

Repo setup	
git init	Create a new repo
git remote add <remote> <URL>	Link local to remote repo
git remote set-url <remote> <URL>	Change link to remote
git clone <URL> [<directory>]	Clone remote (to directory)

Info	
git status	View changed files
git log	View commit history
git show <SHA>	View commit info
git branch [-r] [-a]	View branches (remote / all)

Fixing mistakes	
git reset HEAD --<file>	Remove file from staging
git checkout --<file>	Revert changes to uncommitted file
git reset <remote branch>	Reset commits on current branch to origin, but keep changes
git reset --hard <remote branch>	Reset current branch to origin, discarding changes

Branches	
git checkout <branch>	Switch branches
git checkout -b <branch>	Create and checkout new branch
git branch <branch> <remote branch>	Create new local branch to track remote branch
git push --set-upstream <remote> <branch>	Set current branch to track remote branch
git push <remote> <branch>	Create new branch on remote
git branch -d <branch>	Delete local branch
git push origin --delete <branch>	Delete branch on remote

Stash	
git stash	Stash changes
git stash apply	Apply topmost stashed changes
git stash list	List stashed changes
git stash show	Show topmost stashed changes

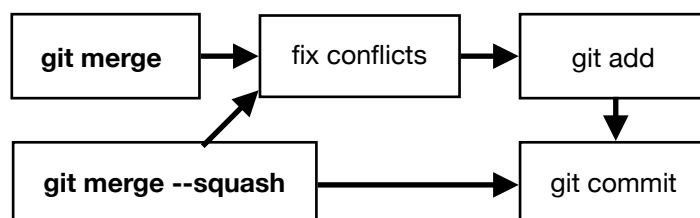
Commit	
git add <file>	Add file to staging
git add .	Add all changes to staging
git commit -m "<message>"	Commit staged changes
git commit	Commit without message (e.g. after during merge)

Communicate with remote	
git pull	Pull changes from remote
git push	Push changes to remote
git fetch	Fetch changes from remote (but don't integrate)

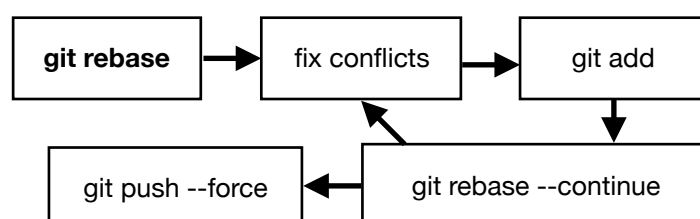
Merging	
git merge <branch>	Merge branch into current
git merge --squash <branch>	Merge and squash branch into current
git merge --abort	Abort uncommitted merge

Rebasing	Use with caution!
git rebase <new base>	Rebase current branch onto new base
git rebase <new base> <branch>	Rebase branch onto new base
git rebase --continue	Continue rebase after resolving conflict
git rebase --abort	Abort rebase
git push --force	Force push (e.g. after rebase)

Merge with conflicts and/or squash workflow



Rebase workflow



.gitignore must be created and edited manually