Git Tutorial and Reference Collection

Introduction

This page provides links to some useful tutorial and reference information about the git version control system. This overview takes the view that the best way to learn to use git effectively is to learn it as a data-structure and a set of algorithms to manipulate that data-structure which is consistent this nice blog article and this little git tutorial. This is important because the git command-line interface is widely considered to be overly complex and confusing or just plain cruddy. For example, the same git command like 'checkout' can do wildly different things depending on the other arguments that are passed into the command, or the state of the git repository. But git is still currently the dominant VC system.

Links to Git tutorial and reference material

For a full featured introduction and reference for git, a good source is the free

online book:

- Pro Git: 2nd Edition by Scott Chacon and Ben Straub

But *Pro Git* is a big book and that is a lot of information for most people to remember. And some people have a hard time finding the specific information they need in that book. Therefore, provided below are some other shorter more targeted references that some people might find very helpful to understand git and will want to consult in day-to-day usage of git.

Assuming you are a complete git beginner and can't wait to start contributing to a git repo, you would be wise to first take note of:

- Critical Beginner Git Usage Tips

After that, for a nice (relatively short) visual reference for common git operations, see:

- Visual Git Reference

A nice interactive visual tutorial (and simple git simulator) and reference for common git operations is:

- Visualizing Git Concepts with D3

Another similar site is:

- Learn Git Branching

(The "Lean Git Branching" site is more interactive with more guided lessons than the "Visualizing Git Concepts with D3" site. It also supports more git commands like "git cherry-pick" and it allows for larger git branching simulations. Try out the sandbox mode to do quick unguided git workflow simulations. However, this is still just a simulator for git so does not support all git commands and options.)

To really be able to use git well, you have to know a little bit about how it works and how it stores changes to a repository. To help understand this, a nice short

overview of the git object model is given in:

- The Git Object Model

The data model for git is not very complicated as data-structures go. The data model is that of a directed acyclic graph (DAG) of commits where each node in the graph is a commit which represents a specific version (or snapshot) of the repository (repo). The edges in the graph between adjacent nodes/commits represent patches or differences between the different versions of the repo's files and provide the "history" of the changes of the repository. Once you see how git stores the different versions (snapshots) of a repo, you can use that information to understand the impact of various decisions and operations. The Visual Git Reference and nearly every other git reference refers to this DAG of git commits in some way. A more technical description of the git object model is given in the Git Pro "Object Model" chapter. (Git also uses a more compact storage format called Packfiles that can store deltas to changes in files for older history.)

For those that have the time to work through a longer online tutorial, consider taking the following excellent course which teaches both the basics of version control and git as well as how to look at git from a data-structure perspective:

- How to Use Git and GitHub, from Udacity.com

The above course is broken down into well named sections so you can browse the course to find topics where you would like to see someone explain it in simple language, diagrams, and examples.

For those that want a short targeted git tutorial (that also tries to teach the basics of the git object model), see:

- Git for Scientists: A Tutorial
- Why the Heck is Git so Hard? The Places Model

Here are some short git cheat sheets that some may find useful:

- Git Cheat Sheet (from Tower)

- Git Cheat Sheet (from Github)
- Git Cheatsheet (interactive page from NDP Software)

Here are some useful pages with various information about git:

- Git First Aid (includes search of FAQs)
- How to undo (almost) anything in Git
- Most common git screwups/questions and solutions

When in doubt, just do a Google search for what you want to know and you will likely find a great explanation for how to do it on StackOverflow (or some other site).

Git for SVN users

If you are an existing SVN user and want to see corresponding git commands (i.e. the simple single-repo single-branch workflow), see these cheat sheets:

- Git for Subversion Users – A Cheat Sheet
- Git workflow for SVN steadfasts

Login