

Git

Basic usage

Initialise directory for git
Get git repository [to directory]

Stage file
Stage all files (recursive)
Stage all files (non-recursive)
Interactive staging

Commit staged [changed] files [with message]
Replace last commit with staged
Change multiple (n) commits

Unstage file
Undo changes to file
Undo all changes since last commit [commit]
Undo last <n> commits
Undo a pushed merge

Move file
Remove file
Stop tracking of file
Remove non-tracked files [dry run] [path]
Set files to be ignored

View status
View changes not staged
View staged changes
View commit log
View last 2 changes
View commit stats
View last commit
View commit tree
View gitk program

Remotes

Add remote
Show remote details
List remote shortnames
List remote urls
Rename remote
Remove remote

Update tracking branch [from remote]
Fetch [from remote] and merge [with branch]
Push [to remote] [localbranch][remotebranch]

Tags

List tags [by pattern]
Create tag [with message] [of commit]
Show tag details
Show last tag and commits to current
Push tag
Push all tags

Misc

Global changes
Debugging
Bisecting

```
git init
git clone <path/url> [local directory]
```

```
git add <file>
git add .
git add *
git add -i
```

```
git commit [-a] [-m "commit message"]
git commit --amend
git rebase -i HEAD~<n>
```

```
git unstage (with config option)
git checkout -- <file>
git reset --hard [commit]
git reset --hard HEAD~<n>
git revert -m 1 <commit>
```

```
git mv <file_from> <file_to>
git rm <file>
git rm --cached <file>
git clean -d [-n] [<path>]
Add to .gitignore
```

```
git status
git diff
git diff --staged
git log
git log -p -2
git log --stat
git last (with config option)
git tree (with config option)
gitk
```

```
git remote add <remotename> <url>
git remote show <remotename>
git remote
git remote -v
git remote rename <old name> <new name>
git remote rm <remote>
```

```
git fetch [remotename]
git pull [remotename] [localbranch]
git push [remotename] [localbranch:][branchname]
```

```
git tag [-l pattern] [commit]
git tag -a <name> [-m "message"] [commit]
git show <tag>
git describe --tags
git push <tag>
git push --tags
```

```
git filter-branch <filtertype> <filter options>
git blame <options> <file>
git bisect <command>
```

Branches

| | |
|--|--|
| Create new branch | <code>git branch <branchname></code> |
| Switch to [new] branch [from remote] | <code>git checkout [-b] <branchname> [remote/remotename]</code> <code>git checkout --track <remote/remotename></code> |
| View all branches [with last commits] | <code>git branch [-v]</code> |
| View [non-]merged branches | <code>git branch [--no]-merged</code> |
| View non-merged commits [base] [branch] | <code>git cherry -v [base] [branch]</code> |
| View changes to remote compared to local | <code>git log <remotebranch> ^<localbranch></code> |
| Select <commit> to add to branch | <code>git cherry-pick <commit></code> |
| Merge <branchname> into current branch | <code>git merge <branchname></code> |
| Visual merge tool | <code>git mergetool</code> |
| Rebase [topicbranch] on basebranch | <code>git rebase <basebranchname> [topicbranchname]</code> |
| Do not rebase commits that you have pushed to a public repository | |
| Force push (eg after rebasing) | <code>git push -f <remotename> <branchname></code> |
| Delete branch | <code>git branch -d <branchname></code> |
| Delete non-merged branch | <code>git branch -D <branchname></code> |
| Delete remote branch | <code>git push <remote> --delete <remotebranch></code> |

Stashing

| | |
|---|---|
| Stash changes for later use [description] | <code>git stash [save description]</code> |
| View stashes | <code>git stash list</code> |
| Open stash [stashname] | <code>git stash apply [stashname]</code> |
| Remove [stashname] from stash list | <code>git stash drop [stashname]</code> |
| Unapply stash | <code>git stash show -p <stashname> git apply -R</code> |
| Create branch from stash | <code>git stash branch <branchname></code> |

Example workflow

| | |
|--|---|
| <code>git clone <url></code> | Get a copy of the code |
| <code>git checkout master</code> | Make sure you're on master branch |
| <code>git pull --rebase</code> | [Optional] Update master if it has been some time since clone/last update |
| <code>git checkout -b topic</code> | Create topic branch and switch to it |
| <code>git commit -m "<message>"</code> | Make some changes and commits |
| <code>git rebase -i Head-<n></code> | [Optional] Change any commits if necessary |
| <code>git push origin topic:topic</code> | [Optional] Push topic to a new branch on origin |
| <code>git checkout master</code> | Switch to master branch |
| <code>git fetch origin</code> | Check for any updates to the remote repository |
| <code>git pull</code> | Update master to the latest version from the remote repository |
| <code>git checkout topic</code> | [Optional] If you want to rebase topic before merging, switch to it |
| <code>git rebase master</code> | [Optional with above] Rebase topic on master |
| <code>git checkout master</code> | [Optional with above] Switch back to master |
| <code>git merge topic</code> | Merge topic into master |
| <code>git push origin</code> | Upload new commits (from topic) to the remote repository |

Useful configuration options (place in ~/.gitconfig)

```
[user]
  name = Richard George
  email = rdg@roe.ac.uk
[alias]
  tree = log --graph --all --decorate --pretty=oneline --abbrev-commit
  co = checkout
  br = branch
  ci = commit
  st = status
  unstage = reset HEAD --
  last = log -1 HEAD
[color]
  ui = auto
[core]
  editor = vim
[merge]
  tool = vimdiff
```