

# Git Quick Reference

## Getting help:

**git help** *command*

or

**git** *command* --help

Show help for a command

## Repository creation:

**git init**

Create a repository in the current directory

**git clone** *url*

Clone a remote repository into a subdirectory

## File operations:

**git add** *path*

Add file or files in directory recursively

**git rm** *path*

Remove file or directory from the working tree

-f

Force deletion of file(s) from disk

**git mv** *path destination*

Move file or directory to new location

-f

Overwrite existing destination files

**git checkout** [*rev*] *file*

Restore file from current branch or revision

-f

Overwrite uncommitted local changes

## Working tree:

**git status**

Show status of the working tree

**git diff** [*path*]

Show diff of changes in the working tree

**git diff** HEAD *path*

Show diff of stages and unstaged changes

**git add** *path*

Stage file for commit

**git reset** HEAD *path*

Unstage file for commit

**git commit**

Commit files that has been staged (with git-add)

-a

Automatically stage all modified files

**git reset** --soft HEAD^

Undo commit & keep changes in the working tree

**git reset** --hard HEAD^

Reset the working tree to the last commit

**git clean**

Clean unknown files from the working tree

## Examining History:

**git log** [*path*]

View commit log, optionally for specific path

**git log** [*from*..*to*]

View commit log for a given revision range

--stat

List diffstat for each revision

-S'pattern'

Search history for changes matching pattern

**git blame** [*file*]

Show file annotated with line modifications

### Remote repositories - *remotes*:

<b>git fetch</b> <i>[remote]</i>	Fetch changes from a remote repository
<b>git pull</b> <i>[remote]</i>	Fetch and merge changes from a remote repository
<b>git push</b> <i>[remote]</i>	Push changes to a remote repository
<b>git remote</b>	List remote repositories
<b>git remote add</b> <i>remote url</i>	Add remote to list of tracked repositories

### Branches:

<b>git checkout</b> <i>branch</i>	Switch working tree to branch
<b>-b</b> <i>branch</i>	Create branch before switching to it
<b>git branch</b>	List local branches
<b>git branch -f</b> <i>branch rev</i>	Overwrite existing branch, start from revision
<b>git merge</b> <i>branch</i>	Merge changes from branch

### Exporting and importing:

<b>git apply</b> -< <i>file</i>	Apply patch from stdin
<b>git format-patch</b> <i>from[.to]</i>	Format a patch with log message and diffstat
<b>git archive</b> <i>rev</i> > <i>file</i>	Export snapshot of revision to file
<b>--prefix=</b> <i>dir/</i>	Nest all files in the snapshot in directory
<b>--format=</b> <i>[tar zip]</i>	Specify archive format to use: <i>tar</i> or <i>zip</i>

### Tags:

<b>git tag</b> <i>name [revision]</i>	Create tag for a given revision
<b>-s</b>	Sign tag with your private key using GPG
<b>-l</b> <i>[pattern]</i>	List tags, optionally matching pattern

### File status flags:

<b>M</b> modified	File has been modified
<b>C</b> copy-edit	File has been copied and modified
<b>R</b> rename-edit	File has been renamed and modified
<b>A</b> added	File has been added
<b>D</b> deleted	File has been deleted
<b>U</b> unmerged	File has conflicts after a merge