Git

Basic usage

Initialise directory for git	`git init`
Get git repository [to directory]	`git clone <path/url> [local directory]`
Stage file	`git add <file>`
Stage all files (recursive)	`git add .`
Stage all files (non-recursive)	`git add *`
Interactive staging	`git add -i`
Commit staged [changed] files [with message]	`git commit [-a] [-m “commit message”]`
Replace last commit with staged	`git commit --amend`
Change multiple (n) commits	`git rebase -i HEAD~<n>`
Unstage file	`git unstage (with config option)`
Undo changes to file	`git checkout -- <file>`
Undo all changes since last commit [commit]	`git reset --hard [commit]`
Undo last <n> commits	`git reset --hard HEAD~<n>`
Undo a pushed merge	`git revert -m 1 <commit>`
Move file	`git mv <file_from> <file_to>`
Remove file	`git rm <file>`
Stop tracking of file	`git rm --cached <file>`
Remove non-tracked files [dry run] [path]	`git clean -d [-n] [<path>]`
Set files to be ignored	*Add to .gitignore*
View status	`git status`
View changes not staged	`git diff`
Vies staged changes	`git diff --staged`
View commit log	`git log`
View last 2 changes	`git log -p -2`
View commit stats	`git log --stat`
View last commit	`git last (with config option)`
View commit tree	`git tree (with config option)`
View gitk program	`gitk`

Remotes

Add remote	`git remote add <remotename> <url>`
Show remote details	`git remote show <remotename>`
List remote shortnames	`git remote`
List remote urls	`git remote -v`
Rename remote	`git remote rename <old name> <new name>`
Remove remote	`git remote rm <remote>`
Update remote branch [from remote]	`git fetch [remotename]`
Update remote branch [from remote] and merge	`git pull [remotename]`
Push [to remote] [localbranch][remotebranch]	`git push [remotename] [localbranch:][branchname]`

Tags

List tags [by pattern]	`git tag [-l pattern] [commit]`
Create tag [with message] [of commit]	`git tag -a <name> [-m “message”] [commit]`
Show tag details	`git show <tag>`
Show last tag and commits to current	`git describe --tags`
Push tag	`git push <tag>`
Push all tags	`git push --tags`

Misc

Global changes	`git filter-branch <filtertype> <filter options>`
Debugging	`git blame <options> <file>`
Bisecting	`git bisect <command>`

Branches

Create new branch	`git branch <branchname>`
Switch to [new] branch [from remote]	`git checkout [-b] <branchname> [remote/remotebranch]` `git checkout --track <remote/remotebranch>`
View all branches [with last commits]	`git branch [-v]`
View [non-]merged branches	`git branch [--no]-merged`
View non-merged commits [base] [branch]	`git cherry -v [base] [branch]`
View changes to remote compared to local	`git log <remotebranch> ^<localbranch>`
Merge <branchname> into current branch	`git merge <branchname>`
Visual merge tool	`git mergetool`
Rebase [topicbranch] on basebranch	`git rebase <basebranchname> [topicbranchname]`

Do not rebase commits that you have pushed to a public repository

Force push (eg after rebasing)	`git push -f <remotename> <branchname>`
Delete branch	`git branch -d <branchname>`
Delete non-merged branch	`git branch -D <branchname>`
Delete remote branch	`git push <remote> --delete <remotebranch>`

Stashing

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

Example workflow

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