git --fast-version-control

Introduction to Git

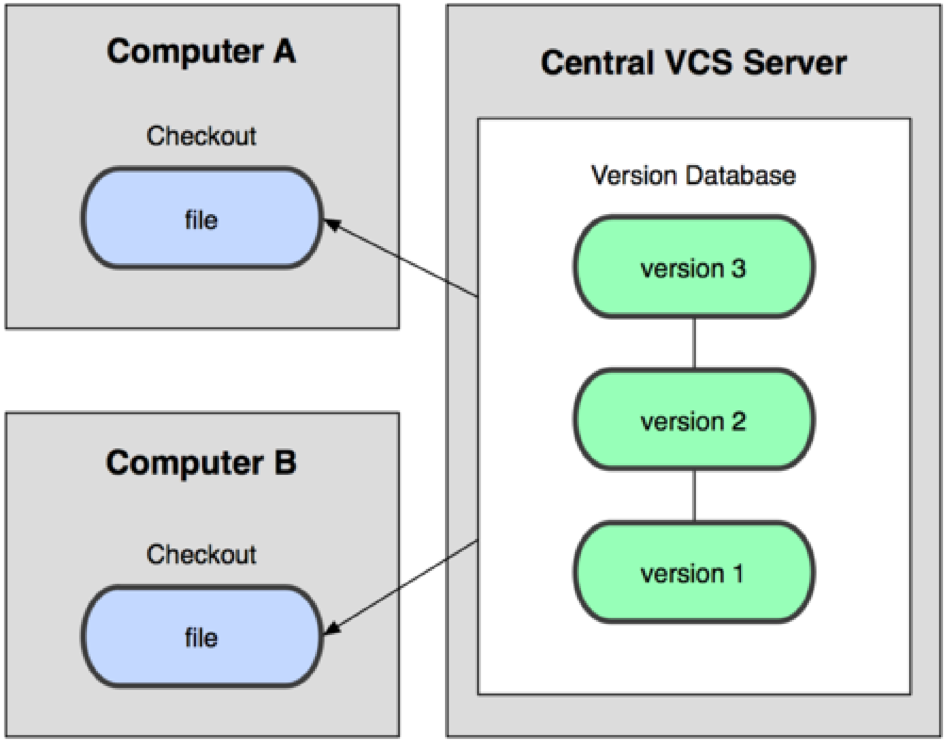
git --distributed-even-if-your-workflow-isnt

Version Control System

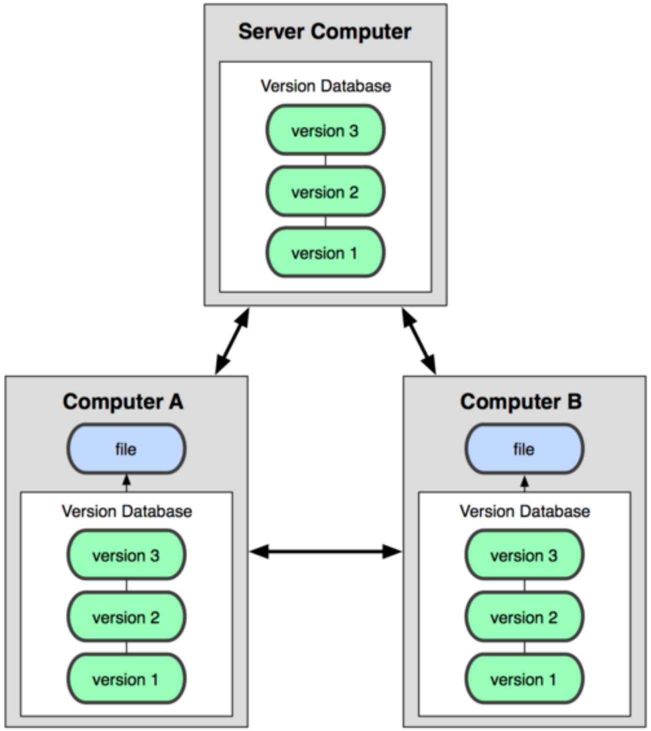
- **Version Control System (VCS)**
  - System that keeps track of changes made to files
- Also known as SCM (Source Code Management)
- Whatever you call it – you got to have one!

Centralized VCS

- CVS, Subversion
  - – Repo resides on some central server
  - – Client only has one version of trunk or branch



Distributed VCS



- Git, Mercurial
  - The full repo resides locally
  - Contains full history
  - Server is (almost) not involved
    - Commit often and offline
    - Work on the beach / train
    - Classified environments?
  - Can push and pull between repos
  - Back ups - trivial and readily available

Git - Brief History

- Developed in 2005 by Linus Torvalds, creator of Linux
- (Prior to Git, Linux was using BitKeeper)
- Developed with the following goals in mind
  - Speed and efficiency
  - Strong support for non-linear development (thousands of parallel branches)
  - Fully distributed

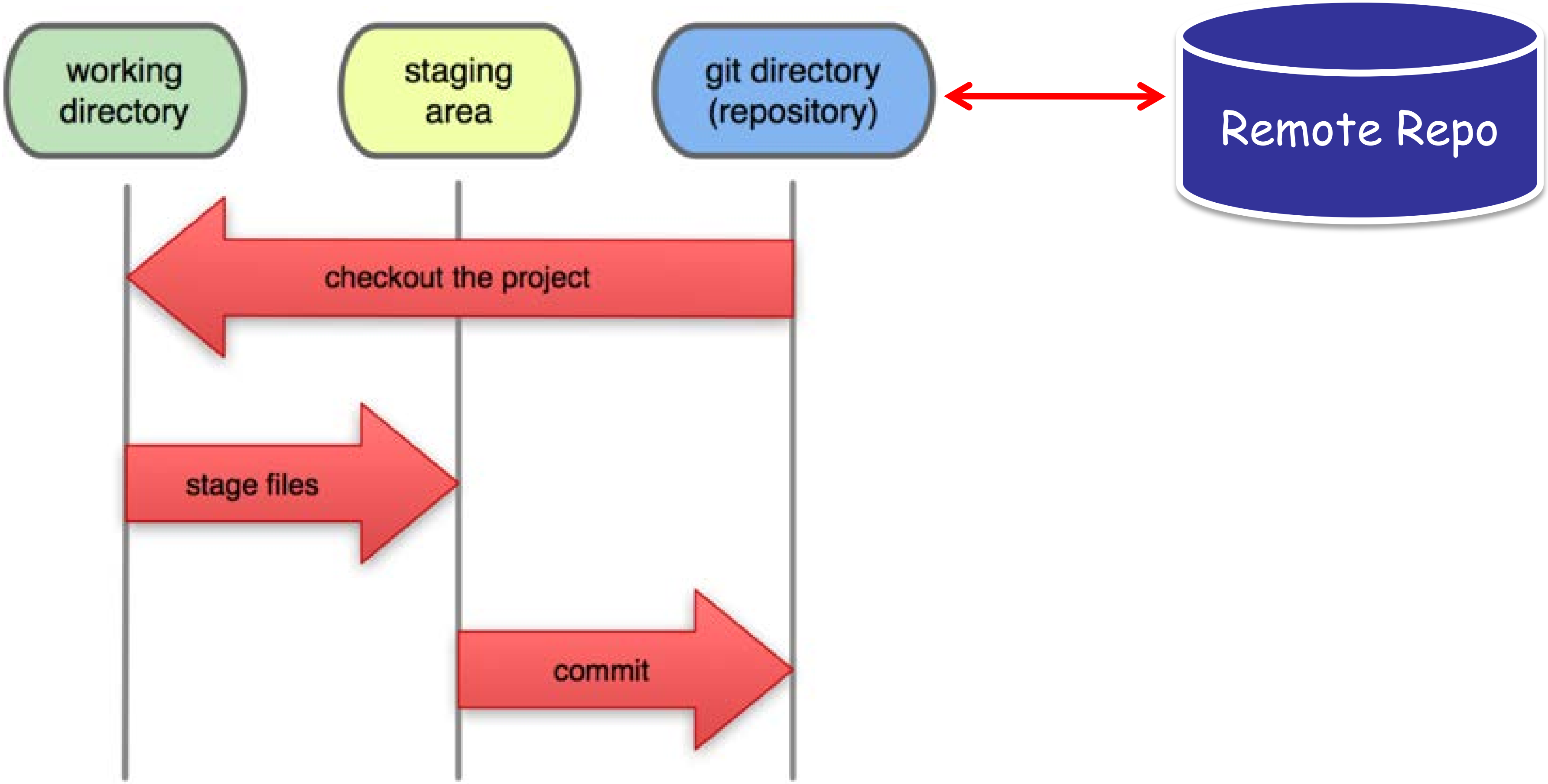
Git Basics

- Nearly every operation is local (FAST!)
- Data managed with integrity
  - 40 character SHA-1 hash (also used in cryptography) computed and assigned to every commit, branch and tag
  - Makes Git aware of hacking attempts or repo corruption

Git Basics (Continued)

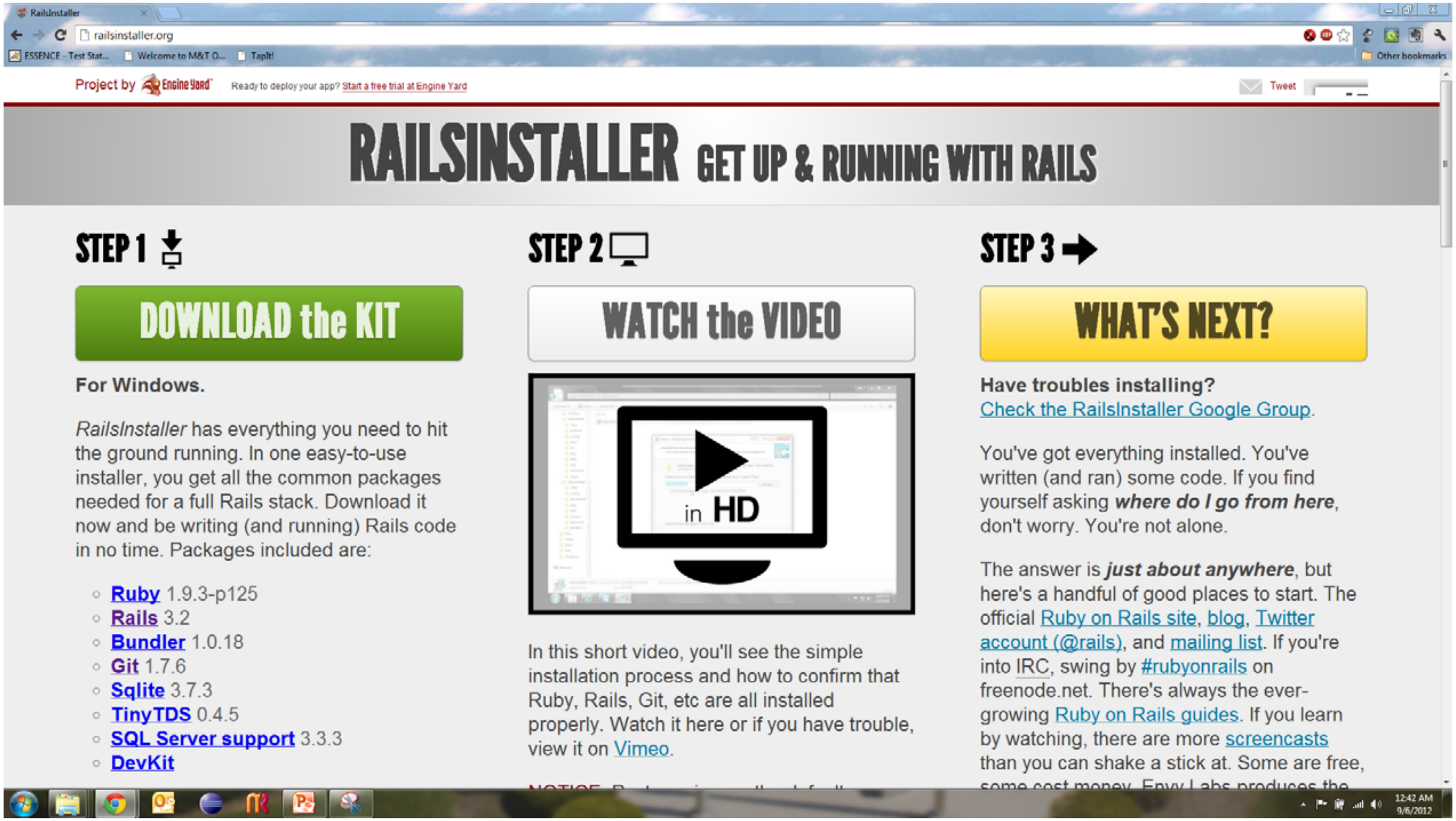
- Generally only adds data
  - Hard to lose committed data
- Unlike other VCSs, Git thinks of data as snapshots - not deltas between files and directories
- Only one `.git` directory at the top level (not sprinkled throughout directory structure like SVN)

General Workflow



Installing Git

- Go to `railsinstaller.org`



RAILSINSTALLER GET UP & RUNNING WITH RAILS

STEP 1

DOWNLOAD the KIT

For Windows.

RailsInstaller has everything you need to hit the ground running. In one easy-to-use installer, you get all the common packages needed for a full Rails stack. Download it now and be writing (and running) Rails code in no time. Packages included are:

- Ruby 1.9.3-p125
- Rails 3.2
- Bundler 1.0.18
- Git 1.7.6
- Sqlite 3.7.3
- TinyTDS 0.4.5
- SQL Server support 3.3.3
- DevKit

STEP 2

WATCH the VIDEO

In this short video, you'll see the simple installation process and how to confirm that Ruby, Rails, Git, etc are all installed properly. Watch it here or if you have trouble, view it on Vimeo.

STEP 3

WHAT'S NEXT?

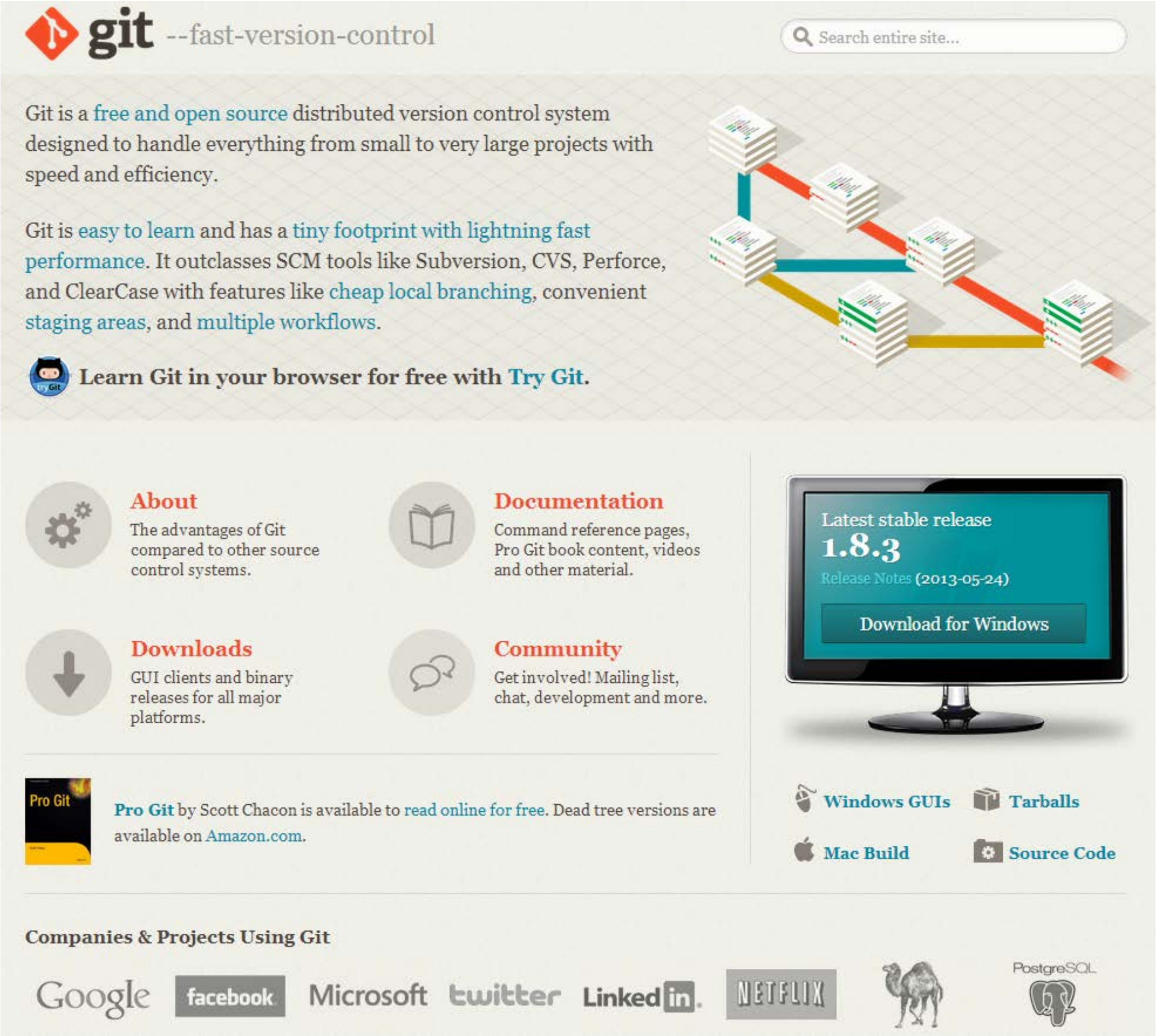
Have troubles installing?
Check the RailsInstaller Google Group.

You've got everything installed. You've written (and ran) some code. If you find yourself asking ***where do I go from here***, don't worry. You're not alone.

The answer is ***just about anywhere***, but here's a handful of good places to start. The official Ruby on Rails site, blog, Twitter account (@rails), and mailing list. If you're into IRC, swing by #rubyonrails on freenode.net. There's always the ever-growing Ruby on Rails guides. If you learn by watching, there are more screencasts than you can shake a stick at. Some are free, some cost money. Envy Labs produces the

Git’s official site

http://git-scm.com



git --fast-version-control

Search entire site...

Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is easy to learn and has a tiny footprint with lightning fast performance. It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like cheap local branching, convenient staging areas, and multiple workflows.

Learn Git in your browser for free with Try Git.

About
The advantages of Git compared to other source control systems.

Documentation
Command reference pages, Pro Git book content, videos and other material.

Downloads
GUI clients and binary releases for all major platforms.

Community
Get involved! Mailing list, chat, development and more.

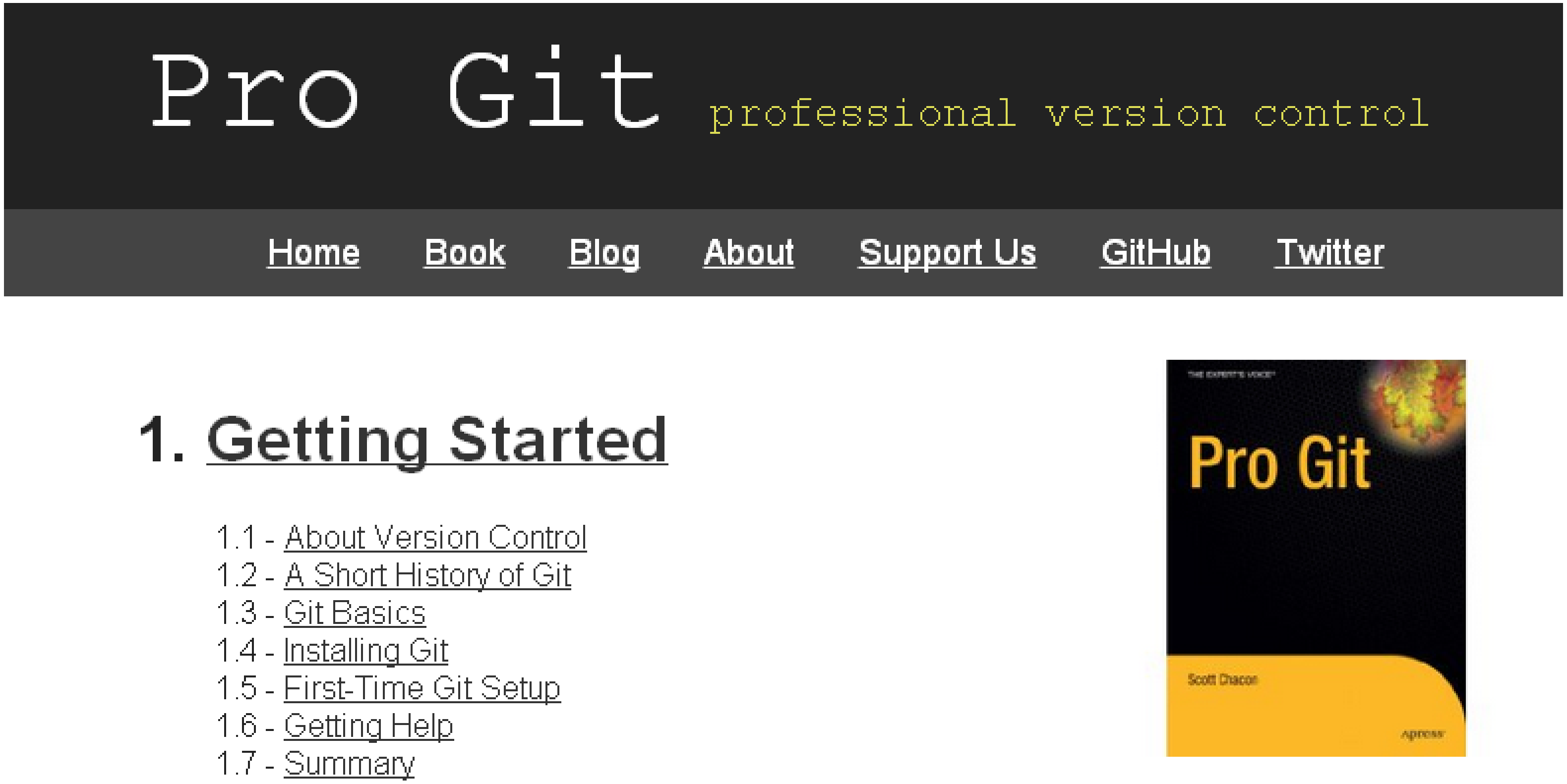
Latest stable release
1.8.3
Release Notes (2013-05-24)
Download for Windows

Windows GUIs
Tarballs
Mac Build
Source Code

Pro Git by Scott Chacon is available to read online for free. Dead tree versions are available on Amazon.com.

Companies & Projects Using Git

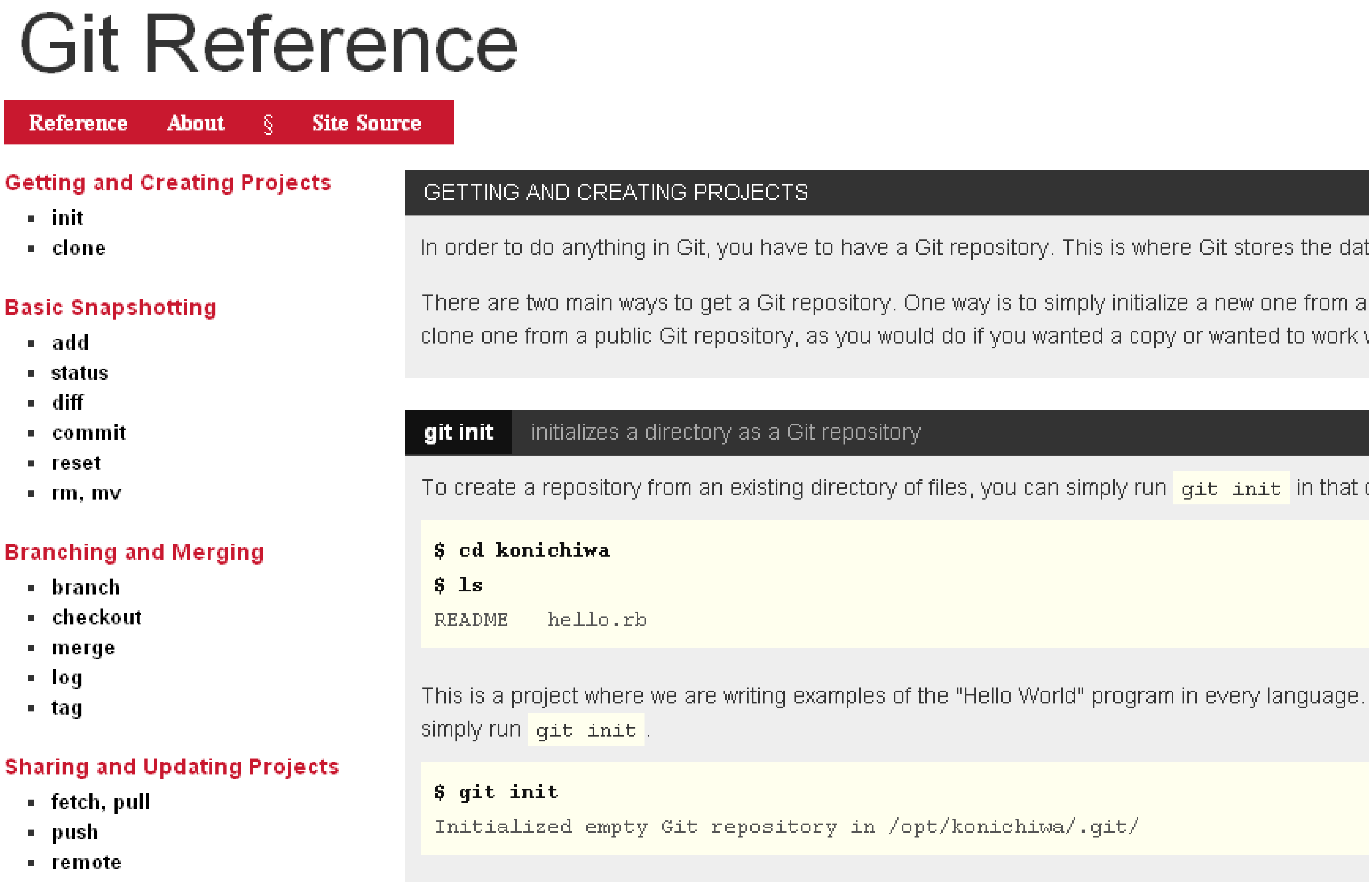
Pro Git - Free Git book (http://progit.org/book/)



1. Getting Started

1.1 - About Version Control
1.2 - A Short History of Git
1.3 - Git Basics
1.4 - Installing Git
1.5 - First-Time Git Setup
1.6 - Getting Help
1.7 - Summary

A nice Git reference - http://gitref.org/



Git Reference

Reference About § Site Source

Getting and Creating Projects

- init
- clone

Basic Snapshotting

- add
- status
- diff
- commit
- reset
- rm, mv

Branching and Merging

- branch
- checkout
- merge
- log
- tag

Sharing and Updating Projects

- fetch, pull
- push
- remote

GETTING AND CREATING PROJECTS

In order to do anything in Git, you have to have a Git repository. This is where Git stores the dat

There are two main ways to get a Git repository. One way is to simply initialize a new one from a clone one from a public Git repository, as you would do if you wanted a copy or wanted to work v

git init initializes a directory as a Git repository

To create a repository from an existing directory of files, you can simply run `git init` in that d

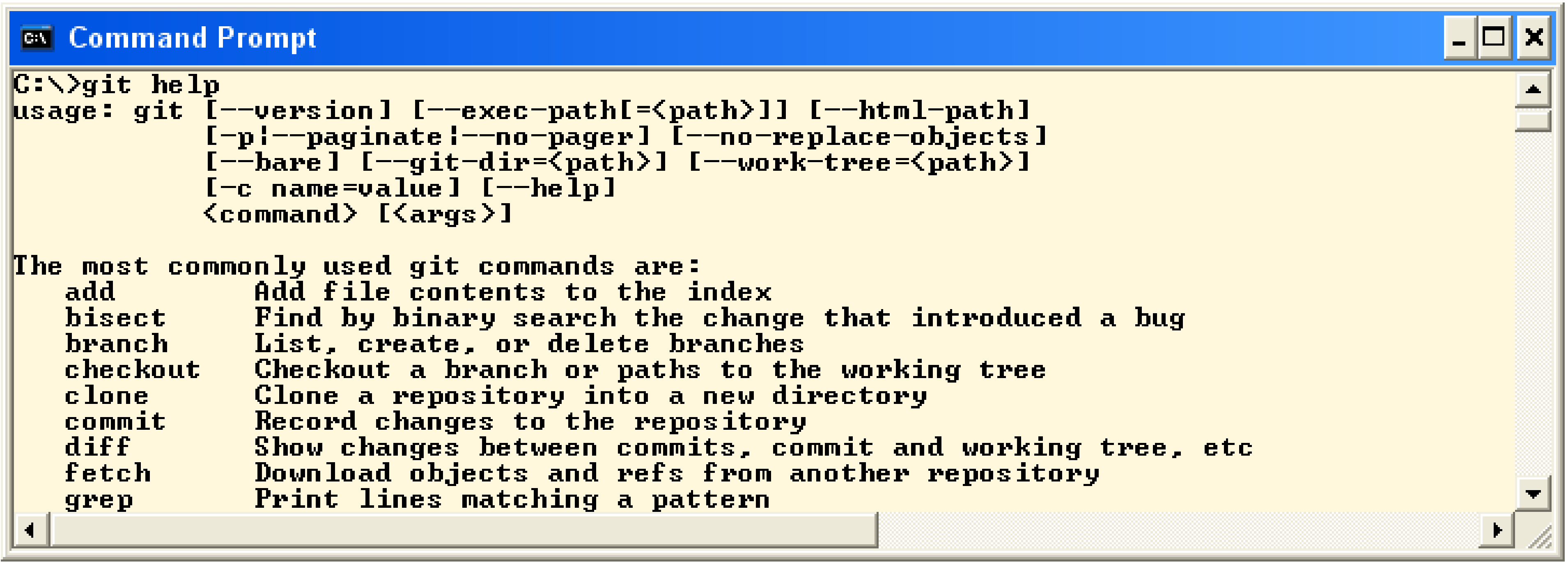
```
$ cd konichiwa
$ ls
README   hello.rb
```

This is a project where we are writing examples of the "Hello World" program in every language. simply run `git init`.

```
$ git init
Initialized empty Git repository in /opt/konichiwa/.git/
```

Before Installing Git (Optional)

- Git has color support and it’s hard to see dark colors on a dark background - consider dark foreground and light background? instead?



```
C:\>git help
usage: git [--version] [--exec-path[=<path>]] [--html-path]
           [-p|--paginate|--no-pager] [--no-replace-objects]
           [--bare] [--git-dir=<path>] [--work-tree=<path>]
           [-c name=value] [--help]
           <command> [<args>]

The most commonly used git commands are:
   add        Add file contents to the index
   bisect     Find by binary search the change that introduced a bug
   branch     List, create, or delete branches
   checkout   Checkout a branch or paths to the working tree
   clone      Clone a repository into a new directory
   commit     Record changes to the repository
   diff       Show changes between commits, commit and working tree, etc
   fetch      Download objects and refs from another repository
   grep       Print lines matching a pattern
```