



HOME

HOME

HOSTING

HOSTING

TECHNOLOGY

TECHNOLOGY

ABOUT US

GIT Services[GIT Hosting](#)[GIT Tutorial](#)[Install GIT on Linux and MAC OS](#)[GIT Installation on Windows](#)[Start a GitHub Repository](#)[GIT Repository Management](#)[GIT Commands](#)[GIT Directory Structure](#)[SG-GIT Tutorial](#)[Create New Repository](#)[Clone GIT Repository Locally](#)[Clone GitHub Project](#)[Files Exclude List](#)[Destroy Repository](#)

GIT Commands Tutorial

Learn what are and how to use the most common GIT commands

This tutorial provides a list of the most common GIT commands, a short description for them and example usage. For a detailed description of all the GIT commands please check this [page](#).

git config

Sets configuration values for your user name, email, gpg key, preferred diff algorithm, file formats and more.

Examples:

```
git config --global user.name "My Name"
git config --global user.email "user@domain.com"
```

git init

Initializes a git repository – creates the initial `.git` directory in a new or in an existing project. Example:

```
git init
Initialized empty Git repository in /home/username/GIT/.git/
```

git clone

Creates a GIT repository copy from a remote source. Also adds the original location as a remote so you can fetch from it again and push to it if you have permissions. Example:

```
git clone git@github.com:user/test.git
```

git add

Adds files changes in your working directory to your index. Example:

```
git add .
```

git rm

Removes files from your index and your working directory so they will not be tracked. Example:

```
git rm filename
```

git commit

Takes all of the changes written in the index, creates a new commit object pointing to it and sets the branch to point to that new commit. Examples:

```
git commit -m 'committing added changes'
git commit -a -m 'committing all changes, equals to git add and git commit'
```

git status

Shows you the status of files in the index versus the working directory. It will list out files that are untracked (only in your working directory), modified (tracked but not yet updated in your index), and staged (added to your index and ready for committing). Example:

```
git status

# On branch master #
# Initial commit #
# Untracked files: #
# (use "git add <file>..." to include in what will be committed) #

README
```

git branch

Lists existing branches, including remote branches if '-a' is provided. Creates a new branch if a branch name is provided. Example:

```
git branch -a * master remotes/origin/master
```

git merge

Merges one or more branches into your current branch and automatically creates a new commit if there are no conflicts. Example:

```
git merge newbranchversion
```

git reset

Resets your index and working directory to the state of your last commit. Example:

```
git reset --hard HEAD
```

git tag

Tags a specific commit with a simple, human readable handle that never moves. Example:

```
git tag -a v1.0 -m 'this is version 1.0 tag'
```

git pull

Fetches the files from the remote repository and merges it with your local one. Example:

```
git pull origin
```

git push

Pushes all the modified local objects to the remote repository and advances its branches. Example:

```
git push origin master
```

git remote

Shows all the remote versions of your repository. Example:

```
git remote origin
```

git log

Shows a listing of commits on a branch including the corresponding details. Example:

```
git log commit
84f241e8a0d768fb37ff7ad40e294b61a99a0abe Author: User <user@domain.com> Date: Mon May 3 09:24:05 201
0 +0300 first commit
```

git diff

Generates patch files or statistics of differences between paths or files in your git repository, or your index or your working directory. Example:

```
git diff
```

git archive

Creates a tar or zip file including the contents of a single tree from your repository. Example:

```
git archive --format=zip master^ README >file.zip
```

git gc

Garbage collector for your repository. Optimizes your repository. Should be run occasionally. Example:

```
git gc
Counting objects: 7, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), done.
Total 7 (delta 1), reused 0 (delta 0)
```

git fsck

Does an integrity check of the Git file system, identifying corrupted objects. Example:

```
git fsck
```

git prune

Removes objects that are no longer pointed to by any object in any reachable branch. Example:

```
git prune
```

SERVICES

- Web Hosting
- Reseller Hosting
- Cloud Hosting
- Dedicated Servers
- Affiliate Program


WHY SITEGROUND

- Top Data Centers
- Outstanding Speed
- Amazing Uptime
- Best Support
- Client Reviews




ABOUT US

- Our People
- Careers
- Events
- Contact Us

RECENT FROM TWITTER

**SiteGround**4 days ago

We couldn't come to Minneapolis this year, but we've sent some cool stuff for #WCMSP attendees. Take a sticker for your laptop and learn from our brochure about all the great things we do for #WordPress with our #hosting pic.twitter.com/DvwUL3hs0t



SERVICE HOSTING

- Email Hosting
- FTP Hosting
- PHP Hosting
- MySQL Hosting
- cPanel Hosting

APPS HOSTING

- WordPress Hosting
- Joomla Hosting
- Magento Hosting
- Drupal Hosting
- PrestaShop Hosting

TEMPLATES

- Joomla Templates
- WordPress Themes
- Magento Themes

TUTORIALS

- Knowledge Base
- WordPress Tutorial
- Joomla Tutorial
- Magento Tutorial
- Drupal Tutorial

OTHERS

- WP eCommerce Hosting
- ImpressPages Hosting
- Student Hosting
- WordPress Speed Optimization

[Links](#) | [Legal](#) | [Privacy](#)

© 2004 - 2018 SiteGround.com. All rights reserved.

