**Git Quick Reference**

| **Getting help:** |  |
| --- | --- |
| **git help** *command* | or |
| **git** *command* --help | Show help for a command |
| **Repository creation:** |  |
| **git init**  --bare | Create a repository in the current directory  Bare init |
| **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** –m *comment* | Commit files that has been staged (with git-add) |
| **git commit**  -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*:** |  |
| **git fetch** *[remote]* | Fetch changes from a remote repository |
| **git pull** *[remote]* | Fetch and merge changes from a remote repository |
| **git push** *[remote]*  **git push origin master** | Push changes to a remote repository  First push to bare repository |
| **git remote** | List remote repositories |
| **git remote add** *remote* *url* | Add remote to list of tracked repositories |

| **Branches:** |  |
| --- | --- |
| **git checkout** *branch* | Switch working tree to branch |
| -b *branch* | Create branch before switching to it |
| **git branch**  **git branch** **–r** | List local branches  List remote branches |
| **git branch** **-f** *branch* *rev*  **git branch** **–d** *branch* | Overwrite existing branch, start from revision  Remove the branch |
| **git merge** **[***to-branch***]** *from-branch*  **git rebase [***to-branch***]** *from-branch*  **git rebase –i**  **git rebase –continue**  **get cherry-pick** *commit1 commit2 …* | Merge changes from branch  Rebase changes from branch  Interactive rebase  Continue rebasing  Pick changes from commits (revisions) |
| **Exporting and importing:** |  |
| **git apply** - < *file* | Apply patch from stdin |
| **git format-patch** *from[..to]* | Format a patch with log message and diffstat |
| **git archive** *rev* > *file* | Export snapshot of revision to file |
| --prefix=*dir*/ | Nest all files in the snapshot in directory |
| --format=*[tar|zip]* | Specify archive format to use: *tar* or *zip* |

| **Tags:** |  |
| --- | --- |
| **git tag** *name* *[revision]* | Create tag for a given revision |
| -s | Sign tag with your private key using GPG |
| -l *[pattern]* | List tags, optionally matching pattern |
| **File status flags:** |  |
| M modified | File has been modified |
| C copy-edit | File has been copied and modified |
| R rename-edit | File has been renamed and modified |
| A added | File has been added |
| D deleted | File has been deleted |
| U unmerged | File has conflicts after a merge |

