git - the simple guide

just a simple guide for getting started with git. no deep shit ;)

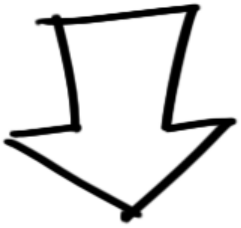
Tweet

by Roger Dudler

credits to @tfnico, @fhd and Namics

de in deutsch, español, français, indonesian, italiano, nederlands, polski, português, русский, မြန်မာ, 日本語, 中文, 한국어 Vietnamese

please report issues on github



setup

Download git for OSX

Download git for Windows

Download git for Linux

create a new repository

create a new directory, open it and perform a

```
git init
```

to create a new git repository.

checkout a repository

create a working copy of a local repository by running the command

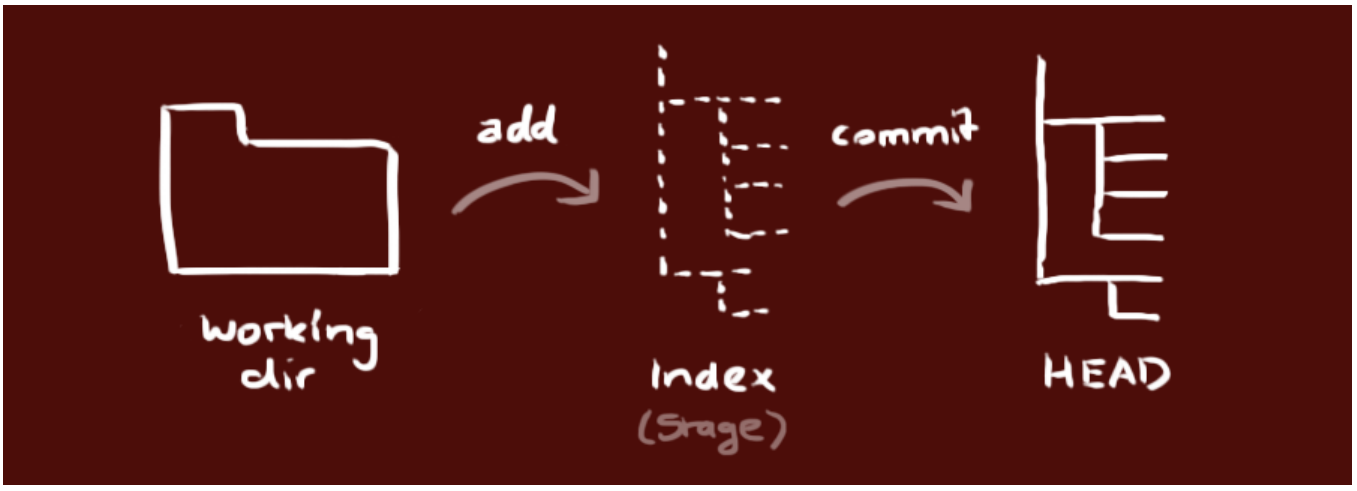
```
git clone /path/to/repository
```

when using a remote server, your command will be

```
git clone username@host:/path/to/repository
```

workflow

your local repository consists of three "trees" maintained by git. the first one is your `Working Directory` which holds the actual files. the second one is the `Index` which acts as a staging area and finally the `HEAD` which points to the last commit you've made.



add & commit

You can propose changes (add it to the **Index**) using

```
git add <filename>
git add *
```

This is the first step in the basic git workflow. To actually commit these

changes use

```
git commit -m "Commit message"
```

Now the file is committed to the **HEAD**, but not in your remote repository yet.

pushing changes

Your changes are now in the **HEAD** of your local working copy. To send those changes to your remote repository, execute

```
git push origin master
```

Change *master* to whatever branch you want to push your changes to.

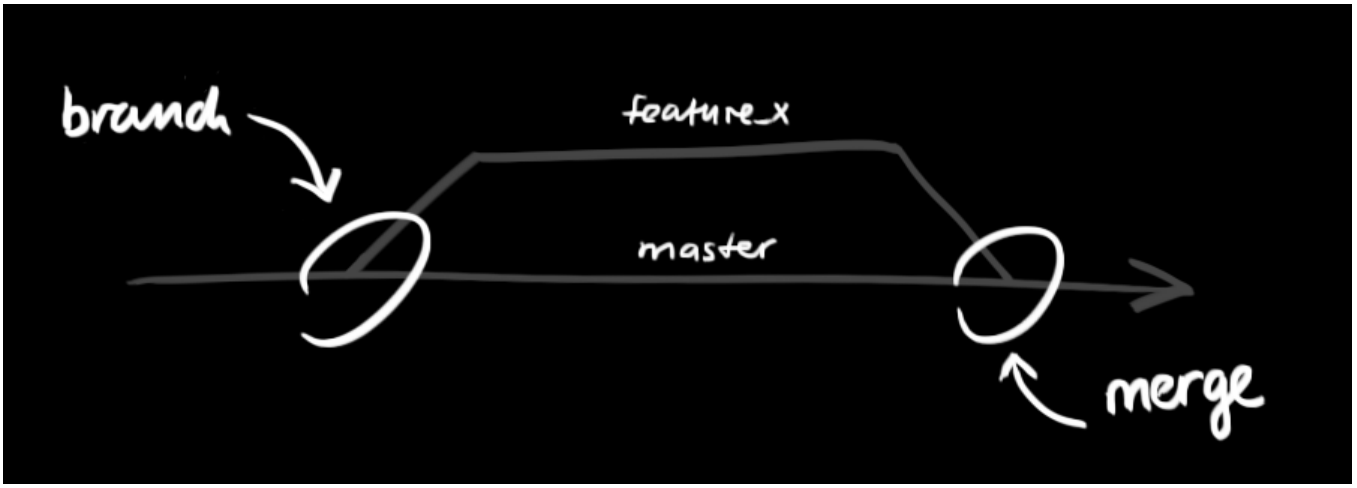
If you have not cloned an existing repository and want to connect your repository to a remote server, you need to add it with

```
git remote add origin <server>
```

Now you are able to push your changes to the selected remote server

branching

Branches are used to develop features isolated from each other. The *master* branch is the "default" branch when you create a repository. Use other branches for development and merge them back to the master branch upon completion.



create a new branch named "feature_x" and switch to it using

```
git checkout -b feature_x
```

switch back to master

```
git checkout master
```

and delete the branch again

```
git branch -d feature_x
```

a branch is *not available to others* unless you push the branch to your remote repository

```
git push origin <branch>
```

update & merge

to update your local repository to the newest commit, execute

```
git pull
```

in your working directory to *fetch* and *merge* remote changes.

to merge another branch into your active branch (e.g. master), use

```
git merge <branch>
```

in both cases git tries to auto-merge changes. Unfortunately, this is not always possible and results in *conflicts*. You are responsible to merge those *conflicts* manually by editing the files shown by git. After changing, you need to mark them as merged with

```
git add <filename>
```

before merging changes, you can also preview them by using

```
git diff <source_branch> <target_branch>
```

tagging

it's recommended to create tags for software releases. this is a known concept, which also exists in SVN. You can create a new tag named *1.0.0* by executing

```
git tag 1.0.0 1b2e1d63ff
```

the *1b2e1d63ff* stands for the first 10 characters of the commit id you want to reference with your tag. You can get the commit id by looking at the...

log

in its simplest form, you can study repository history using.. `git log`

You can add a lot of parameters to make the log look like what you want. To see only the commits of a certain author:

```
git log --author=bob
```

To see a very compressed log where each commit is one line:

```
git log --pretty=oneline
```

Or maybe you want to see an ASCII art tree of all the branches, decorated with the names of tags and branches:

```
git log --graph --oneline --decorate --all
```

See only which files have changed:

```
git log --name-status
```

These are just a few of the possible parameters you can use. For more, see

```
git log --help
```

replace local changes

In case you did something wrong, which for sure never happens ;), you can replace local changes using the command

```
git checkout -- <filename>
```

this replaces the changes in your working tree with the last content in HEAD. Changes already added to the index, as well as new files, will be kept.

If you instead want to drop all your local changes and commits, fetch the latest history from the server and point your local master branch at it like this

```
git fetch origin
git reset --hard origin/master
```

useful hints

built-in git GUI

```
gitk
```

use colorful git output

```
git config color.ui true
```

show log on just one line per commit

```
git config format.pretty oneline
```

use interactive adding

```
git add -i
```

links & resources

graphical clients

GitX (L) (OSX, open source)
Tower (OSX)
Source Tree (OSX & Windows, free)
GitHub for Mac (OSX, free)
GitBox (OSX, App Store)

guides

Git Community Book
Pro Git
Think like a git
GitHub Help
A Visual Git Guide

get help

Git User Mailing List
#git on irc.freenode.net

comments

769 Comments git - the simple guide 1 Login

Recommend 332 Share Sort by Newest

Join the discussion…

Roger Albino • 4 days ago
Muito obrigado. ;)
• Reply • Share ›

mills • 6 days ago
gitkrakenn found on http://www.becomeonewiththecod...
• Reply • Share ›

Buhraz Bolgo • 24 days ago
Any git clients that happen to run on Linux? maybe ... I know OSX and windows are the only two operating systems in the world, but just in case they are not, maybe a a couple of Linux gui clients, for KDE or GTK would have balanced the offering a little.
• Reply • Share ›

Igor Moura → Buhraz Bolgo • 24 days ago
I've used gitkraken and liked it so far.
1 • Reply • Share ›

Buhraz Bolgo → Igor Moura • 24 days ago
then i found GitEye from CollabNet.

Much lighter on cpu, and does seem to have all the functionality. Very professional looking and feeling plus you can click and open the files in your editor, something gitkraken was just not going to do. GitEye is

running now on my desktop, and i can still type normally, something i couldn't say with gitkraken running.

• Reply • Share ›

Igor Moura ➦ Buhraz Bolgo • 23 days ago

Interesting, I didn't have any problem CPU-related with GitKraken (although I have to admit that I use the terminal more far often than GitKraken itself).
I'm glad that you found a nice alternative though :)

• Reply • Share ›

Buhraz Bolgo ➦ Igor Moura • 23 days ago

hoping this will spur a debugging session, the cpu load was coming from X not GitKraken directly, something GitKraken was making X do caused a significant CPU load

• Reply • Share ›

Buhraz Bolgo ➦ Igor Moura • 24 days ago

Fount ot to be nice, liked the ability to do the push and pull the staging and nu-staging and the commit with comment .. nice graph. But i can't use it. It is simply way too heavy on the cpu, consumes 30% of a quad xeon, you have to be kidding me. it literally chewed up my system to the point where i had to wait for mouse clicks. gitkraken need more work for sure, so i went hunting...

• Reply • Share ›

Buhraz Bolgo ➦ Igor Moura • 24 days ago

Just installed it, seems to ask for an e-mail and an activation, looks good tho, will see how it works out. looks very heavy on the cpu

• Reply • Share ›

Jitendra Sonawane • 25 days ago

one of the best article I ever seen .super easy to learn Git

• Reply • Share ›

Evan Brucker • 25 days ago

Awesome! Super simple and useful!

• Reply • Share ›

Jeff HU • a month ago

Awesome one!

• Reply • Share ›

Vijil • a month ago

As this website provided with many useful links users can easily go through each topic. Also, anyone interested finds a tutor then visit the link https://preply.com/. It helps you find private tutors at very affordable prices.

1 • Reply • Share ›

NeuworkNetral • a month ago

Good job

• Reply • Share ›

electorypus • a month ago

Very good idea, nice website! I'll save this for later, quick referencing. Thanks!

• Reply • Share ›

alst • 2 months ago

Nice one!!

• Reply • Share ›

sdfsdf • 2 months ago

lol. pure shit. Doesn't work whatsoever. IT's not for beginners. It's nothing at all like TFS, where you can just right click and check in. This is 100X harder. So, you need an actual working example that makes it 100% clear

• Reply • Share ›

• Reply • Share ›

Terrance Tibbs ➦ sdfsdf • 2 months ago
I hear what you're saying.Try this:

go to youtube.com, and search for:

Github Tutorial For Beginners - Github Basics for Mac or Windows & Source Control Basics

This guy has two videos that covers everything you need to master Git/Github, and he uses an example discusses the general workflow he uses, day-to-day, when working in a team. I hope it helps a bit ;-)
1 • Reply • Share ›

ajhstn • 2 months ago
Can you dig deeper into the workflow section for me please?

Should the workflow be work, git add, wait for review, or git add & commit, wait for review, or git add & commit & push, wait for review?

If multiple editors are making small changes say 1 to css, 1 to html, should they have their own branch or not?
• Reply • Share ›

Arthur Accioly • 2 months ago
Awesome! Thanks!
• Reply • Share ›

mosby josh • 2 months ago
wella that was quick and awesome !, thanks
• Reply • Share ›

Ramesh syangtan • 2 months ago
Thank you very much !!!
• Reply • Share ›

Joseph Akwasi-Kumah Jnr • 2 months ago
thanks a million...
• Reply • Share ›

Sinh • 2 months ago
Thank you very much
• Reply • Share ›

vallabh govindrao potadar • 2 months ago
super and very helpful
• Reply • Share ›

Ryan Jentzsch • 2 months ago
Great guide. Thanks. A free graphical client for Linux is Gitkraken.
• Reply • Share ›

Mohamed Afzal Mulla • 2 months ago
Very nice thanks !
• Reply • Share ›

Webperts • 2 months ago
Excellent, it is very helpful for basic and starters.
• Reply • Share ›

rebotak • 3 months ago
this helps a lot, thank you!
• Reply • Share ›

Victor Chen • 3 months ago
Awesome, thanks!
• Reply • Share ›

Mahesh • 3 months ago
best for beginners
• Reply • Share ›

Allison Seamans • 3 months ago
thank you!
• Reply • Share ›

Dthi Demitrios • 3 months ago
INFINI-UPVOTE!
• Reply • Share ›

Ritesh Bhat • 3 months ago
excellent post for git beginners
• Reply • Share ›

ShloEmi • 3 months ago
Nice 1 - 10x!!
• Reply • Share ›

Csongor • 3 months ago
A very good cross-platform git client is GitKraken. Works on Windows, Linux, and Mac.
• Reply • Share ›

Roman Alexeev • 3 months ago
May I suggest a couple changes?
1) Instead of git add -i, you want git add -p 99% of the time
And git checkout -p is equally awesome

git add -p -- to add chunks to stash interactively
git checkout -p -- to discard unneeded edits interactively (goodbye, debug prints)

2) tig for graphical interface. Awesome and simple and works everywhere. tig blame is particularly useful.
• Reply • Share ›

mist42nz • 3 months ago
EXcellent thank you
• Reply • Share ›

Shamsul Arefin Sajib • 3 months ago
Superb, i am a newbie in git learning, found it awesome .
Thanks a lot author.
• Reply • Share ›

Avi • 3 months ago
I am new to git commands... this page made my life easy...good page... Bookmarkeed :)
..... I see git init command is missing.... add it if possible....
• Reply • Share ›

Avi → Avi • 3 months ago
Oops it is there.... !! ;)
• Reply • Share ›

Chris Teddi-b'poetic Vaughan • 3 months ago
OMG thank you!!!! just got a new job where they use git and I was so lost lol
• Reply • Share ›

Muhammad Irshad • 3 months ago
Nice guide, thanks for making life easy.
• Reply • Share ›

Dwi Ahmad • 4 months ago
i got a problem, how to update cloned git repository to ofline with same directory
• Reply • Share ›

Bookmarked • 4 months ago
Just missing rebase otherwise excellent :)
• Reply • Share ›

Meh • 4 months ago
While this guide is nice and stuff, I would say that `git pull` is usually a source of troubles. Why not `git fetch` and `git merge` or `git rebase`?
• Reply • Share ›

Sergiu • 4 months ago
OMG! i was searching this page whole my life!!
• Reply • Share ›

Vinny Alves • 4 months ago
Bookmarked :)
• Reply • Share ›

Mike S • 4 months ago
Thanks. Very useful.
• Reply • Share ›

Lissette Ganoza • 4 months ago
Excellent!! best guide!
• Reply • Share ›