

git-init - Create an empty Git repository or reinitialize an existing one

```
git init [-q | --quiet] [--bare] [--template=<template-directory>]
```

```
    [--separate-git-dir <git-dir>] [--object-format=<format>]
```

```
    [-b <branch-name> | --initial-branch=<branch-name>]
```

```
    [--shared[=<permissions>]] [<directory>]
```

<https://git-scm.com/docs/git-init>

git-status - Show the working tree status

```
git status [<options>...] [--] [<pathspec>...]
```

<https://git-scm.com/docs/git-status>

git-add - Add file contents to the index

```
git add [--verbose | -v] [--dry-run | -n] [--force | -f] [--interactive | -i] [--patch | -p]
```

```
    [--edit | -e] [--[no-]all | --[no-]ignore-removal | [--update | -u]] [--sparse]
```

```
    [--intent-to-add | -N] [--refresh] [--ignore-errors] [--ignore-missing] [--renormalize]
```

```
    [--chmod=(+|-)x] [--pathspec-from-file=<file> [--pathspec-file-nul]]
```

```
    [--] [<pathspec>...]
```

<https://git-scm.com/docs/git-add>

git-commit - Record changes to the repository

```
git commit [-a | --interactive | --patch] [-s] [-v] [-u<mode>] [--amend]
```

```
    [--dry-run] [(-c | -C | --squash) <commit> | --fixup [(amend|reword):]<commit>]
```

```
    [-F <file> | -m <msg>] [--reset-author] [--allow-empty]
```

```
    [--allow-empty-message] [--no-verify] [-e] [--author=<author>]
```

```
    [--date=<date>] [--cleanup=<mode>] [--[no-]status]
```

```
    [-i | -o] [--pathspec-from-file=<file> [--pathspec-file-nul]]
```

```
    [--trailer <token>[(=|:)<value>]]... [-S<keyid>]
```

```
    [--] [<pathspec>...]
```

<https://git-scm.com/docs/git-commit>

git-log - Show commit logs

```
git log [<options>] [<revision-range>] [--] [<path>...]
```

<https://git-scm.com/docs/git-log>

gitignore - Specifies intentionally untracked files to ignore

\$XDG_CONFIG_HOME/git/ignore, \$GIT_DIR/info/exclude, .gitignore

<https://git-scm.com/docs/gitignore>

git-branch - List, create, or delete branches

git branch [--color[=<when>] | --no-color] [--show-current]

[-v [--abbrev=<n> | --no-abbrev]]

[--column[=<options>] | --no-column] [--sort=<key>]

[--merged [<commit>]] [--no-merged [<commit>]]

[--contains [<commit>]] [--no-contains [<commit>]]

[--points-at <object>] [--format=<format>]

[(--r | --remotes) | (-a | --all)]

[--list] [<pattern>...]

git branch [--track[=(direct|inherit)] | --no-track] [-f]

[--recurse-submodules] <branchname> [<start-point>]

git branch (--set-upstream-to=<upstream> | -u <upstream>) [<branchname>]

git branch --unset-upstream [<branchname>]

git branch (-m | -M) [<oldbranch>] <newbranch>

git branch (-c | -C) [<oldbranch>] <newbranch>

git branch (-d | -D) [-r] <branchname>...

git branch --edit-description [<branchname>]

<https://git-scm.com/docs/git-branch>

git-switch - Switch branches

git switch [<options>] [--no-guess] <branch>

git switch [<options>] --detach [<start-point>]

git switch [<options>] (-c|-C) <new-branch> [<start-point>]

git switch [<options>] --orphan <new-branch>

<https://git-scm.com/docs/git-switch>

git-checkout - Switch branches or restore working tree files

```
git checkout [-q] [-f] [-m] [<branch>]
```

```
git checkout [-q] [-f] [-m] --detach [<branch>]
```

```
git checkout [-q] [-f] [-m] [--detach] <commit>
```

```
git checkout [-q] [-f] [-m] [[-b|-B|--orphan] <new-branch>] [<start-point>]
```

```
git checkout [-f|--ours|--theirs|-m|--conflict=<style>] [<tree-ish>] [--] <pathspec>...
```

```
git checkout [-f|--ours|--theirs|-m|--conflict=<style>] [<tree-ish>] --paths-from-file=<file> [--paths-from-file-nul]
```

```
git checkout (-p|--patch) [<tree-ish>] [--] [<pathspec>...]
```

<https://git-scm.com/docs/git-checkout>

git-merge - Join two or more development histories together

```
git merge [-n] [--stat] [--no-commit] [--squash] [--[no-]edit]
```

```
[-no-verify] [-s <strategy>] [-X <strategy-option>] [-S[<keyid>]]
```

```
 [--[no-]allow-unrelated-histories]
```

```
 [--[no-]rerere-autoupdate] [-m <msg>] [-F <file>]
```

```
 [--into-name <branch>] [<commit>...]
```

```
git merge (--continue | --abort | --quit)
```

<https://git-scm.com/docs/git-merge>

git-diff - Show changes between commits, commit and working tree, etc

```
git diff [<options>] [<commit>] [--] [<path>...]
```

```
git diff [<options>] --cached [--merge-base] [<commit>] [--] [<path>...]
```

```
git diff [<options>] [--merge-base] <commit> [<commit>...] <commit> [--] [<path>...]
```

```
git diff [<options>] <commit>...<commit> [--] [<path>...]
```

```
git diff [<options>] <blob> <blob>
```

```
git diff [<options>] --no-index [--] <path> <path>
```

<https://git-scm.com/docs/git-diff>

git-stash - Stash the changes in a dirty working directory away

git stash list [<log-options>]

git stash show [-u|--include-untracked|--only-untracked] [<diff-options>] [<stash>]

git stash drop [-q|--quiet] [<stash>]

git stash (pop | apply) [--index] [-q|--quiet] [<stash>]

git stash branch <branchname> [<stash>]

git stash [push [-p|--patch] [-S|--staged] [-k|--[no-]keep-index] [-q|--quiet]

[-u|--include-untracked] [-a|--all] [-m|--message <message>]

[--pathspec-from-file=<file> [--pathspec-file-nul]]

[--] [<pathspec>...]

git stash clear

git stash create [<message>]

git stash store [-m|--message <message>] [-q|--quiet] <commit>

<https://git-scm.com/docs/git-stash>

git-restore - Restore working tree files

git restore [<options>] [--source=<tree>] [--staged] [--worktree] [--] <pathspec>...

git restore [<options>] [--source=<tree>] [--staged] [--worktree] --pathspec-from-file=<file> [--pathspec-file-nul]

git restore (-p|--patch) [<options>] [--source=<tree>] [--staged] [--worktree] [--] [<pathspec>...]

<https://git-scm.com/docs/git-restore>

git-reset - Reset current HEAD to the specified state

git reset [-q] [<tree-ish>] [--] <pathspec>...

git reset [-q] [--pathspec-from-file=<file> [--pathspec-file-nul]] [<tree-ish>]

git reset (--patch | -p) [<tree-ish>] [--] [<pathspec>...]

git reset [--soft | --mixed [-N] | --hard | --merge | --keep] [-q] [<commit>]

<https://git-scm.com/docs/git-reset>

git-revert - Revert some existing commits

```
git revert [--no-edit] [-n] [-m parent-number] [-s] [-S[<keyid>]] <commit>...
```

```
git revert (--continue | --skip | --abort | --quit)
```

<https://git-scm.com/docs/git-revert>

git-push - Update remote refs along with associated objects

```
git push [--all | --mirror | --tags] [--follow-tags] [--atomic] [-n | --dry-run] [--receive-pack=<git-receive-pack>]
```

```
    [--repo=<repository>] [-f | --force] [-d | --delete] [--prune] [-v | --verbose]
```

```
    [-u | --set-upstream] [-o <string> | --push-option=<string>]
```

```
    [--no-signed|--signed=(true|false|if-asked)]
```

```
    [--force-with-lease[=<refname>[:<expect>]]] [--force-if-includes]]
```

```
    [--no-verify] [<repository> [<refspec>...]]
```

<https://git-scm.com/docs/git-push>

git-rebase - Reapply commits on top of another base tip

```
git rebase [-i | --interactive] [<options>] [--exec <cmd>]
```

```
    [--onto <newbase> | --keep-base] [<upstream> [<branch>]]
```

```
git rebase [-i | --interactive] [<options>] [--exec <cmd>] [--onto <newbase>]
```

```
    --root [<branch>]
```

```
git rebase (--continue | --skip | --abort | --quit | --edit-todo | --show-current-patch)
```

<https://git-scm.com/docs/git-rebase>

git-config - Get and set repository or global options

git config [<file-option>] [--type=<type>] [--fixed-value] [--show-origin] [--show-scope] [-z|--null] <name> [<value> [<value-pattern>]]

git config [<file-option>] [--type=<type>] --add <name> <value>

git config [<file-option>] [--type=<type>] [--fixed-value] --replace-all <name> <value> [<value-pattern>]

git config [<file-option>] [--type=<type>] [--show-origin] [--show-scope] [-z|--null] [--fixed-value] --get <name> [<value-pattern>]

git config [<file-option>] [--type=<type>] [--show-origin] [--show-scope] [-z|--null] [--fixed-value] --get-all <name> [<value-pattern>]

git config [<file-option>] [--type=<type>] [--show-origin] [--show-scope] [-z|--null] [--fixed-value] [--name-only] --get-regexp <name-regex> [<value-pattern>]

git config [<file-option>] [--type=<type>] [-z|--null] --get-urlmatch <name> <URL>

git config [<file-option>] [--fixed-value] --unset <name> [<value-pattern>]

git config [<file-option>] [--fixed-value] --unset-all <name> [<value-pattern>]

git config [<file-option>] --rename-section <old-name> <new-name>

git config [<file-option>] --remove-section <name>

git config [<file-option>] [--show-origin] [--show-scope] [-z|--null] [--name-only] -l | --list

git config [<file-option>] --get-color <name> [<default>]

git config [<file-option>] --get-colorbool <name> [<stdout-is-tty>]

git config [<file-option>] -e | --edit

<https://git-scm.com/docs/git-config>

git-hash-object - Compute object ID and optionally creates a blob from a file

git hash-object [-t <type>] [-w] [--path=<file>|--no-filters] [--stdin [--literally]] [--] <file>...

git hash-object [-t <type>] [-w] --stdin-paths [--no-filters]

<https://git-scm.com/docs/git-hash-object>

git-cat-file - Provide content or type and size information for repository objects

git cat-file <type> <object>

git cat-file (-e | -p) <object>

git cat-file (-t | -s) [--allow-unknown-type] <object>

git cat-file (--batch | --batch-check | --batch-command) [--batch-all-objects]

[--buffer] [--follow-symlinks] [--unordered]

[--textconv | --filters]

git cat-file (--textconv | --filters)

[<rev>:<path|tree-ish> | --path=<path|tree-ish> <rev>]

<https://git-scm.com/docs/git-cat-file>

git-reflog - Manage reflog information

git reflog <subcommand> <options>

<https://git-scm.com/docs/git-reflog>

