

# Rajeev's Git Survival Kit

## Common Daily Git Commands

git status               # Check status of working directory  
git add <file>           # Stage a file  
git commit -m "message"   # Commit changes  
git log                 # View commit history  
git diff                # Show changes  
git stash                # Save work temporarily  
git stash pop            # Restore stashed work

## Branching & Merging

git branch              # List branches  
git switch <branch>      # Switch to a branch  
git checkout -b <branch>   # Create and switch to a branch  
git merge <branch>       # Merge another branch into current  
git rebase <branch>      # Rebase current branch onto another

## Undo & Reset

git reset --soft HEAD~1   # Undo commit, keep changes staged  
git reset --mixed HEAD~1   # Undo commit, keep changes unstaged  
git reset --hard HEAD~1    # Undo commit and discard changes  
git restore <file>        # Discard local changes to a file

## Remote Collaboration

git clone <url>           # Clone a repository  
git fetch                # Fetch latest changes

# Rajeev's Git Survival Kit

git pull                   # Fetch and merge remote branch  
git pull --rebase         # Rebase your local changes on top  
git push                  # Push local commits to remote  
git push origin <src>:<dest> # Push to different remote branch

## Tags & Releases

git tag                   # List tags  
git tag -a v1.0 -m "msg"   # Create annotated tag  
git push origin v1.0       # Push a specific tag  
git describe              # Show current commit relative to tag

## Inspection & Debugging

git blame <file>         # Who changed each line  
git log -p <file>        # History of a file with diffs  
git show <commit>        # Show details of a commit  
git stash show -p         # See stashed changes

## Pro Tips

Use interactive rebase: git rebase -i HEAD~n  
Use .gitignore to clean up workspace  
Use git clean -fxn to dry run file cleanup  
Use aliases for repetitive tasks  
Use reflog to recover from lost HEADs: git reflog