DZone. (/)

👤 (/users/login.html)

🔍 (/search)

REFCARDZ (/refcardz)   **TREND REPORTS** (/trendreports)   **WEBINARS** (/webinars)   **ZONES** ⌄

DZone (/) > DevOps Zone (/devops-tutorials-tools-news) > Top 20 Git Commands With Examples

Top 20 Git Commands With Examples

(/users/3288749/sahiti-k.html) by Sahiti Kappagantula (/users/3288749/sahiti-k.html)

🎖 MVB ·

Jan. 22, 20 · DevOps Zone (/devops-tutorials-tools-news) · Tutorial

👍 Like (88)   💬 Comment (3)   ☆ Save   🐦 Tweet

In the previous blog, you got an understanding of what git is (https://www.edureka.co/blog/what-is-git/). In this blog, I will talk about the Top 20 Git Commands that you will be using frequently while you are working with Git (https://dzone.com/refcardz/getting-started-git).

Here are the Git commands which are being covered:

- **git config**

- **git init**

- **git clone**

- **git add**

- **git commit**

- **git diff**

- **git reset**

- **git status**

- **git rm**

- **git log**

- **git show**

DZone. (/)

(/users/login.html)   **Q** (/search)

REFCARDZ (/refcardz)   TREND REPORTS (/trendreports)   WEBINARS (/webinars)   ZONES ⌄

- **git checkout**

- **git merge**

- **git remote**

- **git push**

- **git pull**

- **git stash**

So, let's get started!

Git Commands

git config

Usage: `git config –global user.name` "[name]"

Usage: `git config –global user.email` "[email address]"

This command sets the author name and email address respectively to be used with your commits.

```
edureka@master:~$ git config --global user.name "sahitikappagantula"
edureka@master:~$ git config --global user.email "sahiti.kappagantula@edureka.co"
```

git init

Usage: `git init [repository name]`

This command is used to start a new repository.

```
edureka@master:~$ git init /home/edureka/Documents/DEMO
Initialized empty Git repository in /home/edureka/Documents/DEMO/.git/
```

git clone

Usage: `git clone [url]`

This command is used to obtain a repository from an existing URL.

git add

Usage: `git add [file]`

This command adds a file to the staging area.

Git Add Command - Git Commands - Edureka

Usage: `git add *`

This command adds one or more to the staging area.

Git Add Command - Git Commands - Edureka

git commit

Usage: `git commit —m "[ Type in the commit message]"`

This command records or snapshots the file permanently in the version history.

Git Commit Command - Git Commands - Edureka

Usage: `git commit —a`

This command commits any files you've added with the git add command and also commits any files you've changed since then.

Git Commit Command - Git Commands - Edureka

git diff

Usage: `git diff`

![DZone](/) (/)

This command shows the file differences which are not yet staged.

(/users/login.html)

Q (/search)

REFCARDZ (/refcardz)   **TREND REPORTS** (/trendreports)   **WEBINARS** (/webinars)   **ZONES** ˅

Git Diff Command - Git Commands - Edureka

Usage: `git diff —staged`

This command shows the differences between the files in the staging area and the latest version present.

Git Diff Command - Git Commands - Edureka

Usage: `git diff [first branch] [second branch]`

This command shows the differences between the two branches mentioned.

Git Diff Command - Git Commands - Edureka

git reset

Usage: `git reset [file]`

This command unstages the file, but it preserves the file contents.

This command unstages the file, but it preserves the file contents.

Git Reset Command - Git Commands - Edureka

Usage: `git reset [commit]`

This command undoes all the commits after the specified commit and preserves the changes locally.

Git Reset Command - Git Commands - Edureka

Usage: `git reset –hard [commit]` This command discards all history and goes back to the specified commit.

Git Reset Command - Git Commands - Edureka

git status

Usage: `git status`

This command lists all the files that have to be committed.

Git Status Command - Git Commands - Edureka

git rm

Usage: `git rm [file]`

This command deletes the file from your working directory and stages the deletion.

Git Rm Command - Git Commands - Edureka

git log

Usage: `git log`

DZone® (/)   🔍 (/search)

REFCARDZ (/refcardz)   **TREND REPORTS** (/trendreports)   **WEBINARS** (/webinars)   **ZONES** ⌄

Usage: `git log —follow[file]`

This command lists version history for a file, including the renaming of files also.

Git Log Command - Git Commands - Edureka

git show

Usage: `git show [commit]`

This command shows the metadata and content changes of the specified commit.

Git Show Command - Git Commands - Edureka

REFCARDZ (/refcardz)　　**TREND REPORTS** (/trendreports)　　**WEBINARS** (/webinars)　　**ZONES** ⌄

git tag

Usage: `git tag [commitID]`

This command is used to give tags to the specified commit.

Git Tag Command - Git Commands - Edureka

git branch

Usage: `git branch`

This command lists all the local branches in the current repository.

　Git Branch Command - Git Commands - Edureka

Usage: `git branch [branch name]`

This command creates a new branch.

　Git Branch Command - Git Commands - Edureka

Usage: `git branch -d [branch name]`

This command deletes the feature branch.

　Git Branch Command - Git Commands - Edureka

git checkout

Usage: `git checkout [branch name]`

This command is used to switch from one branch to another.

Git Checkout Command - Git Commands - Edureka

Usage: `git checkout –b [branch name]`

This command creates a new branch and also switches to it.

Git Checkout Command - Git Commands - Edureka

git merge

Usage: `git merge [branch name]`

This command merges the specified branch's history into the current branch.

Git Merge Command - Git Commands - Edureka

git remote

Usage: `git remote add [variable name] [Remote Server Link]`

This command is used to connect your local repository to the remote server.

Git Remote Command - Git Commands - Edureka

git push

Usage: `git push [variable name] master`

This command sends the committed changes of master branch to your remote repository.

Git Push Command - Git Commands - Edureka

Usage: `git push [variable name] [branch]`

This command sends the branch commits to your remote repository.

Git Push Command - Git Commands - Edureka

Usage: `git push —all [variable name]`

This command pushes all branches to your remote repository.

Git Push Command - Git Commands - Edureka

Usage: `git push [variable name] :[branch name]`

This command deletes a branch on your remote repository.

Git Push Command - Git Commands - Edureka

git pull

Usage: `git pull [Repository Link]`

This command fetches and merges changes on the remote server to your working directory.

Git Pull Command - Git Commands - Edureka

git stash

Usage: `git stash save`

This command temporarily stores all the modified tracked files.

Git Stash Command - Git Commands - Edureka

DZone. (/)

👤 (/users/login.html)   🔍 (/search)

Usage: `git stash pop`

REFCARDZ (/refcardz)   **TREND REPORTS** (/trendreports)   **WEBINARS** (/webinars)   **ZONES** ⌄

This command restores the most recently stashed files.

Git Stash Command - Git Commands - Edureka

Usage: `git stash list`

This command lists all stashed changesets.

Git Stash Command - Git Commands - Edureka

Usage: `git stash drop`

This command discards the most recently stashed changeset.

Git Stash Command - Git Commands - Edureka

Want to learn more about git commands? Here is a Git Tutorial (https://www.edureka.co/blog/git-tutorial/) to get you started. Alternatively, you can take a top-down approach and start with this DevOps Tutorial. (https://www.edureka.co/blog/devops-tutorial)

Topics: OPEN SOURCE, GIT, GIT COMMANDS, COMMAND EXAMPLES

Popular on DZone

- **7 Essential Books for Every Software Architect (/articles/essential-books-for-every-software-architect?fromrel=true)**

- **Service Locator Design Pattern in Java: An Introduction, Class/Sequence Diagram, and Implementation (/articles/service-locator-design-pattern-in-java-introductio?**

DZone. (/)

fromrel=true)

👤 (/users/login.html)   🔍 (/search)

- **Interview Tips for Agile Roles: Scrum Masters, Product Owners, and Agile Coaches**
REFCARDZ (/refcardz)   **TREND REPORTS** (/trendreports)   **WEBINARS** (/webinars)   **ZONES** ⌄
(/articles/interview-tips-for-scrum-masters-product-owners-ag?fromrel=true)

- **The five skills you must master to be outstanding as a software engineer (/articles/the-five-skills-you-must-master-to-be-outstanding?fromrel=true)**

ABOUT US

About DZone **(/pages/about)**

Send feedback **(mailto:support@dzone.com)**

Careers **(https://devada.com/careers/)**

Sitemap **(/sitemap)**

ADVERTISE

Advertise with DZone **(/pages/advertise)**

CONTRIBUTE ON DZONE

Article Submission Guidelines **(/articles/dzones-article-submission-guidelines)**

MVB Program **(/pages/mvb)**

Become a Contributor **(/pages/contribute)**

Visit the Writers' Zone **(/writers-zone)**

LEGAL

Terms of Service **(/pages/tos)**

Privacy Policy **(/pages/privacy)**

CONTACT US

600 Park Offices Drive

Suite 300

Durham, NC 27709

support@dzone.com **(mailto:support@dzone.com)**

+1 (919) 678-0300 **(tel:+19196780300)**

Let's be friends: 📡 🐦 𝘧 in

(/pages/feedburner/DZone)(https://twitter.com/DZoneInc)(https://www.facebook.com/DZone)(https://www.linkedin.com/company/dzone/)

DZone.com is powered by 🖼️AnswerHub logo **(https://devada.com/answerhub/)**