`git` "Cheat Sheet"

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

August 30, 2016

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 My github accounts

d80b2t
npr247

1.1 notes

On Jul 23, 2014, at 7:37 PM, Nic Ross <npross@lbl.gov> wrote:

Gang,

In an effort to get our code repository going, here's my github details:
https://github.com/d80b2t
https://github.com/d80b2t/spies
https://github.com/d80b2t/group_code

So, after creating a repository, you can go to that repository, click settings, and click "collaborators" to add collaborators. I have done this for John and Rob for the https://github.com/d80b2t/group_code repository.

Thanks, Nic

2 git Notes

2.1 To Check-out:

```
> git init
> git clone [url]
```

e.g.
> `git clone https://github.com/crazygirl9991/SeniorResearch`

2.2 Status check:

> `git status`

2.3 Update your local branch:

> `git pull`

2.4 To Add new files/material:

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

2.5 To then place on repository:

> `git push`

2.6 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"`

3 Git-Specific Commands

THESE NOTES ARE STRAIGHT FROM http://readwrite.com/2013/09/30/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, youll 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, its 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 youre 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 youre currently working on.

`git add`: This does not add new files to your repository. Instead, it brings new files to Gits attention. After you add files, theyre included in Git's "snapshots" of the repository.

`git commit`: Gits 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 youre 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 youre 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 youre 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/ !!!

4 Things to learn...

git remote add <name> <url >

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

http://gitref.org/remotes/

5 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)