

## Git Cheat Sheet

### 1 Creating Snapshots

```
git init % Initialize a new git repository (usage: starting new project)
git add file.js % Stage a specific file (usage: preparing for commit)
git add . % Stage all files in current directory (usage: bulk staging)
git status % Display current staging status (usage: check staged files)
git commit -m "message" % Commit staged files with message (usage: saving changes)
git commit -am "message" % Stage and commit in one step (usage: quick commits)
git rm file.js % Remove file from working directory and staging (usage: file deletion)
git rm --cached file.js % Unstage file (usage: remove from staging)
git mv file1.js file2.js % Rename or move file (usage: file rename/move)
git diff % Show unstaged changes (usage: review edits)
git diff --staged % Show staged changes (usage: confirm staging)
```

### 2 Browsing History

```
git log % Display full commit history (usage: viewing history)
git log --oneline % Compact commit history (usage: quick overview)
git log -3 % Last 3 commits (usage: recent commits)
git show HEAD % Show latest commit details (usage: inspect last commit)
git blame file.js % Show line-by-line author (usage: identify code authorship)
git tag v1.0 % Tag latest commit (usage: mark version release)
```

### 3 Branching and Merging

```
git branch bugfix % Create branch (usage: start feature)
git checkout bugfix % Switch to branch (usage: change working branch)
git merge bugfix % Merge branch into current (usage: integrate feature)
git merge --abort % Abort ongoing merge (usage: cancel conflict merge)
git stash push -m "WIP" % Temporarily store changes (usage: pause work)
git stash pop % Apply stashed changes (usage: resume paused work)
git cherry-pick commitID % Apply specific commit (usage: selective integration)
```

### 4 Collaboration

```
git clone url % Clone remote repository (usage: initial setup)
git fetch origin % Fetch remote updates (usage: synchronize without merging)
git pull % Fetch and merge (usage: sync with remote)
git push % Push changes (usage: update remote)
git remote add upstream url % Add new remote repository (usage: manage forks)
```

### 5 Rewriting History

```
git reset --soft HEAD^ % Undo last commit, keep staged (usage: correct recent commit)
git reset --hard HEAD^ % Discard last commit and changes (usage: rollback)
git revert commitID % Revert a specific commit (usage: undo specific commit)
git reflog % Recover lost commits (usage: find lost history)
git commit --amend % Amend last commit (usage: fix commit message)
git rebase -i HEAD~3 % Interactive rebase last 3 commits (usage: reorder, squash commits)
```