

git “Cheat Sheet”

Nicholas P. Ross et al. (your name here....)

November 24, 2020

Abstract

This is a simple few notes that will hopefully form something like a “cheat sheet” for folks using git, and e.g. github. Slightly “meta”-ly, I’ve uploaded this file to github:

https://github.com/d80b2t/git_cheatsheet/.

1 Learn Git branching

¹This link::

<https://learngitbranching.js.org/>

seems to be the best thing I've seen so far.

Welcome to Learn Git Branching Interested in learning Git? Well you've come to the right place! "Learn Git Branching" is the most visual and interactive way to learn Git on the web; you'll be challenged with exciting levels, given step-by-step demonstrations of powerful features, and maybe even have a bit of fun along the way.

After this dialog you'll see the variety of levels we have to offer. If you're a beginner, just go ahead and start with the first. If you already know some Git basics, try some of our later more challenging levels.

You can see all the commands available with show commands at the terminal.

PS: Want to go straight to a sandbox next time? Try out this special link.

¹from Loki, thank you!! :-)

2 Some Youtube links

Git It? How to use Git and Github.

3 My github accounts

d80b2t

npr247

4 git Notes

4.1 To Check-out:

```
> git init
> git clone [url]
e.g.
> git clone https://github.com/crazygirl9991/SeniorResearch
```

4.2 Status check:

```
> git status
```

4.3 Update your local branch:

```
> git pull
```

4.4 To Add new files/material:

```
> git add
> git commit -m "[descriptive message]"
```

4.5 To then place on repository:

```
> git push
```

4.6 Version Control Basics (1)

```
> git log
```

will give the log of commits, where the beginning of the hash (which will be something a bit crazy like 80dd432df626ebf67d75adbd75bf0fad2a01f024E) is also the 'revision number', in this case:

Latest commit 80dd432 6 minutes ago

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

looks better ;-), or even:

```
> git log -pretty=format:"%h - %an, %ar : %s"
```

5 Git-Specific Commands

THESE NOTES ARE STRAIGHT FROM

“Understanding Github a-journey for beginners-part-1/ Since Git was designed with a big project like Linux in mind, there are a lot of Git commands. However, to use the basics of Git, you’ll only need to know a few terms. They all begin the same way, with the word “git”.

git init: Initializes a new Git repository. Until you run this command inside a repository or directory, it’s just a regular folder. Only after you input this does it accept further Git commands.

git config: Short for “configure” this is most useful when you’re setting up Git for the first time.

git status: Check the status of your repository. See which files are inside it, which changes still need to be committed, and which branch of the repository you’re currently working on.

git add: This does not add new files to your repository. Instead, it brings new files to Git’s attention. After you add files, they’re included in Git’s “snapshots” of the repository.

git commit: Git’s most important command. After you make any sort of change, you input this in order to take a “snapshot” of the repository. Usually it goes `git commit -m “Message here”`. The `-m` indicates that the following section of the command should be read as a message.

git branch: Working with multiple collaborators and want to make changes on your own? This command will let you build a new branch, or timeline of commits, of changes and file additions that are completely your own. Your title goes after the command. If you wanted a new branch called “cats”, you’d type `git branch cats`.

git checkout: Literally allows you to “check out” a repository that you are not currently inside. This is a navigational command that lets you move to the repository you want to check. You can use this command as `git checkout master` to look at the master branch, or `git checkout cats` to look at another branch.

git merge: When you’re done working on a branch, you can merge your changes back to the master branch, which is visible to all collaborators. `git merge cats` would take all the changes you made to the “cats” branch and add them to the master.

git push: If you’re working on your local computer, and want your commits to be visible online on GitHub as well, you “push” the changes up to GitHub with this command.

git pull: If you’re working on your local computer and want the most up-to-date version of your repository to work with, you “pull” the changes down from GitHub with this command.

`git help`: Forgot a command? Type this into the command line to bring up the 21 most common git commands. You can also be more specific and type “git help init” or another term to figure out how to use and configure a specific git command.

CONTINUE READING <http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/> !!!

6 Things to learn...

`git remote add <name> <url>`

<https://help.github.com/articles/removing-a-remote/>

<http://gitref.org/remotes/>

7 Useful URLs

<http://stackoverflow.com/questions/5617211/what-is-git-remote-add-and-git-push-origin-master>

<https://help.github.com/articles/git-cheatsheet/> for the e.g. [github-git-cheat-sheet.pdf](#)

<https://github.com/AlexZeitler/gitcheatsheet>

Resources

<http://readwrite.com/2013/09/30/understanding-github-a-journey-for-beginners-part-1/>

<http://readwrite.com/2013/10/02/github-for-beginners-part-2/>

YouTube

What is Git - A Quick Introduction to the Git Version Control System

Learn Git in 20 Minutes

Copying a GitHub Repository to Your Local Computer (Youtube)