



1.800.828.9231

Live Chat

Login

[Home](#)[Web Hosting](#)[WordPress Services](#)[Joomla Services](#)[Our Technology](#)[Our Support](#)[About Us](#)[Affiliates](#)

GIT Services

[General GIT Tutorial](#)

- ▶ [Install GIT on Linux and Mac](#)
- ▶ [GIT Installation on Windows](#)
- ▶ [Start a GitHub Repository](#)
- ▶ [GIT Repository Management](#)
- ▶ [Major GIT Commands](#)
- ▶ [GIT Directory Structure](#)

[SG-GIT Tutorial](#)

NO.1

WEB HOSTING

- **150+ Sitebuilding Tools**
- **Script installation**
- **Free Domain Name**
- **Unlimited email**
- **24/7 Support**
- **99.9% Uptime**
- **30 Days Money**

GIT Tutorial: Commands

Major GIT commands with examples

Here you will find a list with the major commands, their short descriptions and exemplary usage. For a detailed description of all the GIT commands please visit

<http://www.kernel.org/pub/software/scm/git/docs/>

Major GIT commands:

`git config`

Sets configuration values for your user name, email, gpg key, preferred diff algorithm, file formats and more. Example: `git config --global user.name "My Name" git config --global user.email "user@domain.com" cat ~/.gitconfig [user] name = My Name email = user@domain.com`

`git init`

Initializes a git repository – creates the initial '.git' directory in a new or in an existing project. Example: `cd /home/user /my_new_git_folder/ git init`

`git clone`

Makes 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`

[Back](#)[Learn More](#)**SIGN UP****\$3.95/month**

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 nothing added to commit but untracked files present (use "git add" to track)

`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 checkout`

Checks out a different branch – switches branches by updating the index, working tree, and HEAD to reflect the chosen branch. Example: `git checkout newbranch`

`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 stash`

Temporarily saves changes that you don't want to commit immediately. You can apply the changes later. Example: `git stash` Saved working directory and index state "WIP on master: 84f241e first commit" HEAD is now at 84f241e first commit (To restore them type "git stash apply")

`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 fetch`

Fetches all the objects from the remote repository that are not present in the local one. Example: `git fetch origin`

`git pull`

Fetches the files from the remote repository and merges it with your local one. This command is equal to the `git fetch` and the `git merge` sequence. 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 2010 +0300 first commit

`git show`

Shows information about a git object. Example: `git show commit 84f241e8a0d768fb37ff7ad40e294b61a99a0abe` Author: User <user@domain.com> Date: Mon May 3 09:24:05 2010 +0300 first commit diff --git a/README b/README new file mode 100644 index 0000000..e69de29

`git ls-tree`

Shows a tree object, including the mode and the name of each item and the SHA-1 value of the blob or the tree that it points to. Example: `git ls-tree master^{tree}` 100644 blob e69de29bb2d1d6434b8b29ae775ad8c2e48c5391 README

`git cat-file`

Used to view the type of an object through the SHA-1 value. Example: `git cat-file -t e69de29bb2d1d6434b8b29ae775ad8c2e48c5391` blob

`git grep`

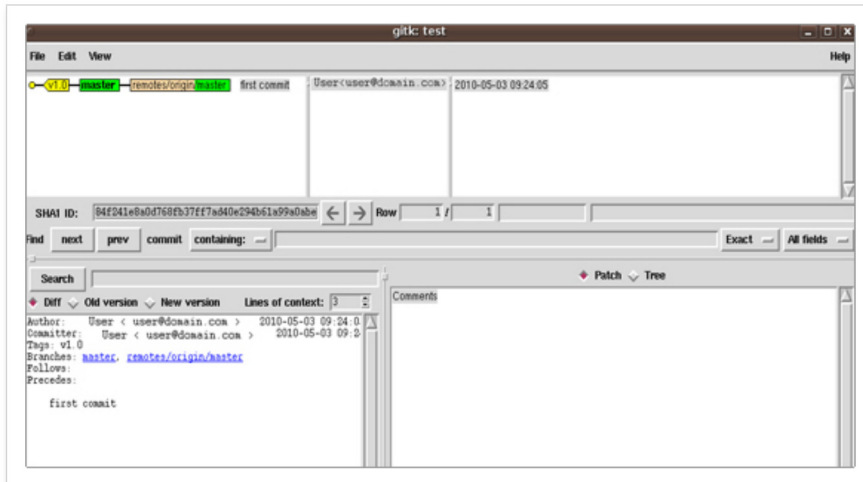
Lets you search through your trees of content for words and phrases. Example: `git grep "www.siteground.com" -- *.php`

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

gitk

Graphical Tcl/Tk based interface to a local Git repository. Example: gitk



git instaweb

Runs a web server with an interface into your local repository and automatically directs a web browser to it. Example: git instaweb --httpd=webrick git instaweb --stop



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`

SiteGround is not affiliated with or endorsed by the Git Project. Git is a member project of [Software Freedom Conservancy](#).

 Site Search

SEARCH

SERVICES

Web Hosting
Reseller Hosting
Cloud Hosting
Dedicated Servers
Affiliate Program



WHY SITEGROUND ABOUT US

Top Data Centers
Outstanding Speed
Amazing Uptime
Best Support
Client Reviews



Our People
Careers
Events
Contact Us

RECENT FROM BLOG

Jetpack Critical Security Vulnerability

 27 May, 2016  Comments

Top Customer Care of The Month: May

 26 May, 2016  Comments

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

SkaDate Hosting
WP eCommerce
Hosting
Student Hosting
Hosting Web en
Español

REGION: World ▼

WE ACCEPT



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

© 2004 - 2016 SiteGround. All rights reserved.

