit.